Copyright is owned by the Author of the thesis. Permission is given for a copy to be downloaded by an individual for the purpose of research and private study only. The thesis may not be reproduced elsewhere without the permission of the Author.

The Design Of An Electric Fence Monitoring System

A thesis presented in partial fulfilment of the requirements for the degree of Master of Technology in Production Technology at Massey University

Paul Robert Adamson

1996


Acknowledge Alarm
Set Threshold
Page Down 1000's
Move Down 100's

Abstract

This thesis presents the design of an Electric Fence Monitoring System (EFMS) which detects and announciates fence malfunctions indicating operational ineffectiveness.

The EFMS consists of a master unit and up to sixteen slave units. Each slave unit monitors a single remote point on the fence. Slave units gain their power from the electric fence pulse itself. They use a unique transmission algorithm to transmit the peak electric fence voltage, to the master unit. The electric fence wire is used to convey this transmission.

The master unit uses a non-linear switched capacitor filter with variable gain control, to detect the slave unit transmissions. This unit displays the peak voltage at each monitored point and allows the setting of alarm trigger points.

This thesis includes modelling of the electric fence energiser and typical electric fence line, and the detailed design of the two units that makeup the EFMS.

Acknowledgements

I would especially like to thank Dr Ross Nilson for all of his assistance thoughout the duration of this work. His door was always open and he was ready to discuss problems and suggest solutions. Thanks.

Chris Keith of Speedrite for his assistance, advice and always being ready to put aside his tasks to answer my questions on electric fencing.

Thanks also to Ralph Ball for his initial liason with Speedrite and the instigation of the EFMS project.

Contents

List Of Figures

Figure

Figure

Figure

List Of Tables

Table

Units And Symbols

Quantity	Symbol for Quantity	Unit	Symbol for Unit
Admittance	Y	siemens	S
Angular velocity	ω	radians/second	rad/s
Area	A	square metre	m^2
Capacitance	C	Farad	F
Charge	Q	Coulomb	C
Conductance	G	siemens	S
Current	I	Ampere	A
Energy	E	Joule	J
Flux	θ	Weber	Wb
Flux Density	B	Tesla	T
Frequency	f	Hertz	Hz
Impedance	Z	Ohm	Ω
Inductance	L	Henry	H
Instantaneous current	i	Ampere	A
Length	l	metre	m
Magnetic Field Strength	H	Ampere/metre	A/m
Number of turns on winding	N	(dimensionless)	
Period	T	second	s
permeability	μ	henry/metre	H/m
Power	P	Watt	W
Resistance	R	Ohm	Ω
Selectivity	Q	(dimensionless)	
Time	t	second	s
Voltage	V	Volt	V

The following prefixes are used in conjunction with the above units.

Prefix	Symbol	Fraction
pico	p	10^{-12}
nano	n	10^{-9}
micro	μ	10^{-6}
milli	m	10^{-3}
kilo	k	10^{3}
mega	M	10^{6}

The following symbols are used throughout this thesis:

Quantity	Symbol for Quantity	Unit	Symbol for Unit
Propagation Constant	γ	(dimensionless)	
Transformer gap permeability	μ_{AIR}	Henry/metre	H/m
Transformer core permeability	μ_{CORE}	Henry/metre	H/m
Sample duration	Δt	second	s
Effective area of transformer core	A_l	square metre	m^2
Energiser storage capacitance	C_e	Farad	F
Electric fence capacitance	C_F	Farad	F
Electric fence capacitance	C_f	Farad	F
RFI suppression capacitance	C_S	Farad	F
Energy lost into inter-winding capacitance	E_{cap}	Joule	J
Energy lost in resistance of transformer windings	E_{copper}	Joule	J
Energy lost in resistance of transformer primary winding	$E_{copper.primary}$	Joule	J
Energy lost in resistance of transformer secondary winding	$E_{copper.secondary}$	Joule	J
Energy lost in diodes	E_{diode}	Joule	J
Energy lost in rectifier diodes	$E_{diode.rectifier}$	Joule	J
Energy lost in regulator diode	$E_{diode.regulator}$	Joule	J
Energy delivered into primary of transformer	E_{in}	Joule	J
Energy lost in resistance of inductor	$E_{inductor.wire}$	Joule	J
Energy lost in measurement components	$E_{measure}$	Joule	J
Transmission energy	E_T	Joule	J
Energy loss not attributable to one specific cause	$E_{transfomer}$	Joule	J
Filter centre frequency	f_0	Hertz	Hz
Filter clock frequency	f_{CLK}	Hertz	Hz
Power supply form factor	ff	(dimensionless)	
Transformer gap factor	G_F	(dimensionless)	
Filter output high pass gain	H_{OHP}	(dimensionless)	
Filter output low pass gain	H_{OLP}	(dimensionless)	
Filter output notch gain	H_{ON2}	(dimensionless)	
Power supply AC current	I_{AC}	Ampere	A
Power supply DC current	I_{DC}	Ampere	A
Transformer input current	I_{in}	Ampere	A
Transformer output current	I_{out}	Ampere	A
Peak current	I_{peak}	Ohms	Ω
Diode saturation current	I_S	Ampere	A
Chebyshev filter gain constant	K	(dimensionless)	

Transformer core magnetic path length	l_{CORE}	metre	m
Transformer gap length	l_{GAP}	metre	m
Transformer magnetising inductance	L_M	Henry	H
Slave unit power supply inductance	$L_{power.supply}$	Henry	H
RFI suppression inductance	L_S	Henry	H
Leakage Inductance of transformer primary	L_{T1}	Henry	H
Leakage Inductance of transformer secondary	L_{T2}	Henry	H
Electric fence inductance	L_W	Henry	H
Number of energiser pulses counted	m	(dimensionless)	
Number of data samples	N	(dimensionless)	
Number of energiser pulses	n	(dimensionless)	
Number of slave units	n	(dimensionless)	
Number of capacitive current data samples	N_C	(dimensionless)	
Number of falling output current data samples	N_F	(dimensionless)	
Number of rising output current data samples	N_R	(dimensionless)	
Transformer turns ratio	N_R	(dimensionless)	
Number of positive responses	p	(dimensionless)	
AGC resistance	R_{AGC}	Ohms	Ω
Ground Resistance	R_G	Ohms	Ω
Load Resistance	R_L	Ohms	Ω
Transformer core loss resistance	R_M	Ohms	Ω
Resistance of measurement components	$R_{measure}$	Ohms	Ω
Filter equivalent feedback resistance	R_N	Ohms	Ω
Transformer primary resistance	R_{T1}	Ohms	Ω
Transformer secondary resistance	R_{T2}	Ohms	Ω
Thevenin equivalent resitance	R_{TH}	Ohms	Ω
Electric fence wire resistance	R_W	Ohms	Ω
Resistance of transformer winding	R_{wire}	Ohms	Ω
Time taken for capacitor to charge	t_c	second	s
Transformer utilisation constant	TU	(dimensionless)	
Power supply AC voltage	V_{AC}	Volt	V
Voltage drop across transistor base emitter junction	V_{be}	Volt	V
Measurement capacitor voltage	V_c	Volt	V
Power Supply Voltage	V_{CC}	Volt	V

Power supply DC voltage	V_{DC}	Volt	V
Power supply rectifier diode voltage drop	V_{DIODE}	Volt	V
Power supply regulator voltage drop	V_{DROP}	Volt	V
Peak fence voltage	V_{fence}	Volt	V
Voltage drop across FET drain source	V_{FET}	Volt	V
Transformer input voltage	V_{in}	Volt	V
Maximum output voltage	V_{max}	Volt	V
Transformer output voltage	V_{out}	Volt	V
Peak transmission voltage	V_P	Volt	V
Voltage drop across pullup resistor	V_{pullup}	Volt	V
Measurement resistor voltage	V_r	Volt	V
Power supply ripple voltage	V_{Ripple}	Volt	V
Diode threshold voltage	V_T	Volt	V
Transformer voltage	V_T	Volt	V
Thevenin equivalent voltage	V_{TH}	Volt	V
Voltage across energiser	V_{Ze}	Volt	V
Shunt admittance per unit length	y	Siemens	S
Equivalent Admittance	Y'	Siemens	S
Series Impedance per unit length	z	Ohms	Ω
Equivalent Impedance	Z'	Ohms	Ω
Characteristic Impedance	Z_c	Ohms	Ω
Energiser impedance	Z_e	Ohms	Ω
Impedance of slave unit	Z_{slave}	Ohms	Ω

Chapter 1
Introduction

This thesis sets out the modelling, design, and testing of a Electric Fence Monitoring System (EFMS) developed for Speedrite International. Due to the nature of the construction of electric fences, ensuring the integrity of the electric fence system is not an easy task. Should an electric fence fail and no corrective action taken, the consequences of stock breaking through the fence could result in damage to crops, loss of stock, breeding problems or pasture management problems. The EFMS is designed in order to alert the farmer to any malfunction in the operation of the electric fence.

The EFMS consists of up to sixteen slave units and a master unit. The slave unit is located at a desired measurement point on the fence and periodically transmits, on the fence line itself, to the master unit the peak fence voltage experienced at the slave unit. Several slave units enable a picture of the entire fence to be built up at the master unit. The master unit interprets the voltage responses from the slave unit and alerts the farmer if necessary. The master unit includes an interface which allows the user to set their own desired alarm parameters for the EFMS. Figure 1.1 shows the configuration of the EFMS.

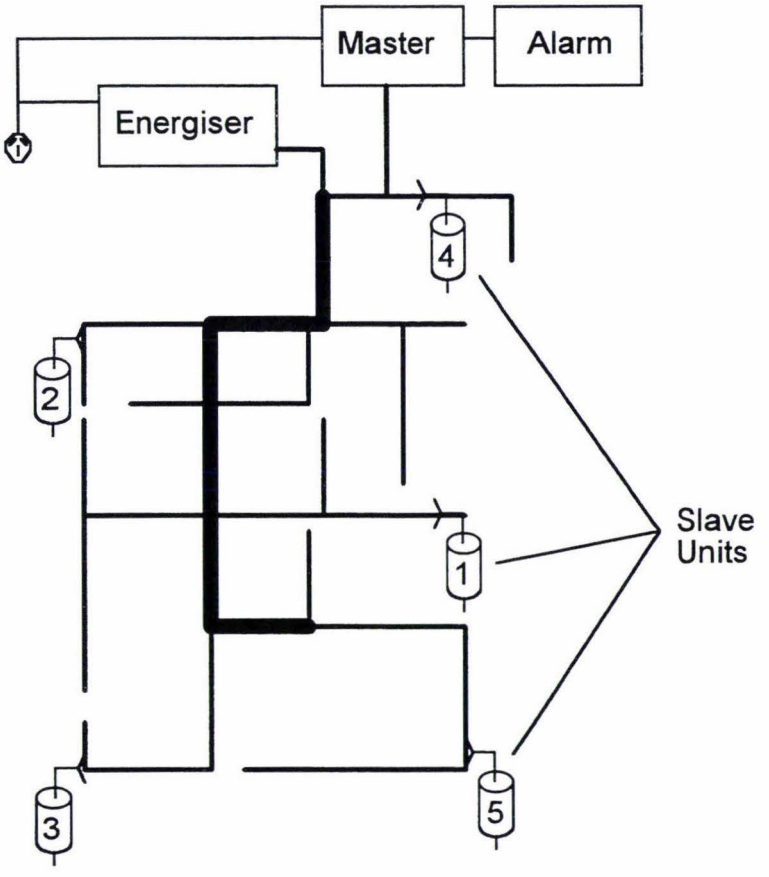


Figure 1.1 EFMS System Configuration.

The EFMS is a practical system which has been fully developed to a commmercially viable prototype.

Chapter 2 outlines the theory of operation of electric fence energisers and the environmental conditions in which they operate. Current methods of fence monitoring are presented and their merits discussed. The Operating Parameters for the EFMS are presented. Important issues for the design of the EFMS are also discussed.

Models of an electric fence and an electric fence energiser are presented together with the simulation results in chapter 3. The energiser model includes the modelling of a non-linear energiser output transformer, where the inductance of the transformer is varied according to the current flowing through the inductor. The fence and energiser models are combined to help define the operating environment for the electric fence monitoring system. Chapter 3 also presents a transmission algorithm for the EFMS system where the energy used to transmit the peak fence voltage is related to the energy available to transmit the peak fence voltage.

Chapter 4 introduces the slave unit. A power supply is presented which enables the slave unit to be powered completely from the energiser pulse. The transformer in this power supply is critical to the efficiency of the power supply. This is studied and optimised toward maximising the power supply efficiency. Several options for a measurement circuit are presented and each evaluated according to cost, accuracy and

power efficiency criteria. A transmission circuit is presented along with a model of the circuit which is used to evaluate the performance of this circuit. The additional slave unit circuitry required, along with the structure of the software, is also discussed.

Chapter 5 introduces the master unit. Slave transmission detection is based around a switched capacitor chebyshev filter with a narrow passband. By varying the filter clock speed this filter is able to remove artifacts from previous slave unit transmissions and double the number of slaves units, while keeping the same noise rejection specifications. An algorithm using the filter output to detect the presence of slave transmissions is discussed. The additional circuitry that is required for the master to complete its tasks is discussed. This includes energiser detection circuitry, microprocessors, microprocessor reset circuitry, EEPROM, siren/buzzer circuitry, filter automatic gain control, filter signal conditioning, optical isolation, and power supply circuitry. The structure of the master unit software is also presented.

A series of tests have been performed on the EFMS. The tests and the results of these tests are presented in chapter 6.

Chapter 7 concludes the work presented in this thesis. Extensions to the EFMS are also discussed.

Chapter 2
The Electric Fence

This chapter outlines the purpose and the operation of standard electric fence systems and discusses issues and implementations involved in providing remote monitoring of the fence.

Section 2.1 contains a detailed discussion of the working of an electric fence system. The principles of operation and legal requirements of the electric fence energiser are presented. An example circuit for an energiser is presented along with typical output pulse waveforms. The electric fence is also characterised.

Section 2.2 gives a discussion of the difficulties involved in monitoring the electric fence.

In section 2.3 existing monitoring systems and devices are discussed. This leads to the definition of the ideal electric fence monitoring system, which is an aim of this work.

2.1 Electric Fence Operation

This section outlines the operation of an electric fence system and gives a detailed discussion of the electric fence energiser.

2.1.1 The Electric Fence System

The electric fence originated in the 1930s. These fences were of crude but simple design. They produced an electric pulse of a high voltage (in some cases possibly lethal) between a fence wire and ground. An animal in contact with both the fence and the ground completes the electric circuit (shown in Figure 2.1). The resulting shock hopefully deters the animal from approaching the fence again. Several texts contain a description of these energisers, and various stock reactions to them (Armstrong, Banks and Gill (1981), East (1993), McCutchan (1980), Studman (1989)).

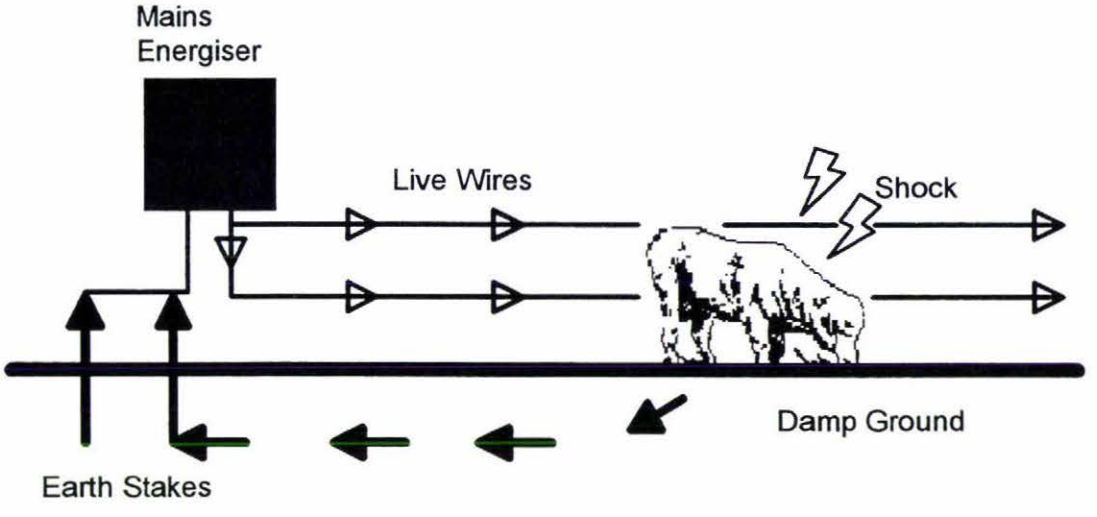


Figure 2.1 Electric Fence Operation (From Speedrite International (1995)).

In cases where the ground resistance is high and the animal does not receive a shock of significant magnitude, an earth return wire can be used. In this case the animal would have to come into contact with both of the wires in order to complete the circuit.

Any leakage (at fence posts, tree branches etc) will reduce the effectiveness of the electric fence system. Therefore it is important to keep the insulation between the wires and the ground.

2.1.2 Legal Energiser Pulse Requirements

Due to the hazardous nature of electric fences most legislation introduced aims to prevent misuse and provide safe operating regions. This legislation limits the energy the fence can deliver. The basis for these requirements come from an IEC standard.

Details of the New Zealand/Australia requirements are given in AS/NZS 3129.1:1993 and AS/NZS 3129.2:1993. These are summarised in Table 2.1.

The energiser pulses must be separated by intervals of not less than 1s. The characteristics for the output must not exceed the values given in the following table:

Table 2.1 Legal Energiser Requirements

Peak value of voltage	10 000 V
Maximum duration of impulse	0.05 s
Maximum quantity of electricity per impulse	2.5 mC
Maximum discharged energy per impulse	8.0 J

Note that there are also requirements for RFI suppression and Isolation (NZS3350.1:1994).

2.1.3 Energiser Operation

The main principle of the operation of a modern electric fence energiser is the production of a high voltage pulse by connecting a precharged storage capacitor across a stepup transformer attached to the fence. The pulse rapidly decays as the capacitor discharges via the transformer onto the fence. With the origin of electric fences this switching had to be accomplished through the use of mechanical switches or gas discharge tubes, however today solid state switches such as thyristors or silicon controlled rectifiers (SCR's) are used.

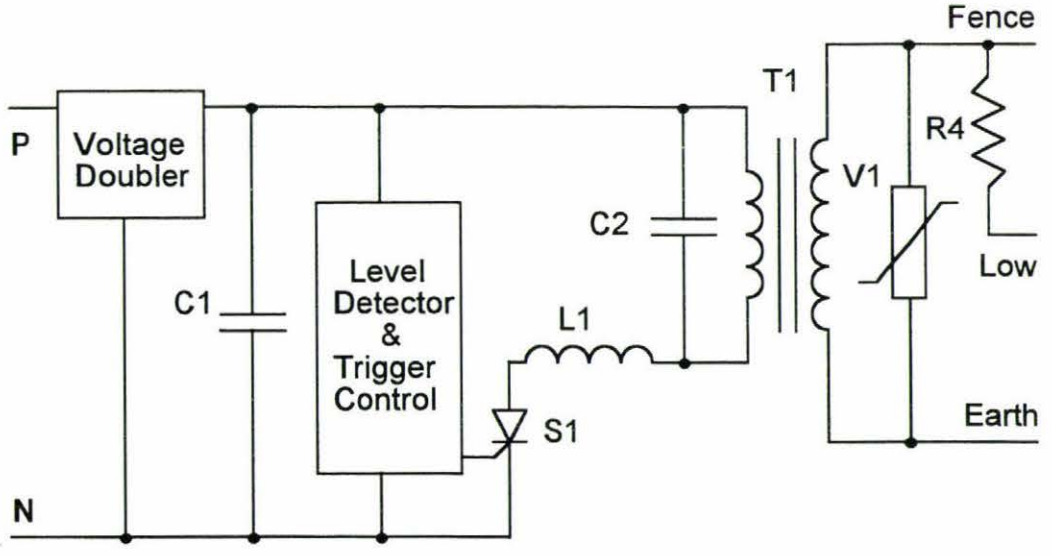


Figure 2.2 Typical Energiser Circuit Diagram

A typical energiser circuit diagram is shown in Figure 2.2. C1 is the main storage capacitor for the energiser pulse. This is charged up to 650V through the voltage doubler circuit. The pulse discharge is controlled by the level detector. When the

trigger voltage is reached SCR S1 is triggered. This SCR causes the storage capacitor C1 to rapidly discharge through the step-up transformer and form the energiser pulse at the output.

C2 and L1 are included as a filter to remove the RFI content of the energiser pulse. In lower power energisers they are not required. Varistor V1 alters its resistance dependant upon the voltage across it. It is added to prevent the output peak voltage exceeding legal limits, and to help protect the energiser from some of the effects of lightning.

R4 is included as a series impedance to give the energiser a low output option for use in cases of high fire risk, or where the animals are particularly sensitive to electric shock.

Early electric fence energisers employed a different output topology and suffered from an inability to maintain a large peak voltage under conditions of a heavy load (or 'low fence impedance'). The modern energiser overcomes this problem and has thus (perhaps inappropriately) been termed a low impedance energiser.

2.1.4 Typical Energiser Output

For a typical 1.2J energiser the output reaches a peak value of 6300 Volts and lasts for a duration of approximately 60μs on a 5kΩ load. The output for this energiser is shown in Figure 2.3

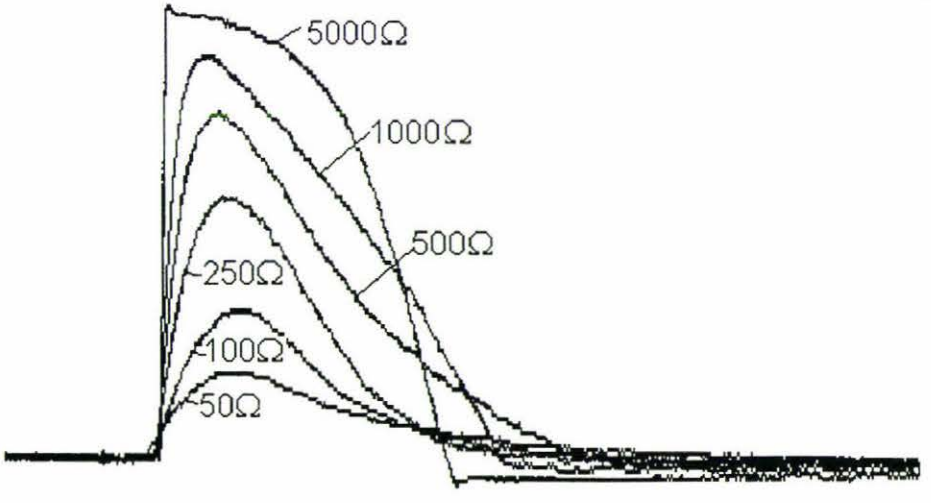


Figure 2.3 Typical Low Power Energiser Output Form for Various Loads

The more powerful energisers produced usually involve RFI suppression componentry as the pulses are a lot larger in magnitude and duration and exhibit more 'ringing'. A typical RFI suppression energiser output characteristic is shown in Figure 2.4.

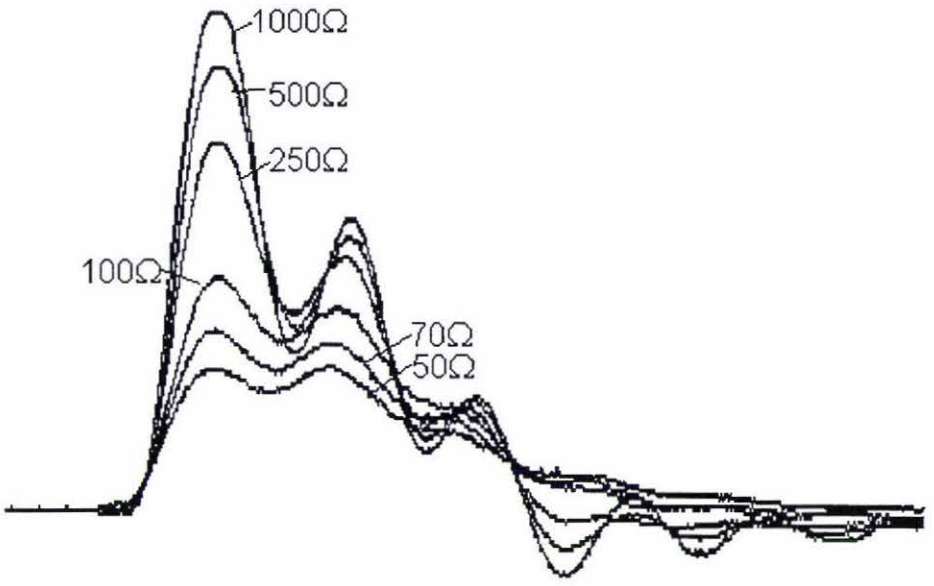


Figure 2.4 Typical RFI Suppression Energiser Output Form, for Various Loads

2.1.5 Fence Characteristics

In practice the characteristics of an agricultural electric fence line are very similar to those of a transmission line and can be defined by distributed resistance, capacitance, inductance, and conductance.

In order to simplify the system, the fence could be represented by a nominal Π circuit where the parameters are the characteristics per unit length multiplied by the length of the fence. Unfortunately this is inaccurate for long fence lengths. It has been shown that adjustments can be made such that as far as the conditions at the terminals are concerned the equivalent Π circuit can be made accurate (Stevenson (1962)).. Figure 2.5 shows the equivalent Π circuit and the accompanying equations are given in equations 2.1 to 2.4.

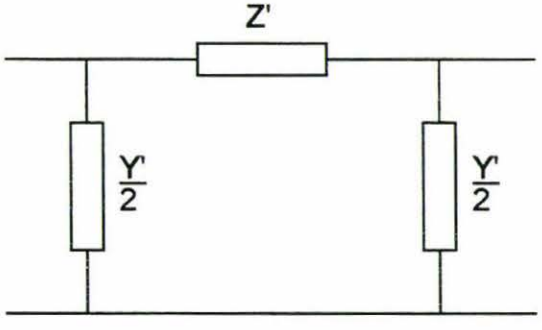


Figure 2.5 Equivalent Π Circuit

$$Z' = Z_c \sinh \gamma l \tag{2.1}$$

$$\frac{Y'}{2} = \frac{1}{Z_c} \tanh \frac{\gamma l}{2} \tag{2.2}$$

$$Z_c = \sqrt{\frac{z}{y}} \tag{2.3}$$

$$\gamma = \sqrt{yz} \tag{2.4}$$

Where Zc is the characteristic impedance, γ is the propagation constant and l is the length of the fence. z is the series impedance per unit length and y the shunt admittance per unit length. The hyperbolic sin and tan are the adjustment factors that convert from the nominal circuit to the equivalent circuit. For the typical electric fence circuit the impedance is made of approximately 44Ω/km and 4mH/km. For the shunt admittance the conductance is usually negligible and the capacitance is approximately 8nF/km. For the electric fence circuit consideration of the ground resistance must also be made, however this is extremely variable (depending upon soil condition etc.) and for the entire fence should not be in the order of more than a few hundred ohms for correct operation.

The fence line behaves like a low pass filter. This necessitates that any transmission on the fence line be of low frequency to avoid attenuation of the transmission. Chapter 3 contains a case study on the fence line showing the effects of the fence on the desired transmission frequency.

2.2 Problems With Electric Fence Operation

Today the use of electric fences has become widespread, especially in countries such as New Zealand and Australia. In some cases these fences are very large, perhaps several hundred kilometres of wire in the extreme case. These fences have a single energiser connected only at one point. The fences also do not form a single closed loop, but rather are characterised by a tree structure, containing a main trunk to distribute the pulse, and limbs to branch out from this main trunk. Figure 2.6 shows an example electric fence layout.

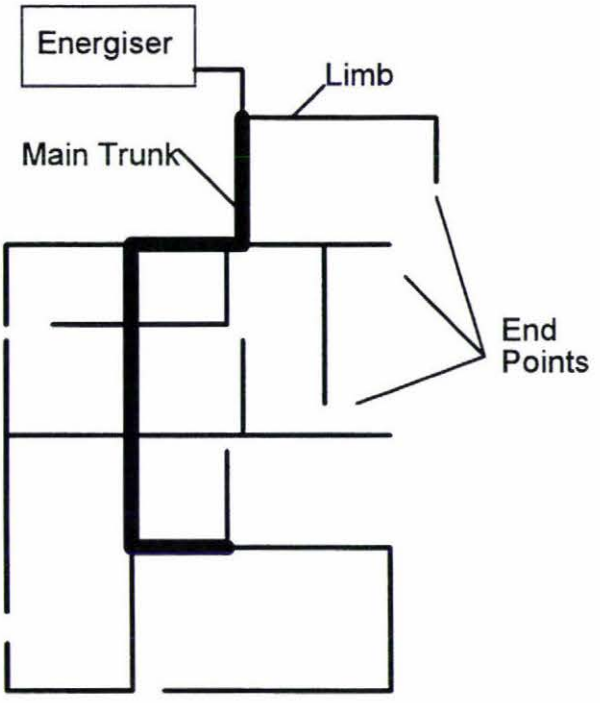


Figure 2.6 Example Electric Fence Layout

This layout reduces the reliability of the electric fence system since there are several limbs, preventing a simple closed loop monitor system being employed. A section could fail, due to partial shorts or breaks, with no obvious indications of this failure. Here the energiser itself will still be running and even giving the characteristic audible click, along with associated flashing lights.

To the farmer the consequences of a electric fence failure can be very severe. In time stock will discover the fence failure and will break through the fence, causing stock to be lost, crop damage, fence damage etc.

2.3 Current Fence Monitoring Solutions

There is a need to ensure the integrity of the electric fence system. This means ensuring the energiser pulse is reaching the distant points of the fence. This could be achieved if some device was connected to the fence at these points to monitor the energiser pulse.

Some products are currently available to test the integrity of a fence. This section outlines a few of these and details the disadvantages which could be overcome with the EFMS.

2.3.1 Electric Fence Voltmeter

The simplest product is the electric fence voltmeter (Speedrite International (1995)). This connects between the fence and earth, and displays the peak fence voltage. An example is shown in Figure 2.7.

Figure 2.7 Speedrite Electric Fence Voltmeter

While this may be practical for small fences to have the farmer check every point manually, for large fences where the limbs may be tens of kilometres apart, this is impractical.

2.3.2 Agricultural Monitor

This unit connects between the fence and earth and detects fence pulses (Speedrite International (1995)). When no fence pulse is detected an alarm is set warning of the fence failure. Figure 2.8 shows an example of the product.

Figure 2.8 Agricultural Fence Monitor

The unit is small and is suitable for remote location. This device provides the ability to detect failure but needs extra equipment to announce or transmit this information to the farmer.

2.3.3 Fixed Monitoring Devices

An alternative product for large fence systems uses fixed monitoring devices with remotely activated lights to give a go/fail indication. Observation of these can be made from aircraft in very large systems.

While providing effective remote monitoring this system does not provide continuous monitoring and the cost is prohibitive for most applications.

2.3.4 Wireless Alarm Monitoring Systems

This system allows central monitoring of the fence system [REF]. It is generally used in a security application, but can be adapted for fence monitoring. Transmitters each with unique identifiers are constantly polled to indicate the fence status. This system allows continuous multipoint distant remote monitoring. Figure 2.9 shows an example system.

Figure 2.9 Wireless Monitoring System

The major disadvantages of this system are that the monitor units all require an external power source and the cost of the system is very high.

2.3.5 McCutchan Monitoring System

In the early 1970's McCutchan proposed the idea of a multipoint fence monitoring system that communicated on the electric fence wire itself. A diagram of his system is shown in Figure 2.10.

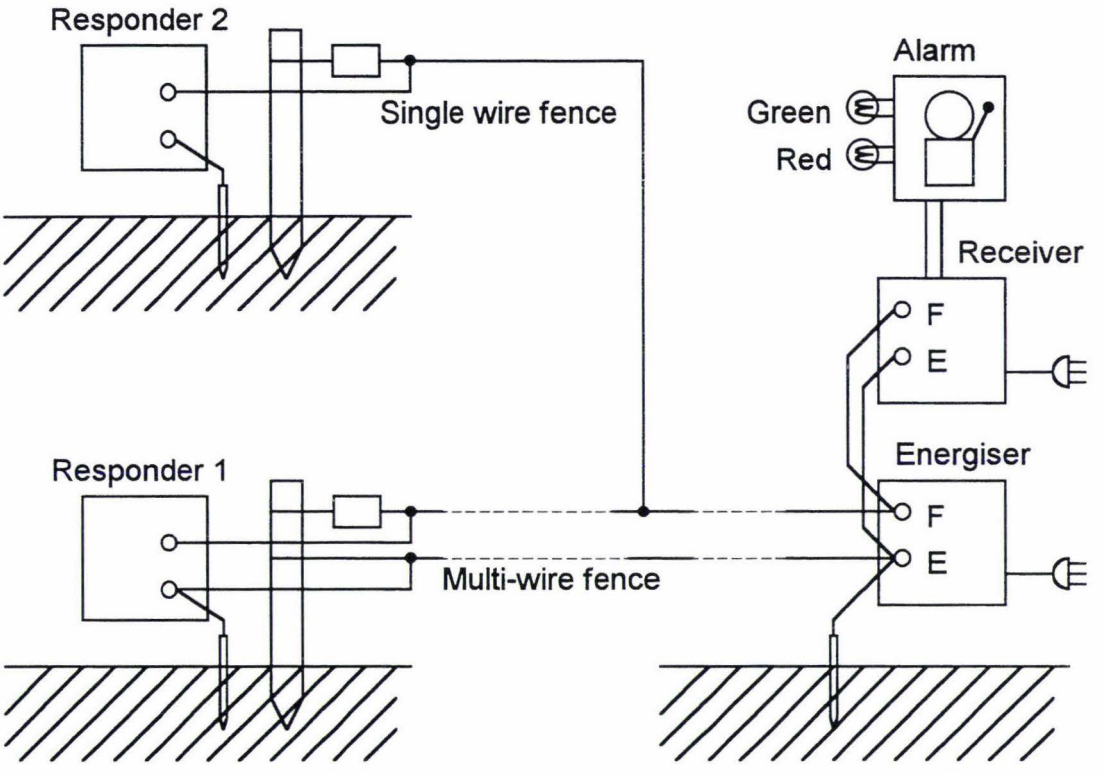


Figure 2.10 McCutchan Monitoring System (McCutchan (1980)).

The commercial system allowed up to four remote sites and of the systems presented here comes closest to the EFMS system ideal.

It has limitations in that the voltage reading at the site is limited in its accuracy, only four remote units are available for each system, and most importantly is this system's inability to withstand noise on the fence line.

2.3.6 Problems With Current Monitoring Solutions

While the systems mentioned above go some way to providing a level of assurance of fence integrity, they all have their limitations. These include: a failure to work for distant sites, non continuous monitoring, a large amount of auxiliary equipment required, the need for a external power supply, a failure to work in the presence of noise on the fence line, or a high cost required to set up and run the system.

2.4 Parameters For Ideal Monitoring System

A more ideal monitoring system would employ a device which when connected to the fence had no need for an external power supply, used distant continuous communication to a central monitoring site, allowed for multipoint monitoring, was cost effective, simple, robust and accurate.

This specification implies that there will be two units required for the task. One which is remote and placed at the desired monitoring point, and the second which is located at a central location.

The remote units could incorporate a transmission circuit. As the units are already connected by a wire, the easiest method of coupling a transmission between the units would be on the fence wire itself. Any transmission between the units would have to occur between energiser pulses, this means that both units would have to be able to detect energiser pulses, and use the gaps between consecutive pulses to transmit when required.

The need for an external power supply is eliminated if the unit is powered from the energiser pulse. This means that it is necessary that the power consumption, including the transmission power, be less than the power available from the minimum effective fence voltage.

The use of multipoint monitoring implies that each of the remote units will require a unique identification such that the central monitoring point can distinguish which remote unit is transmitting a signal. Communication is only required to be unidirectional, as the remote units will need only to transmit information and not to receive.

Chapter 3
The EFMS System

This chapter is concerned with defining the operational requirements for the EFMS (Electric Fence Monitoring System). This study considers the two units (Slave and Master), the energiser and the electric fence which make up the operating system and the interaction between these units. Chapters 4 and 5 will fully detail the slave and master units.

Section 3.1 presents a system outline which splits the required tasks into two units and briefly details the interaction and operation of these units.

Section 3.2 is concerned with modelling the EFMS environment. A model of the electric fence is presented This model is then extended to include the energiser and electric fence. The energiser portion of the model accounts for the heavy stresses that the energiser output transformer is under during an energiser pulse. Uses of the models and their accuracy are discussed.

A case study of the energy available to a slave unit is presented in section 3.3. It shows how much power is available to a slave unit from a typical electric fence energiser. This allows estimation of the power available for slave unit to run and communicate with the master unit.

Section 3.4 details the method of communication between the slave and master units. Several options are considered and the result is a method of communication with an energy consumption that is proportional to the energy available.

3.1 System Outline

This section concerns itself with the design parameters of the ideal system described in the previous chapter. It is logical that the EFMS consists of two types of units. The first unit performs the tasks required at the remote location of measuring the peak fence voltage and transmitting on the fence. This unit is termed the Slave Unit. The second unit performs the tasks at the central location. Termed the Master Unit it is responsible for detecting the Slave Unit transmissions and interfacing with the user in order to determine alarm conditions. The alarm will be located at the master unit and controlled directly by the master unit.

The system is detailed in Figure 3.1.

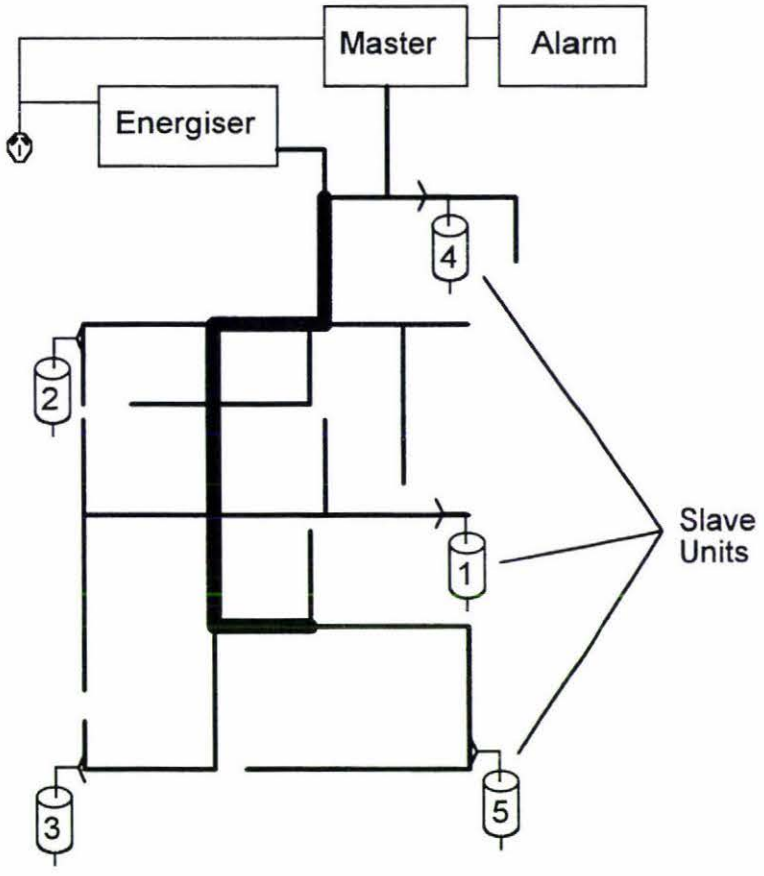


Figure 3.1 EFMS Configuration

The user will locate the master unit at a convenient central location (ie inside the house or barn etc), and it is reasonable to expect that this location will have mains power. The master unit does not have to be placed near the energiser. The slave unit will have to be located at the point of the fence that the user wishes to monitor the fence voltage and will derive its power from the fence pulse.

The chosen method for multiplexing the communication is Time Division Multiplexing (TDM) as opposed to Frequency Division Multiplexing (FDM). This reduces the need for a receiver that can detect multiple frequencies or the need for a transmitter that can produce anyone of several different frequencies. The maximum number of

slave units was selected as sixteen as this would still allow a considerable time for each slave to transmit, and it is expected that the user would very rarely require more than sixteen slave units.

The time between energiser pulses is divided up to allow each slave unit to measure the pulse, and to transmit to the master unit at least once for every energiser pulse. Legal requirements dictate that there must be at least one second between consecutive energiser pulses and thus the EFMS must finish all communication within one second of the energiser pulse. The time division is shown in Figure 3.2.

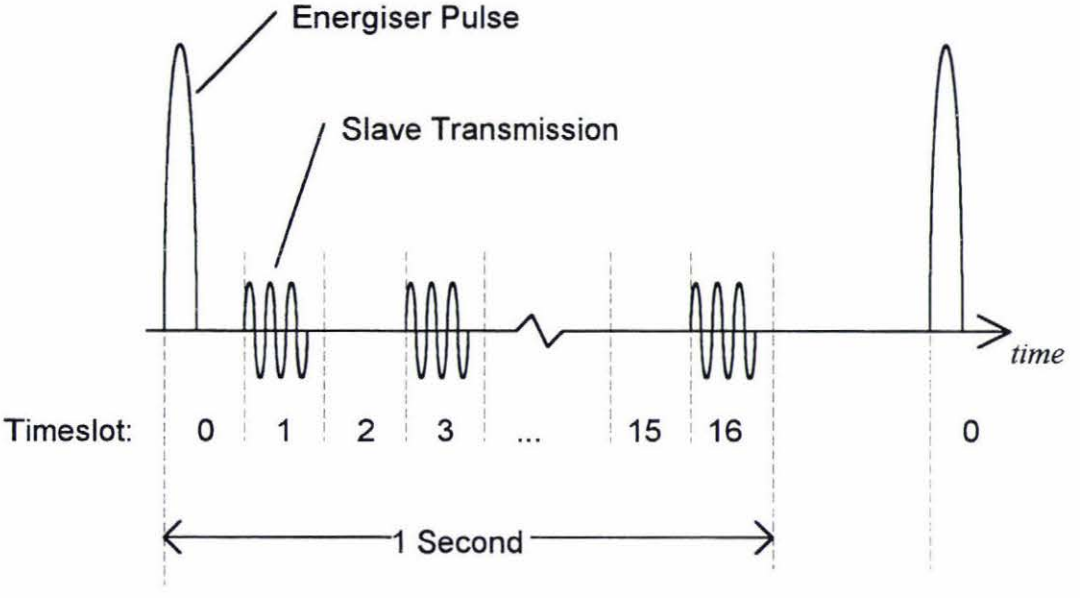


Figure 3.2 Time Division Between Pulses For EFMS

The first timeslot is allocated for the energiser pulse and to the slave units for measurement of the energiser pulse. This ensures that the slave units will not be required to measure the energiser pulse and to transmit at the same time. Following the pulse amplitude measurement each slave unit will have a seventeenth of a second in which to transmit to the master unit.

The time available for the total transmission could be extended if the slave units performed some averaging upon several consecutive pulse voltages, although this would be reducing the response time of the system. Averaging of the fence pulses is also necessary that the drop in fence pulse magnitude from a stock contact does not cause an alarm at the master unit. It is not necessary for the farmer to know the exact second at which the fence fails. Response within a couple of minutes would be still very acceptable.

The maximum voltage legally allowed for a fence pulse is 10kV. It is desirable that the slave unit be able to transmit right down to a voltage at which the stock controlled by the fence no longer receives an appreciable shock from the pulse. For the EFMS, this threshold level value is chosen as 1.5kV. This means that the monitoring system would have to measure the range 1.5kV to 10kV.

The monitoring system should be able to monitor to the end of the fence line (regardless of the distance from the central unit to the end point). Thus if 1.5kV energiser pulse can reach the end point of the fence then the transmission should be able to reach the central unit.

The smallest energiser with which the system should still be expected to run is a mains powered 1.2J energiser. With this energiser the monitoring system would be expected support a couple of slave units. Higher power energisers would mean that more slaves could be used, and that they could be placed further down the fence line from the central monitoring site. The reasons for choosing this energiser as the minimum is that it is the lightest mains powered energiser and is also the smallest energiser that would be expected to control a fence large enough to make a monitoring system such as the EFMS practical.

Speedrite currently use a 12VDC alarm with products such as the Agricultural Fence Monitor (shown in Chapter 2). It is desirable that the same alarm interface with the master unit.

The operation parameters are summarised in Table 3.1.

Table 3.1 Operation Parameters For The EFMS.

Minimum Working Voltage	1.5kV
Maximum Working Voltage	10kV
Minimum Energiser Rating	1.2J
Maximum Number of Slave Units	16
Slave Unit Transmission Time	1/17 Second
Maximum Response Time	2 Minutes
Maximum Slave Distance	End Of Electric Fence
Slave Power Source	Energiser Pulse
Slave Construction	Small, Robust
Master Construction	N/A
Master Power Source	Mains
Master Unit Alarm Output	12VDC

3.2 Modelling

A transmission line model is presented for an electric fence line. By defining the energiser waveform as an input to the model the waveform at the end of the fence was predicted. This model is later used in section 3.3 for an energy case study.

The energiser circuit representing a standard electric fence energiser in chapter 2 is converted to a dynamic model. This model uses ordinary differential equations to describe the circuit. A Ranga-Kuuta method running in Matlab is used to solve the model. Adjustments were made to the components within the model to account for the changing parameters of the circuit during the stresses of operation.

The models are used such that waveforms of the electric fence pulse are obtained for various lengths of the electric fence. These results help to define an operating environment for the electric fence monitoring system.

3.2.1 The Need For Modelling:

The electric fence energiser of today is claimed to be able to send a significantly sized electric pulse over large distances. In some cases these distances may be as large as hundreds of kilometres of wire. The agricultural environment of the Manawatu, (and most of New Zealand) is such that electric fences of such a large construction are very rare. Yet hopes of exporting the EFMS are such that the distances over which the energisers and the EFMS will still work are important to be known. A model of fence energisers and the fence line provides a method of knowing the operating extremes of the fence energisers and the EFMS. In addition an energiser model provides a very helpful design tool for Speedrite International.

In the case of the EFMS the model can initially be used to review the feasibility of line powering the slave units. In future, once exact operating parameters of the system are known, the model can also be used to define the conditions in which the system will continue to operate.

3.2.2 The Electric Fence Model

As mentioned in chapter 1, an electric fence is constructed of one or more unshielded wires running parallel to the ground at various heights. Thus the fence has series resistance and inductance due to the wire itself and the connections between different wires. It also has a capacitance due to electrostatic coupling between the wire and ground. Due to the insulating mounts the shunt resistance is very large. All of these parameters mentioned will vary depending upon the length of the fence giving a transmission line with distributed inductance, capacitance and resistance.

3.2.2.1 Fence Model Circuit

The following parameters per kilometre for traditional electric fences are assumed for use in the model (McCutchan (1980), New Zealand Agricultural Engineering Institute (1986), Studman (1989)).

Table 3.2 Electric Fence Parameters

Parameter	/km
R_L	44Ω
R_G	100Ω
L_W	4mH
C	8nF
G	0

The fence model circuit is constructed using the nominal Π circuit approximation for the transmission line. The circuit is shown in Figure 3.3.

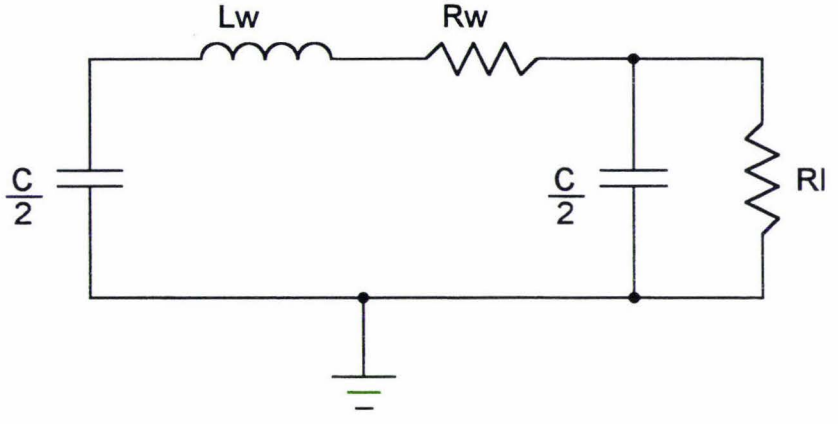


Figure 3.3 Electric Fence Model Circuit

The model uses a transfer function derived from the circuit input to the model. The equation of the transfer function is shown in equation 3.5.

$$G(s) = \frac{R_L}{s^2 \times \frac{L_W R_L C}{2} + s \times \left(L_W + \frac{R_L (R_W + R_G) C}{2} \right) + (R_L + R_W + R_G)} \qquad (3.5)$$

The transfer function is converted to a state space equation and this is used with Matlab routines to simulate the response of the equation to an external input. Appendix 1 shows the Matlab code for this model.

This model of the fence line itself is used in section 3.3 for a case study of the power available to a slave unit.

3.2.2.2 Fence Model Results

Comparisons between simulations of the model and measurements from fences showed that there was a large interaction between the energiser and the fence. This means that the fence model results were only accurate when the energiser input waveform was measured on an energiser that had the same load as was used for the fence model. For different model loads (ie. fence lengths) different energiser input waveforms were required.

3.2.3 The Energiser Model

The large interaction between the energiser and the fence indicates that the fence and energiser should be modelled together. The problem with modelling the energiser is that during operation the stresses upon the transformer are such that it will no longer work in a linear mode. The nonlinearities can be accounted for by making adjustments to the transformer parameters while the simulation is running. In order to construct a model in Matlab that would allow this, a different method had to be used than that used for the fence model. The model was converted to a series of ordinary differential equations and the Runga-Kuuta method was used to solve them. During the simulation the inductance of the transformer was modified according to the current flowing through the transformer to account for the nonlinear effects.

3.2.3.1 Energiser Model Circuit

The circuit used for the energiser and fence model is shown in Figure 3.4. The circuit comprises the standard output circuit for the energiser, with an ideal transformer model, and discrete components added to account for transformer losses (Lowdon (1981), Hughes (1973), Slemon and Straughten (1980)). The fence circuit is the lumped parameter model defined in the fence model (section 3.2.2).

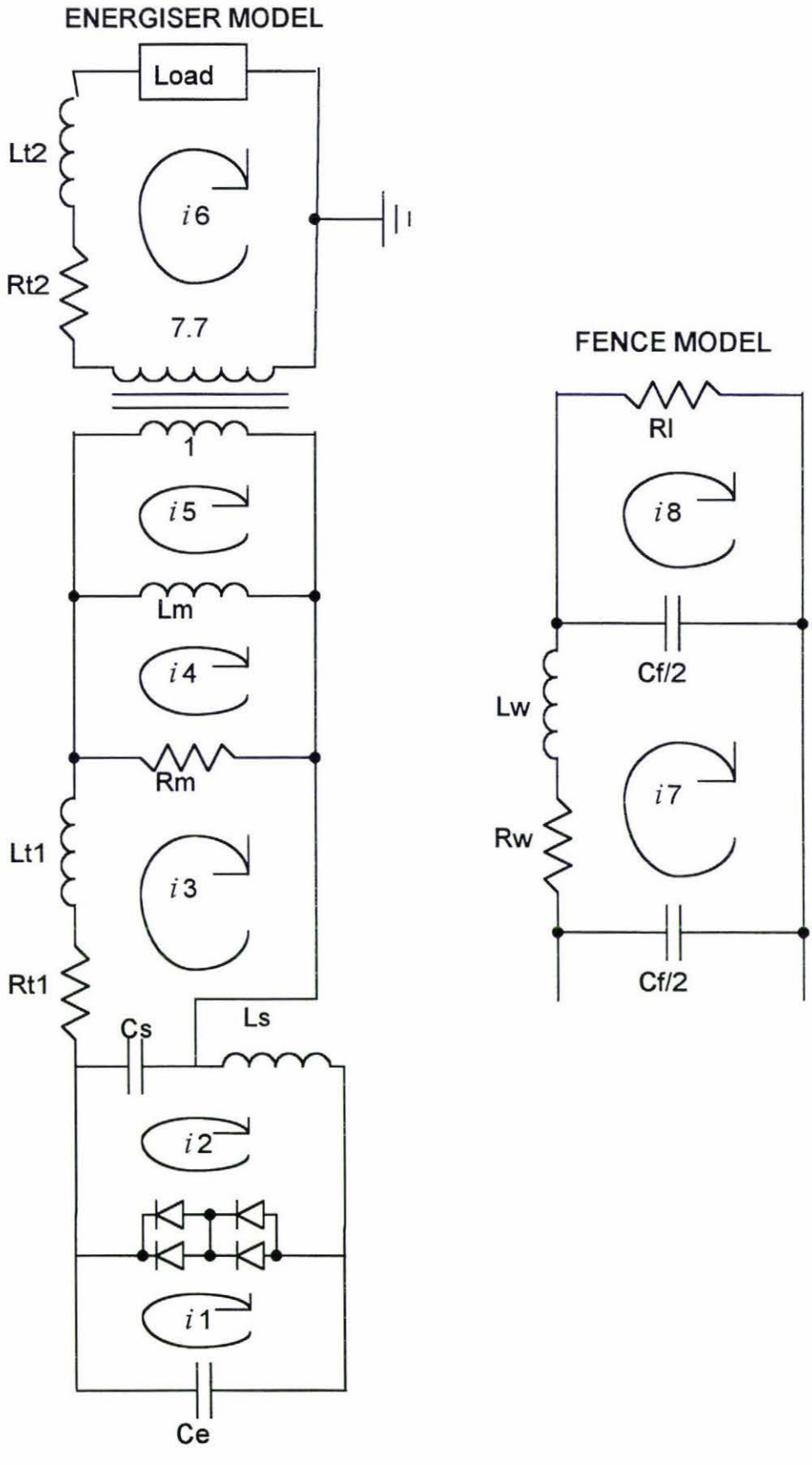


Figure 3.4 Energiser and Electric Fence Model.

where C_e represents the main energiser storage capacitors. C_S and L_S represent the RFI suppression components (to block the high frequency outputs of the energiser), R_{T1} and R_{T2} represent the resistance of the windings of the turns of the transformer, L_{T1} and L_{T2} represent the leakage inductance of the transformer windings, R_M compensates for the core losses (hysteresis and iron loss) of the transformer, L_M represents the magnetising inductance of the transformer, R_W represents the resistance of the electric fence wire, L_W represents the inductance of the electric fence, C_f represents the electric fence's capacitance between the wire and ground, R_g is the resistance of the ground between the end of the fence and the energiser and R_l is the resistance of the load (slave units).

Due to the saturation of the transformer under operation, the magnetising inductance L_M is non-linear due to the variation in μ (Lowdon (1981), Hughes (1973)). For each step in the simulation the inductance was recalculated according to equation 3.6. (Hughes (1973)).

$$L_M = \frac{N^2 AG}{\frac{l_{CORE}}{\mu_{CORE}} + \frac{l_{GAP}}{\mu_{AIR}}} \tag{3.6}$$

where μ_{CORE} is the absolute permeability of the core, μ_{AIR} is the absolute permeability of air, A is the cross-sectional area of the core, N is the number of turns in the primary, G is a gap factor, l_{CORE} is the length of the magnetic circuit and l_{AIR} is the length of the air gap. The gap factor accounts for the non-magnetic insulation between the laminations which reduce the effective cross-sectional area of the core. In order to calculate μ_{CORE} the core lamination characteristics were digitised, and μ_{CORE} was found by calculating H (equation 3.7) (Hughes (1973), Slemon and Straughten (1980)) and referencing a lookup table.

$$H = \frac{Ni}{l_{CORE}} \tag{3.7}$$

where H is the magnetic field strength and i is the magnetising current.

For calibration of the energiser part of the model, the load was replaced with a resistor and comparisons made between the model results and the known energiser output into a resistive load. This enabled the setting of the core loss, gap factor and gap length. Note that the core loss is frequency dependant (Lowdon (1981)), and thus for different energiser circuits using the same transformer, the model must be recalibrated. The gap factor is dependant upon the construction of the transformer and the insulation of the laminations and not the additional energiser circuitry. Thus for each transformer the Gap factor must be recalibrated. Once the unknown quantities of the model are evaluated, the resistive load was replaced by the fence model.

The equations generated from the energiser circuit without the fence model attached are shown in equations 3.6 to 3.11. Appendix 2 shows the code for the energiser model.

$$\frac{dV_{C_e}}{dt} = \frac{-i_2 + I_S(e^{\frac{-V_{C_e}}{V_T}} - 1)}{C_e} \tag{3.6}$$

$$\frac{di_2}{dt} = \frac{V_{C_e} - V_{C_s}}{L_S} \tag{3.7}$$

$$\frac{dV_{C_s}}{dt} = \frac{i_2 - i_3}{C_S} \tag{3.8}$$

$$\frac{di_3}{dt} = \frac{V_{C_s} - i_3(R_{T1} + R_m) + i_4 R_m}{L_{T1}} \tag{3.9}$$

$$\frac{di_4}{dt} = \frac{(i_3 - i_4)R_m}{L_m} - \frac{N^2 R_m(i_3 - i_4) - N i_6(R_{T2} + R_L)}{L_{T2}} \tag{3.10}$$

$$\frac{di_6}{dt} = \frac{N R_m(i_3 - i_4) - N i_6(R_{T2} + R_L)}{L_{T2}} \tag{3.11}$$

When the fence model is included with the energiser model the equations for use in Matlab are shown in equations 3.12 to 3.20. Appendix 2 shows the code for the energiser and fence model.

$$\frac{dV_{C_e}}{dt} = \frac{-i_2 + I_S(e^{\frac{-V_{C_e}}{V_T}} - 1)}{C_e} \tag{3.12}$$

$$\frac{di_2}{dt} = \frac{V_{C_e} - V_{C_s}}{L_S} \tag{3.13}$$

$$\frac{dV_{C_s}}{dt} = \frac{i_2 - i_3}{C_S} \tag{3.14}$$

$$\frac{di_3}{dt} = \frac{V_{C_s} - i_3(R_{T1} + R_m) + i_4 R_m}{L_{T1}} \tag{3.15}$$

$$\frac{di_4}{dt} = \frac{(L_{T2} - N^2 L_m)(i_3 - i_4)R_m}{L_{T2} L_m} + \frac{N(i_6 R_{T2} + V_{C_1})}{L_{T2}} \tag{3.16}$$

$$\frac{di_6}{dt} = \left(\frac{1}{L_{T2} - N^2 L_m}\right)\left(N L_m \frac{di_4}{dt} - i_6 R_{T2} - V_{C_1}\right) \tag{3.17}$$

$$\frac{dV_{C_1}}{dt} = \frac{i_6 - i_7}{C_1} \tag{3.18}$$

$$\frac{di_7}{dt} = \frac{V_{C_1} - V_{C_2} - i_7(R_W + R_G)}{L_W} \tag{3.19}$$

$$\frac{dV_{C_1}}{dt} = \frac{i_6 - \frac{V_{C_2}}{R_L}}{C_2} \tag{3.20}$$

3.2.3.2 Energiser Model Output

Calibration was required of the model in setting the core loss resistance and gap factor.

The output measured from a SM5800 Speedrite Energiser is shown in Figure 3.5.

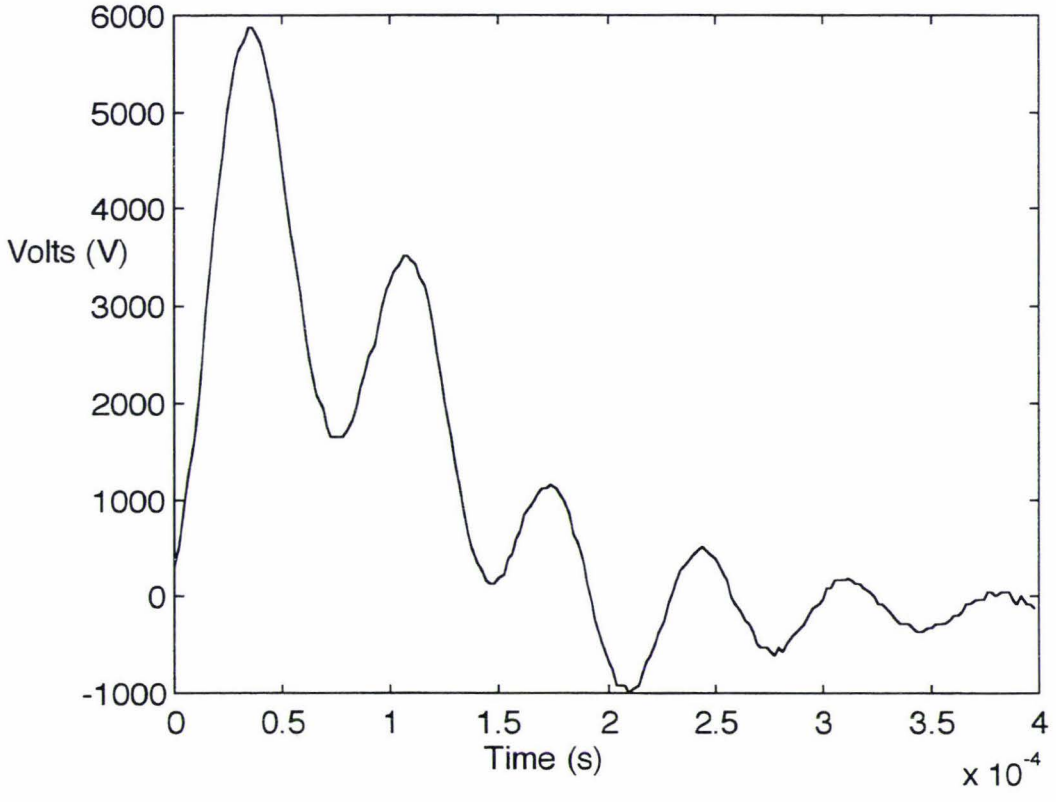


Figure 3.5 SM5800 Speedrite Energiser Output.

An example of the model output is shown below, after calibration, in Figure 3.6

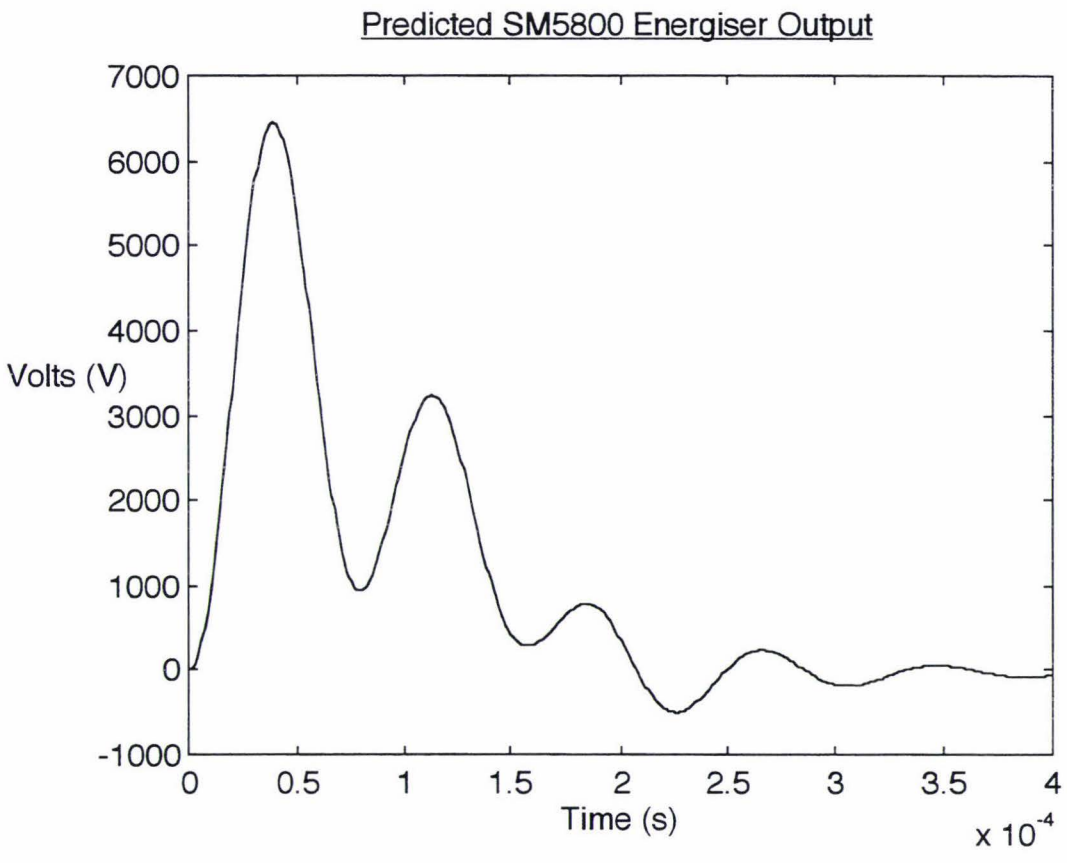


Figure 3.6 Energiser Model Output

The model results show that the frequency of the suppression of the RFI content of the output waveform is correct. The duration of the pulse is also accurate and the general shape of the model output is accurate.

3.2.3.3 Uses Of The Energiser Model

The model has already been used to determine the energy available to a slave unit as shown in section 3.3.

In future the model could be used to determine the maximum operating range of systems such as the EFMS and to determine the effectiveness of the energiser over large lengths or unusual configurations of wire.

The model shows that there are some factors which are critical to the efficiency of the transformer and thus the energiser. For example the model shows that during operation the transformer is heavily saturated, and an increase in inductance of the primary can avoid this and make the energiser more efficient. This increase could be accomplished through a larger core cross-sectional area, or shorter magnetic path length. Increasing the G factor constant by improving the connection between the core will reduce the variation between energiser outputs, and provide a more efficient energiser. The use of thinner laminations may increase R_M and reduce the power lost inside the core of the transformer. Adjustments in these parameters can be trialed before construction to speed up the design process.

The model can be extended to other energisers available from Speedrite with the same output circuitry configuration. This requires a few measurements of the output transformer, and changes to component parameters in the output circuit and recalibration of the model.

The model will provide a very useful design tool for Speedrite International.

3.3 Case Study Analysis

The following case study is intended to define working parameters for a practical monitoring system. The nominal Π circuit approximation discussed in chapter 1 is used to provide a model for the fence the errors associated with this circuit are discussed in section 3.3.1. The energiser and fence model is then used to provide an estimate of the power available to the slave units in section 3.3.2. The power consumption of the slave unit is discussed in section 3.3.3. Section 3.3.4 uses the fence model to predict the relative effects of slave unit transmissions to the master unit. The issues highlighted by this case study are summarised in section 3.3.5.

3.3.1 Nominal Π Circuit Approximation.

For a 10km line, with an energiser pulse with bulk frequency of 2.5kHz (ie half-period of 200μs), ground resistance of 200Ω, and the line characteristics the same as those mentioned in section 3.2.2.1, can be calculated and are shown in Table 3.3. These show the error involved in applying the nominal Π lumped parameter model circuit instead of the equivalent Π lumped parameter model circuit.

Table 3.3 Nominal Π Circuit Approximation

	Z	Y/2
Nominal Π Model	$896\angle 44°$	$4\times10^{-8}\angle 90°$
Equivalent Π Model	$787\angle 53°$	$6.71\times10^{-4}\angle 86°$

The differences above are not large enough to cause a problem for the case study. In the more dominant series impedance arm there is less than a 15% difference. Thus the nominal Π circuit is used for the rest of this case study understanding that it will not be exact, yet will give approximately correct results. Note that there is little to be gained from applying the equivalent Π circuit as changes in fence configurations would have a greater effect than the errors introduced by lumped parameter model approximations.

3.3.2 Energiser To Slave Line Simulation

The energiser and fence model with the fence characteristics mentioned above were used to show the energy at 10km from the 5.8J energiser available to a 5kΩ resistive load. The simulation results are shown in Figure 3.7.

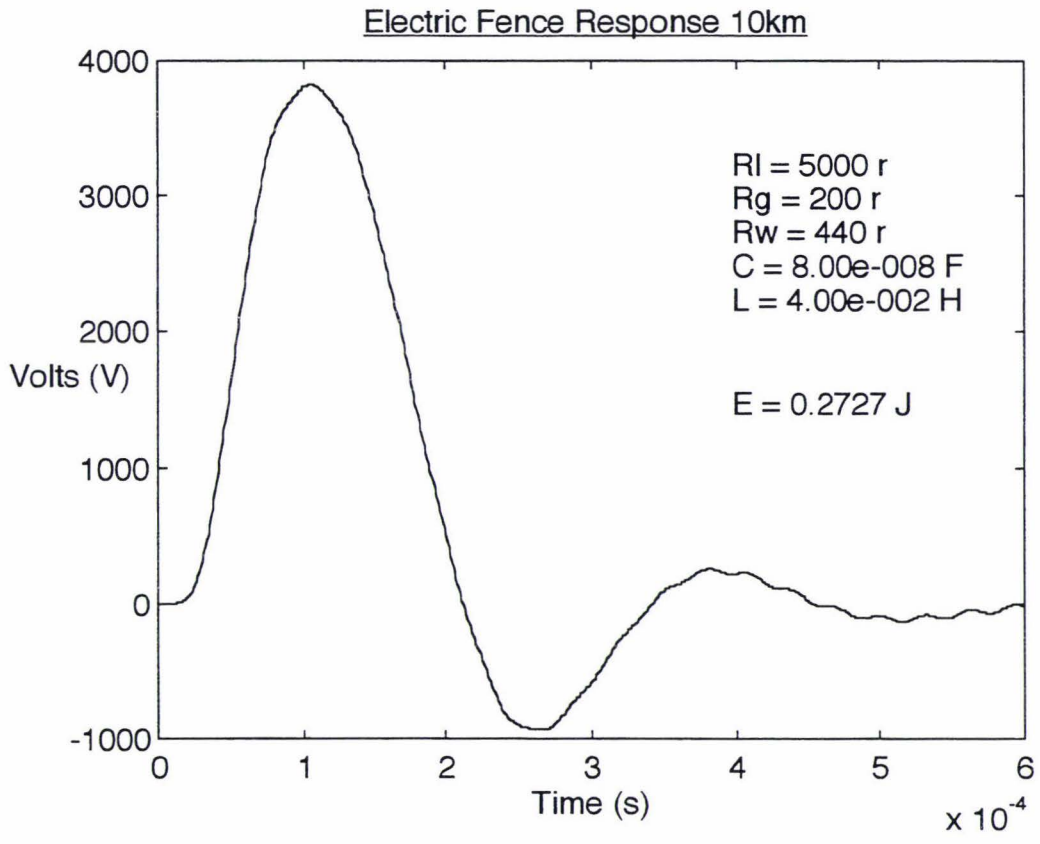


Figure 3.7 Energiser, No Fault Conditions at Device

The simulation was then repeated to give a peak output voltage of 1.5kV by introducing a fault resistance. This was done to give the worst case operating conditions for the slave unit. The total load resistance in this case was 500Ω. This included the two resistances, the first, the impedance of the sixteen slaves, totalling 5kΩ, the second the fault resistance of 556Ω. The results of this simulation are shown in Figure 3.8.

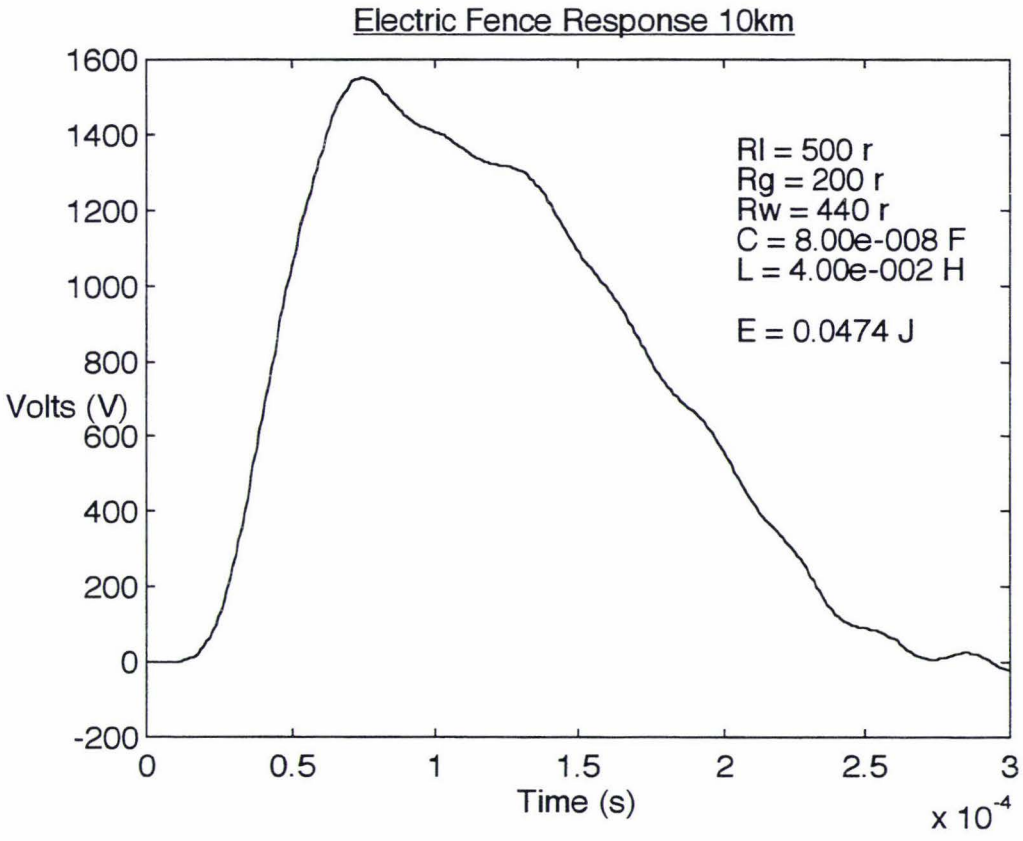


Figure 3.8 Energiser Over 10km Fence, With Fault.

The energy in this output waveform available to the sixteen slaves is that absorbed in the 5kΩ resistor. This was calculated using equation 3.21.

$$E = \frac{1}{n}\int \frac{V^2}{5k\Omega} \tag{3.21}$$

where n is the number of slaves. This leaves 3mJ per slave.

3.3.3 Slave Unit Power Consumption

If at best the slave units could convert the 3mJ at an efficiency rate of approx 50%, thus this would leave 1.5mJ per slave per pulse.

As the pulse repeats once every second, the energy available to the slave would equate to the power available to the slave. For a slave running on a 5V supply, this corresponds to a current of 0.0015/5=0.3mA.

If the slave contained a CMOS microprocessor and support circuitry, running at a slow speed (of 32kHz say) to perform its control tasks, this would take approx 0.2mA. In energy terms this corresponds to 0.2×5=1mJ leaving just 0.5mJ for the transmission back to the master unit.

In a system with up to 16 slaves a transmission should be of duration up to 1/17th of a second so that the other 15 slaves would also have an opportunity to transmit. If it is assumed that to transmit the energy there is another 50% energy conversion efficiency drop, this gives a transmission energy of Et=0.25mJ.

3.3.4 Slave To Master Line Simulation

For a transmission time (T) of 0.0588s the peak voltage (V_P) that could be placed upon the fence can be calculated, for a rectangular pulse, if it is assumed that the load is the combination of the fault resistance, the fence and the energiser load. For this case study the load (R) is 273Ω. The calculation is shown in equation 3.22.

$$V_p = \sqrt{\frac{E_T \times R}{T}} \tag{3.22}$$

This gives a V_P = 1.08V. This 1.08V pulse was again simulated using the Matlab program, with results shown in Figure 3.9 that the bulk frequency of the signal is much lower than that of the energiser pulse, thus the fence filtering characteristics have little effect on the waveform. There is attenuation due to the voltage divider formed by the energiser load and the wire resistance.

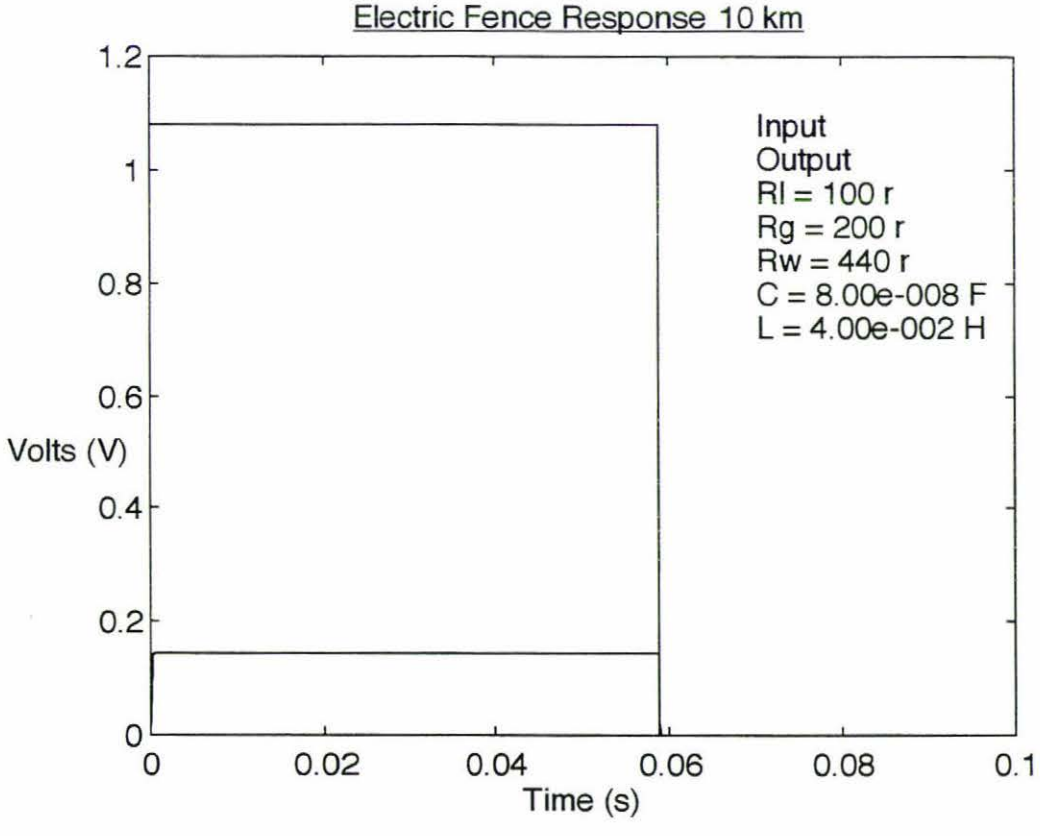


Figure 3.9 Pulse Reply

It should be possible to detect this pulse at the master unit 10km away. However, noise pickup is likely to be several times larger in magnitude than the pulse. It would be more robust if a frequency burst was transmitted instead of a pulse. This would allow filtering to be employed for the frequency detection method. It is desirable that this frequency be as low as possible, to reduce the attenuation effects of the fence, but high enough to be distinguishable from 50Hz mains and harmonics. For this case study a frequency of 273Hz is chosen (Chapter 4 contains a discussion of the frequency selection).

Now the energy for transmission becomes:

$$E = \int \frac{V_p^2 \sin^2(2\pi ft)}{R} dt \tag{3.23}$$

$$= V_p^2 \frac{4\pi ft - \sin(4\pi ft)}{8\pi fR} \tag{3.24}$$

After substituting in the equation parameters, and integrating over the interval 0 to 0.0588s, Vp=1.53V.

Again this waveform can be simulated in Matlab to show the fence response.

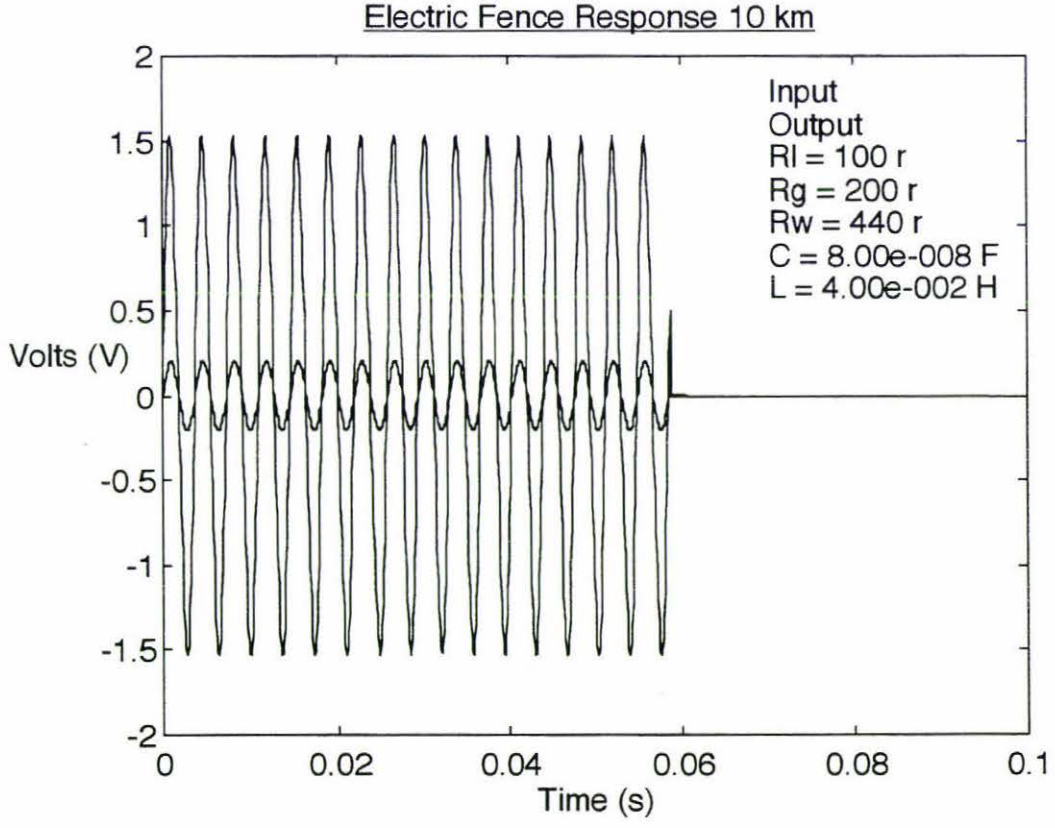


Figure 3.10 Frequency Burst Reply

As Figure 3.10 shows there is again attenuation caused by the voltage divider but there is no discernible attenuation from the low pass filter effects of the fence line.

This can be verified by repeating the simulation with a very large terminating resistance.

3.3.5 Case Study Conclusions

The case study highlights several problems with the proposed ideal fence monitoring system.

Given the limitations above the system is able to remain operating under fault conditions where the energiser pulse peak voltage drops down to 1.5kV.

The first is that the slave unit is definitely starved for energy, with only 0.25mJ available for transmission. This may be further reduced with movements in the assumed parameters.

The transmission may be of a very much lower frequency than the frequencies in the energiser pulse itself and the fence will have far less effect on the transmission signal than on the energiser pulse. This implies that should the energiser pulse reach the slave unit the slave transmission should be able to reply effectively.

Although not mentioned in the case study, it is possible for the device to store the transmission energy and only transmit every n pulses, and therefore be able to use n times as much power for the transmission. Alternatively the operation of the measurement devices could be improved if a transmission algorithm was devised such when there was a lot of energy available the slave transmitted often, and when there was little energy available the slave unit transmitted rarely. Thus the energy used to retransmit would be proportional to the energy available to the slave unit.

3.4 Methods Of Communication

The communication is concerned with reporting the peak voltage on the fence at the slave back to the originating point.

There are three main criteria which the method chosen for transmission must adhere to in order to be ideal for this application. The first is noise immunity. The long fences are susceptible to noise pickup from overhead power distribution wires, other electric fences, lightning etc. The second is the time response of the system. Obviously it is preferable that the communication method chosen respond fast in proportion to the effects of a fence failure. The third, and possibly most important, is the energy efficiency of the chosen method. The power available to the transmitter is very small and related to the voltage sensed by the unit. It is desirable that the unit use as little power as possible, especially when the fence voltage is low.

Two methods of coding for transmission are presented below. The first is the traditional quantise and code method (Haykin (1989), Malmstadt et al (1981)). The second is similar to that of a charge balanced A to D converter and gives output pulses in proportion to the voltage measured (and therefore related to the energy available). This method is called the Pulse Density method. An improvement to the Pulse Density method is also presented.

3.4.1 Traditional Approach

This method is based upon quantising the input signal then converting to a multi-bit binary code ready for transmission. If a greater accuracy is required, then the system need only use a greater number of bits to quantise and code the input signal. This method lends itself to high bit rates in order to transmit the multi-bit data, to minimum packet sizes which encode a single measurement, and to various protocols which govern the transmission.

The slave unit must be able to measure the voltage and quantise it to a digital signal itself. It will then take a time dependant upon the accuracy of the quantisation (and the bit rate) to send the signal back along the fence line.

There are several advantages to this method. The first, and perhaps main advantage, is versatility. It can easily be expanded by increasing the complexity of the receiver and transmitter to cope with various extensions. A higher level protocol would enable error detection and the communication of other data. It may even be desirable to in the future add to the communication the measurement of water troughs heights, control signals for automated gates or irrigation systems etc.

Other advantages are that the quantisation algorithm is independent of the transmission and need not be linear. For example using 4 bit accuracy, 16 different voltage levels could be found, however they could be placed at levels such that the probability of a measurement in each level is the same, thus reducing the redundancy in the transmission and maximising the information transfer. Or if greater accuracy is

required in one range of the measurement then more levels could be placed around this area. Figure 3.11 shows an example with a concentration of voltage levels at lower voltages.

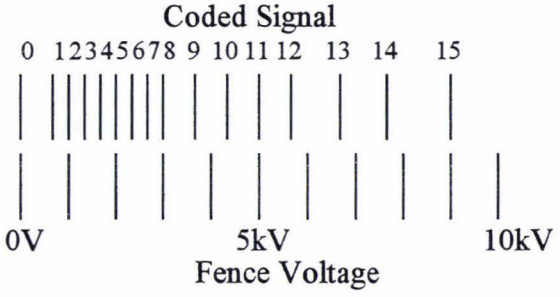


Figure 3.11 Non-linear Quantisation

One of the main disadvantages of this method is the complexity that would be required for the receiver and transmitter units. The measurement device must be able to both quantise the signal, and to convert this quantised signal to a binary code ready for transmission.

For the response time of the system, this method is adequate. The system can be made to give an arbitrary response time and accuracy if the bit rate is fast enough. For example with 16 signalling levels four bits are required for every transmitting device. If there are 16 monitoring devices, 16x4 = 64 bits per second (bps) must be transmitted. If a slower bit rate is necessary, the monitoring devices could perform some averaging on a series of pulses and use a series of pulses to transmit only the averaged measurement. However, now the energiser pulse itself is no longer available for the synchronisation of the units and a start code, or unique start bit is required to signal the start of a transmission.

In relation to the energy requirement for transmission, this method does not preform well. The higher the accuracy of measurement required, the more bits the unit must transmit and the more energy required. For example if a transmission for a value of half of the maximum is required, then all but one of the bits will be a logic '1', and there will be a large energy consumption, when only half of the maximum energy is available. It is likely that there are cases where the unit does not have enough energy to transmit the required signal.

3.4.2 Pulse Density Method

This method relies on the slave unit only transmitting when it has enough energy to do so. Thus it will continually charge itself up from the fence line until it has enough energy to transmit a signal. This overcomes the main disadvantage of the traditional method where the unit may not have enough energy to transmit.

The Pulse Density method relies on the slave unit to sum up the peak voltages until a threshold value (the point at which the device should have enough energy to transmit)

is reached. When the threshold value is reached, a single bit is transmitted from the slave unit.

The decoding device, at the master unit, can then use the knowledge of how many times the slave responded within a given number of pulses to determine what the peak voltage on the fence is. In order to do this successfully a careful distinction must be made. The desirable measurement is the peak voltage of the fence pulse, and not the energy present in the fence pulse. The slave unit must respond with pulse density related to the summation of the peak voltages and not the summation of the energy.

In theory the energy in the fence pulse is proportional to the voltage squared which gives an abundance of energy at high voltages. In practice however, due to the changing pulse shape, E tends to be proportional to V over the operating range.

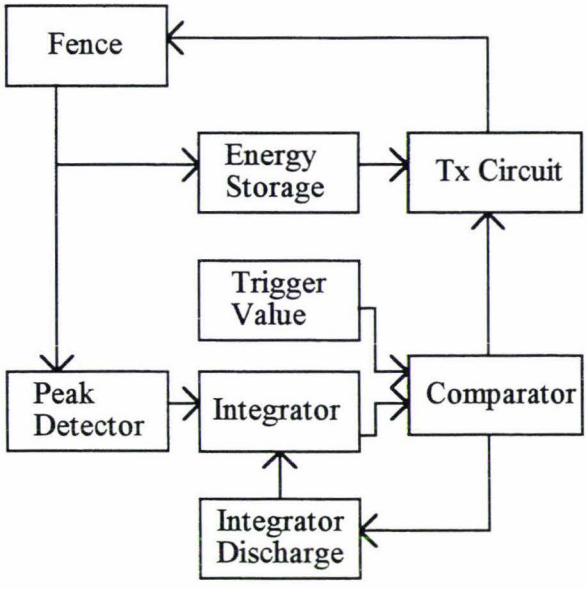


Figure 3.12 Pulse Density Method Transmission

Figure 3.12 shows the function of the method. The peak detector holds the peak pulse voltage. This voltage (or a scaled version of it) is added to the peak voltage sum in the integrator. At the next incoming pulse the integrator voltage is compared to the threshold level voltage. If higher the comparator then triggers the integrator discharge, in order to subtract the threshold voltage level from the integrator and advises the transmission circuitry to transmit a 1 bit response back to the energiser.

If this constant value subtraction was not present, and instead the integrator was reset, the decoder would lose resolution when there is a high response rate. For example consider a measured voltage of 75% of the threshold level. Without the constant value subtraction the slave unit transmits once every two pulses with the decoder giving an output of only 50% of the threshold voltage. With the constant value subtraction the slave will transmit after two pulses, and then after one pulse with the decoder giving an output of 75% of the threshold voltage.

The decoder can multiply the threshold voltage by the number of positive responses and divide by the number of energiser pulses in order to obtain the average voltage on the fence during the summation period.

An example of the operation of the Pulse Density method is shown in Figure 3.13.

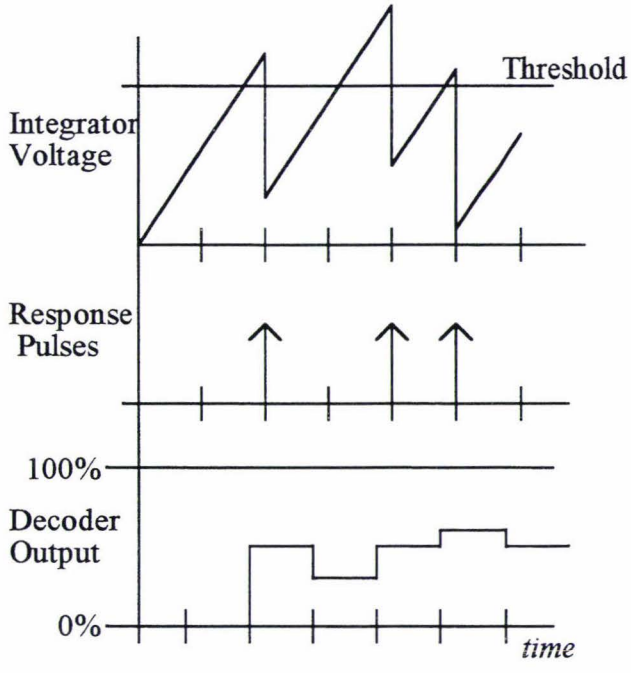


Figure 3.13 Pulse Density Method

In Figure 3.13 it can be seen that the slave transmitted 3 times in 6 pulses and giving a decoder output of 50%. The actual voltage in the above diagram is 60% of the threshold voltage which the decoder will output as more pulses are viewed by the decoder.

This method has the advantage in that it is simple and output responses are related to the energy available. However, as can be seen from above there is a trade off for response time against accuracy.

Consider that an output was determined for five pulses. In this case if the fence voltage is less than 20% of the threshold value, after five pulses the measurement device will not have responded. The decoder will output 0% of the threshold value, while as mentioned above the voltage could be as high as 20% of the threshold value. This corresponds to a maximum error of 20%. Generalising for the case of n pulses used to determine a decoder output the maximum error for the PD method is 1/n.

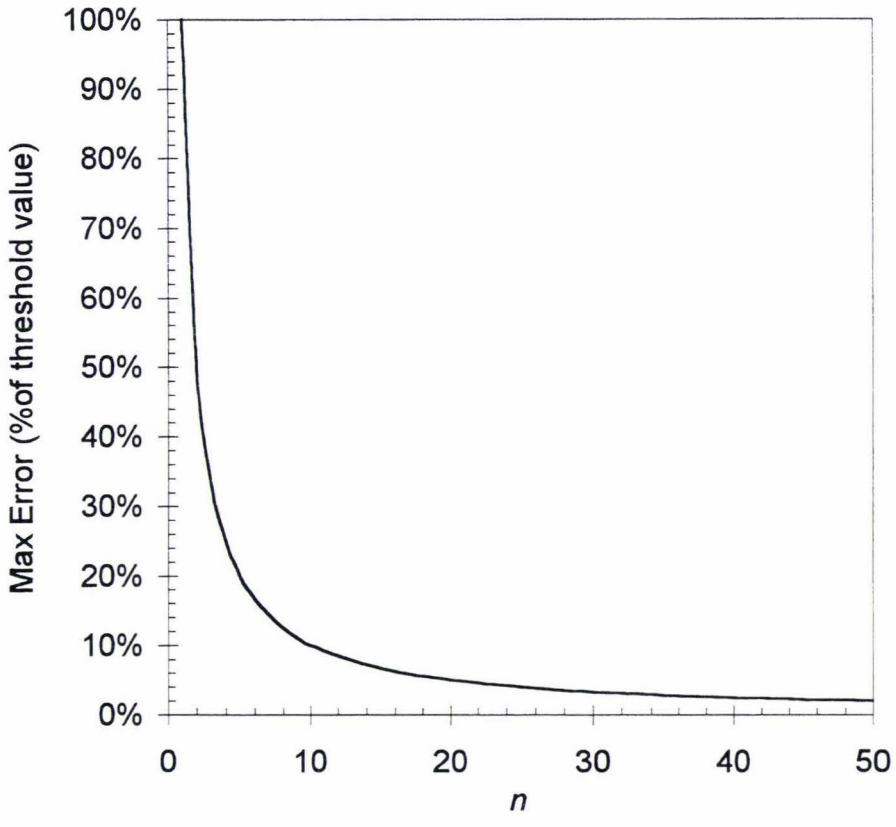


Figure 3.14 Max PD error vs Number of Pulses Tested

In order to increase the accuracy of the Pulse Density method, n should be as large as possible. Also in order to minimise the absolute error, the threshold voltage should be as low as possible provided that for the maximum peak voltage the device will not respond every pulse.

3.4.3 Improved Pulse Density

In the Pulse Density method if n is not large compared to the average time between two responses then the decoder output will be seen to fluctuate wildly around the correct value. For example consider the last 10 pulses tested (n=10), a threshold value of 50kV and a average peak fence voltage of 7kV. For any 10 pulses there could be either 1 or 2 responses indicating either 10% (5kV) or 20%(10kV) of the threshold value. Should the fence voltage drop below 5kV the decoder would even respond with 0kV. These errors are still a maximum fraction of 1/n of the threshold value but an improvement can be made.

The improvement is made if the pulse count is started directly after transmission from the slave. The count is terminated on a pulse in which there was a slave transmission, even if this is not the last pulse tested. This energiser pulse count (m) is less than or equal to the number of pulses tested (n). The decoded output is then found by dividing the number of positive slave transmissions (p) by m and multiplying by the threshold value.

In the previous example, with a threshold value of 50kV and n=10, the gap between the two successive pulses (m) would be either 7 or 8 pulses, thus if there was a pulse in the slot just before the receiver started counting, the gaps after the last response would be ignored. Thus values of m=7 or m=8 would be used, giving decoded values of 7.1kV or 6.25kV respectively. For a large n the decoder output would tend towards a voltage of 7kV.

This improvement means now that the maximum error occurs when the pulse response rate is maximum and this is 1/n, however if the response rate is lower than this the error is reduced significantly when compared to the Pulse Density method. If p is the number of positive responses for m energiser pulses then the maximum error is shown in equation 3.25.

$$\text{maximum error} = \frac{p}{m-1} - \frac{p}{m} \tag{3.25}$$

Figure 3.15 shows an example of this error reduction. Note that error for the 100% transmission rate would indicate a voltage greater than the threshold value. Thus to ensure complete confidence in the maximum error the threshold value should be set higher than the maximum electric fence voltage level. The dotted line represents the 1/n and previous maximum error. The solid lines show the errors, now dependant upon the number of positive responses, for the improved pulse density method. The numbers indicate p (positive responses).

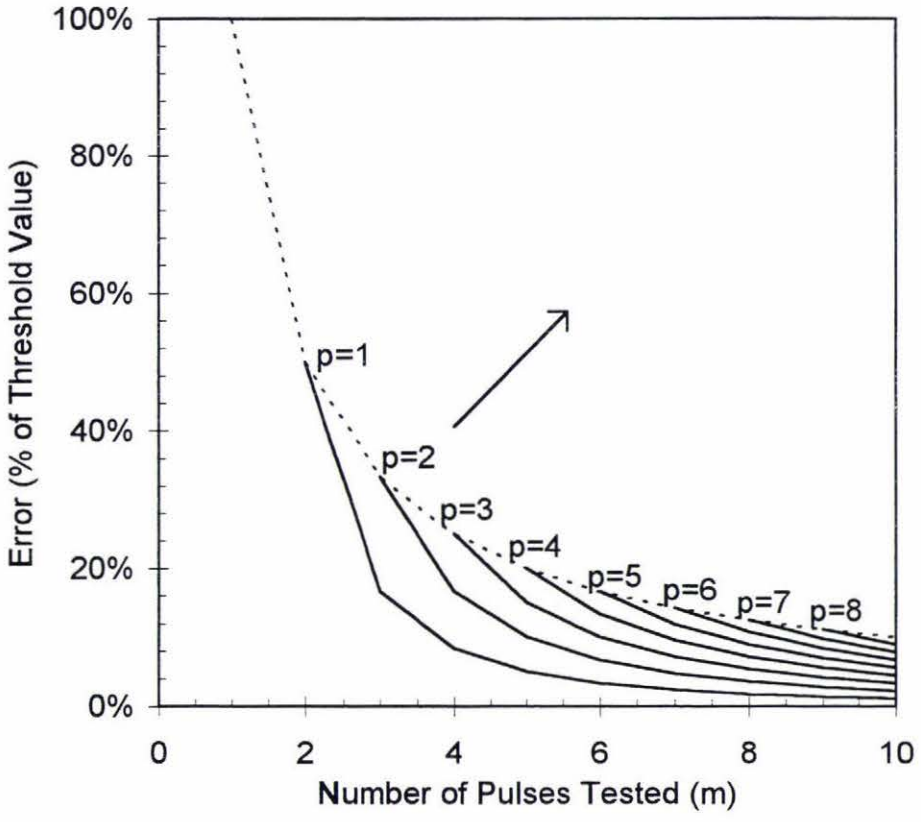


Figure 3.15 Improved PD Errors

This improvement now means that the error is no longer a percentage of the threshold value, but rather a percentage of the decoder output. For example, in the case where m=n=10, the error becomes 11% (1/9th) of the decoder output, rather than 10% of the threshold value. Thus when the voltage is 10%, the error has improved from 10% of the threshold value to 1.1% of the threshold value. For the case when m<n, the error increases as m decreases, however m will never be less n/2, thus the improved method will never respond with an error worse than the Pulse Density method.

Although this method is improved a corresponding improvement to the decoder is required. Now the decoder must be able to divide by a variable number for m, as m could range from n down to the n/2. Also the decoder must now start counting only after a pulse is received.

Chapter 4
The Slave Unit

The primary function of the slave unit is to measure the peak energiser fence voltage at the slave unit and transmit this back to the master unit. To provide this function within a practical device, power supply and power consumption, voltage measurement techniques and transmission techniques are the key areas of focus. Alongside the technology is the issue of cost. The slave is a consumer product and costs must be minimised.

A small amount of power is available from the fence pulse itself. If the slave unit can derive power from the pulse, the need for an external power source or a battery is eliminated. This is essential if the slave unit is to be a reliable and low maintenance unit for the user. Section 4.1 of this chapter outlines the power supply for the slave unit. This section details the design of an efficient power supply and the construction of the components used in this design.

The slave unit requires a measurement system to measure the peak fence voltage to help determine when a transmission reply is required. Section 4.2 details alternatives for this measurement system and rates them according to the power efficiency, accuracy and cost design criteria.

Section 4.3 discusses the circuit used to transmit onto the fence line. The circuit must sufficiently couple the transmission reply onto the fence line, and be protected against energiser pulses. This section also presents a Matlab model that is used to model the reply from the slave onto the fence line. Chapter 3 outlines the actual transmission algorithm used by the slave.

Sections 4.4 to 4.6 discuss issues such as energy efficiency in microprocessor circuits, the microprocessor power up, circuit board layout and slave unit packaging.

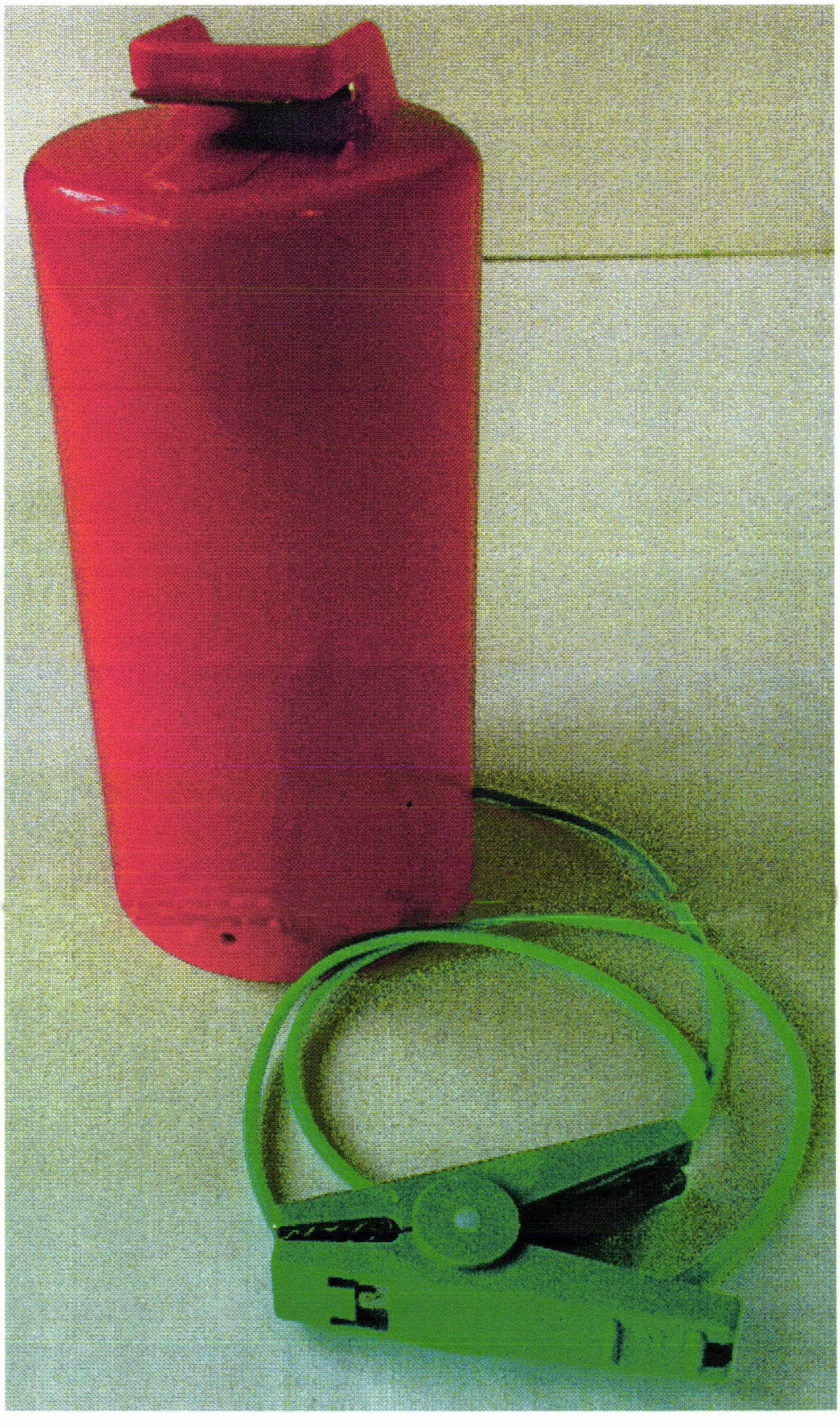


Figure 4.1 The Slave Unit

4.1 The Power Supply:

This section details the construction of the slave unit power supply. The supply derives power from the fence pulse. Options for the power supply are first discussed. In the final design the transformer used is shown to be a major loss element. Methods for improving the transformer, such as gapping the core, using grain orientated laminations and changing the number of turns in the transformer are considered. Experimental results presented for these options. The final transformer specification is also presented.

4.1.1 Power Supply Requirements

The power supply must take enough power from the fence pulse to power the slave unit but leave the pulse largely unaltered to power other slave units and still deter livestock (the essential function of an electric fence). Due to the small amount of power available, the power supply itself must be as efficient as possible.

4.1.2 Circuit Alternatives

As the fence pulse is narrow an efficient power supply is difficult. In a simple circuit energy would be transferred by a high current for the pulse duration which gives high I^2R and other losses. These losses can be reduced if the current transfer time can be increased and the current peak correspondingly decreased. Traditionally this is accomplished through the use of a flyback transformer as shown in Figure 4.2 (Billings (1989)). In this method the energy is stored by a low current transfer in the high voltage high inductance primary of the transformer while the pulse is present. No current flows in the secondary at this time because of the blocking diode. After the pulse the stored magnetic energy forces the transformer to 'flyback' and the energy is transferred out of the low voltage low inductance secondary as a low current over an extended time. The FET switch prevents energy being transferred out of the primary during the "flyback" phase.

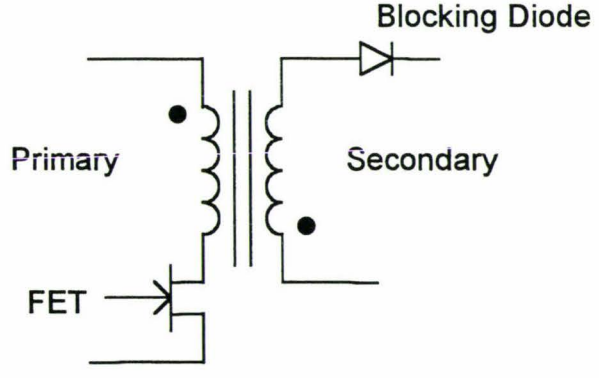


Figure 4.2 Flyback Transformer

In this application the traditional flyback transformer would need a FET that could block 10kV and thus the traditional approach is not possible. A flyback style

operation would be possible if a step-down transformer was used such that the FET would no longer have to block 10kV, however this would uneconomic and would still contain a transformer in forward converter mode. Although the traditional flyback circuit is not possible, the same principles can be used to design an efficient power supply. The power supply should contain a series inductor in order to slow the energy transfer down and to set the minimum impedance of the slave unit. With the addition of a forward converter centre tapped transformer the circuit can be made bipolar. Figure 4.3 shows the circuit for the power supply:

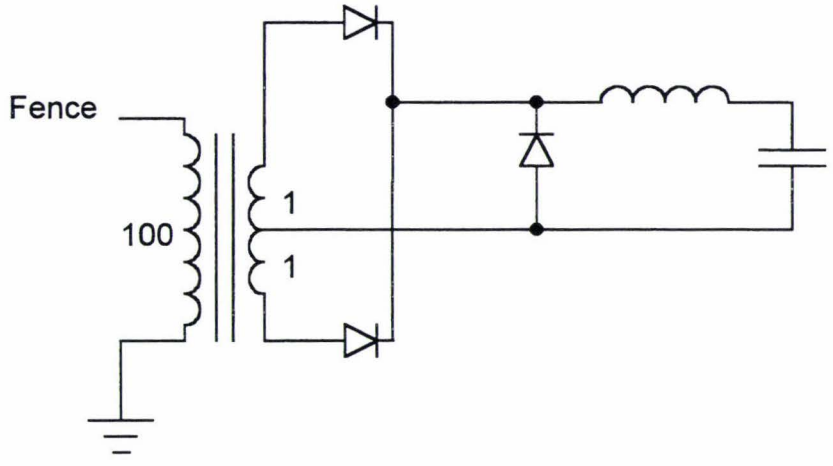


Figure 4.3 Slave Unit Power Supply

A large stepdown transformer is necessary to step the voltage down to a level where common non-expensive components can be used. A capacitor is used as a main storage element for the slave device, yet if this was directly connected to the fence it would take a lot of energy out of the pulse, at a very large current, thus an inductor is used to temporarily store the energy for the slave unit. Initially the inductor will present an infinite impedance to the fence, thus not affecting the pulse at all. As the inductor starts to permit current to flow, it stores energy itself, and starts to charge up the storage capacitor(s). Thus the longer the pulse lasts the lower the impedance the slave unit presents to the fence, and the more energy that the slave unit takes. When the pulse disappears the diode from ground to the inductor forms a current path to allow the inductor to keep the current flowing in the circuit, without going through the transformer, and release all of its energy into the storage capacitor(s) and not back onto the fence.

The step-down transformer also allows ease of measurement of the pulse for the measurement circuit.

4.1.3 Chosen Power Supply Method

Figure 4.3 shows the power supply circuit. With the correct selection of values, the slave unit will not affect the start of the fence pulse, but the current through the slave will ramp up (and the fence voltage drop off), the longer the pulse lasts, and thus provides more energy for the unit. The diodes should all be at least 200V diodes for blocking the stepped down 10kV pulse when required.

The capacitor values are chosen such that the voltage drop between pulses, is not enough to bring the voltage level below that of the minimum working voltage of the microprocessor. For a maximum drop of 0.2V and a maximum current of 2mA the capacitor value is:

$$\begin{aligned} C &= \frac{I\tau}{\Delta V} \\ &= \frac{0.002 \times 1}{0.2} \\ &= 10000\mu F \end{aligned} \tag{4.1}$$

This can be achieved with two 4700μF capacitors.

The inductor calculations can be performed using the minimum impedance that the slave network is to present to the fence. It is desired that the network of 16 slave units represent a 5kΩ load after 100μs of a 10kV pulse. Hence

$$\begin{aligned} Z_{1slave} &= 5000 \times 16 \\ &= 80k\Omega \end{aligned} \tag{4.2}$$

$$\begin{aligned} I &= \frac{V}{Z} \\ &= \frac{10000}{80000} \\ &= 0.125A \end{aligned} \tag{4.3}$$

If the current at the fence is 0.125A the current out of the 100:1 transformer will be 12.5A (after 100μs of a 10kV pulse). This gives an inductance of:

$$\begin{aligned} L &= \frac{Vt}{I} \\ &= \frac{100 \times 100 \times 10^{-6}}{12.5} \\ &= 0.8mH \end{aligned} \tag{4.4}$$

This is the minimum inductance required if the desired impedance is to be met. This inductor was made with an air core to ensure that it would not saturate at the high currents, and that the weight of the slave unit could be kept as low as possible. A "9/16 inch" square bobbin was selected for the inductor as this is the largest that would fit in the slave unit. This gives a maximum winding window thus minimising the resistance of the wire in the inductor.

A 432 turn inductor was wound using this bobbin in order to produce an A_l value for the bobbin. Direct calculation is not possible due to the uncertainty in the flux path of

the air core. The inductance of the inductor was calculated from the resonant frequency of an LC circuit.

$$L = \frac{1}{4\pi^2 f^2 C} \tag{4.5}$$

$$= \frac{1}{4\pi^2 \times 8791^2 \times 100 \times 10^{-9}}$$

$$= 3.3\text{mH}$$

$$A_l = \frac{L}{N^2} \tag{4.6}$$

$$= \frac{3.3 \times 10^{-3}}{432^2}$$

$$= 17.68 \times 10^{-9}$$

$$N = \sqrt{\frac{L}{A_l}} \tag{4.7}$$

$$= \sqrt{\frac{0.8 \times 10^{-3}}{17.68 \times 10^{-9}}}$$

$$= 213 \text{ Turns}$$

4.1.3.1 Power Supply Circuit Refinement

Experimentation was performed in order to test the energy efficiency of this circuit, and to produce the optimum design of the step down transformer.

The experiment was setup as shown in
Figure 4.4 to measure the performance of the transformer under loaded conditions:

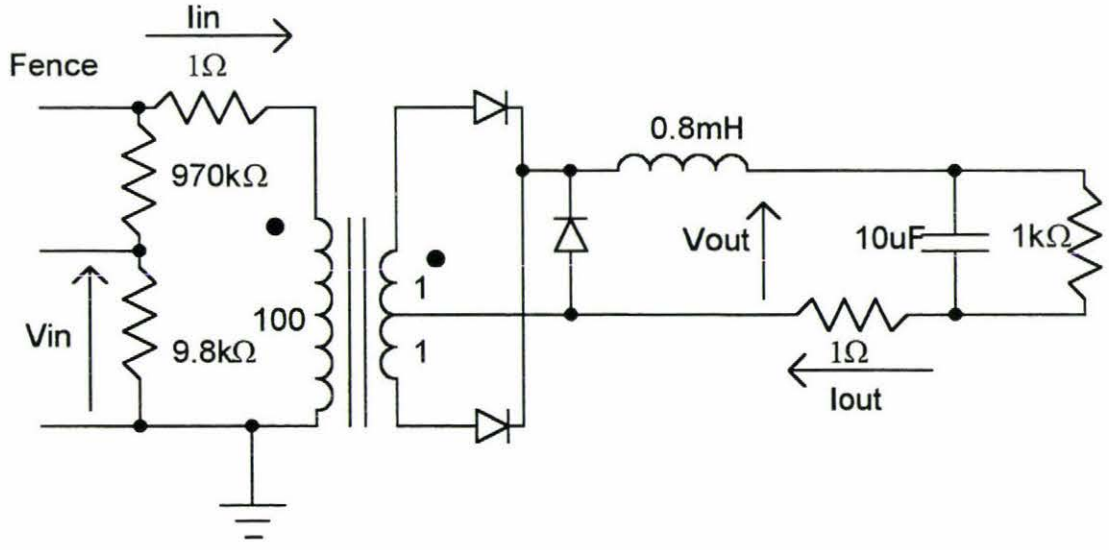


Figure 4.4 Power Supply Circuit Testing Under Load

The 100:1 voltage divider across the fence allows the use of an oscilloscope to measure the input voltage waveform. The 10Ω resistor is used to measure the current into the transformer. A 10Ω resistor is required (rather than 1Ω) as the current into the transformer is small and would be lost in noise pickup from the energiser pulse. The 1Ω resistor in the output circuit allows the measurement of the current charging the 10μF capacitor but not the discharge.

A second experimental setup was required in order to test the transformer under no load (open circuit) conditions. This was used to give an indication of magnetising currents and core losses in the transformer. This circuit is shown in Figure 4.5.

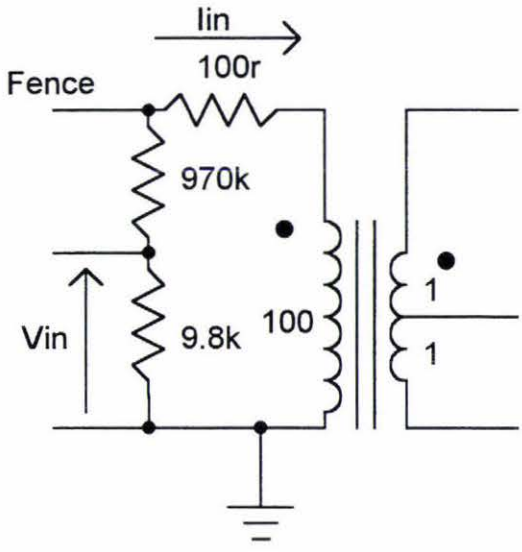


Figure 4.5 Power Supply Testing With No Load

Again the 100:1 divider is necessary for the 10kV fence pulse. A 100Ω resistor is needed as the current into the transformer is a lot smaller than in the loaded case, and would otherwise be lost in noise pickup from the energiser pulse. This increase in resistance is negligible compared to the open circuit load referred through the transformer.

The objective of these experiments is to compare the losses in several different transformers. For the experiment four transformers were used. There were two combinations of turns (same ratio) and two combinations of lamination. The four transformers are shown in Table 4.1.

Table 4.1 Slave Unit Transformer Configurations

Transformer	**1**	**2**	**3**	**4**
Laminations	Standard	Grain Orientated	Standard	Grain Orientated
Prim Turns	2000	2000	3400	3400
Ratio	100:1	100:1	100:1	100:1

The experimental data was captured using a storage scope and then transferred to a computer for calculation. Matlab was used to calculate the energy balance for the power supply system (equations 4.8 to 4.18).

$$E_{in} = \sum_{i=1}^{N} V_{in_i} I_{in_i} \Delta t \tag{4.8}$$

$$E_{measure} = \sum_{i=1}^{N} I_{in_i}{}^2 \times R_{measure} \Delta t \tag{4.9}$$

$$E_{cap} = \sum_{i=1}^{N_C} V_{in_i} I_{in_i} \Delta t \tag{4.10}$$

$$E_{copper.primary} = \sum_{i=1}^{N} I_{in_i}{}^2 \times R_{wire} \Delta t \tag{4.11}$$

$$E_{copper.secondary} = \sum_{i=1}^{N_R} I_{out_i}{}^2 \times R_{wire} \Delta t \tag{4.12}$$

$$E_{diode.rectifier} = \sum_{i=1}^{N_R} I_{out_i} \times 0.7 \Delta t \tag{4.13}$$

$$E_{diode.regulator} = \sum_{i=1}^{N_F} I_{out_i} \times 0.7 \Delta t \tag{4.14}$$

$$E_{inductor.wire} = \sum_{i=1}^{N} I_{out_i}{}^2 \times 0.74 \Delta t \tag{4.15}$$

$$E_{out} = \sum_{i=1}^{N} V_{out_i} I_{out_i} \Delta t \tag{4.16}$$

$$E_{Transformer.out} = E_{out} + E_{cap} + E_{copper} + E_{diode} + E_{inductor} \tag{4.17}$$

$$E_{Transformer.loss} = E_{in} - E_{measurement} - E_{out} \tag{4.18}$$

where N = Length of data
N_C = Length of Capacitive Current
N_R = Length of Rising Output Current
N_F = Length of Falling Output Current
E_{in} = Energy delivered into the primary of the transformer
$E_{measure}$ = Energy lost in the current measurement resistor at the input to the transformer (10Ω in the loaded and 100Ω in the open circuit case)
E_{cap} = Energy lost into the capacitance of the transformer windings
E_{copper} = Energy lost due to the resistance of the windings of the transformer
E_{diode} = Energy lost in the rectifier diodes.
$E_{transformer}$ = Energy lost in transformer that is not attributable to a specific cause.

Some interpretation of the current waveforms is required for use of the equations in Matlab. For the length of the capacitive current it is assumed that this current is that which is the current spike through the transformer as the voltage pulse arrives (as there is no other capacitance in the circuit). Figure 4.2Figure 4.6 shows the interpretation of the current in, where it is separated into capacitive current and inductive current for the standard lamination 2000 turn case.

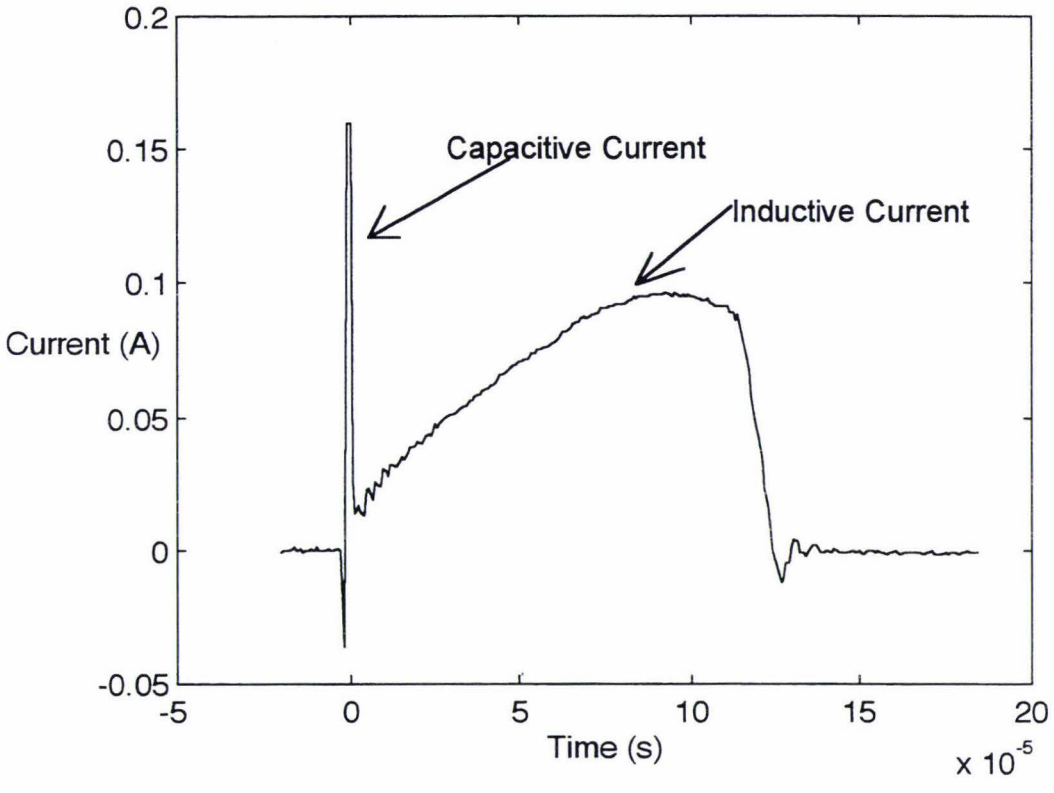


Figure 4.6 Capacitive and Inductive Transformer Current In (Loaded)

For the copper losses in the secondary case only the rising output current is used to calculate the energy as the falling current will be after the pulse has left and will be the result of the energy flowing out of the inductor (and not through the transformer). Figure 4.7 shows the distinction between the rising output current and the falling output current.

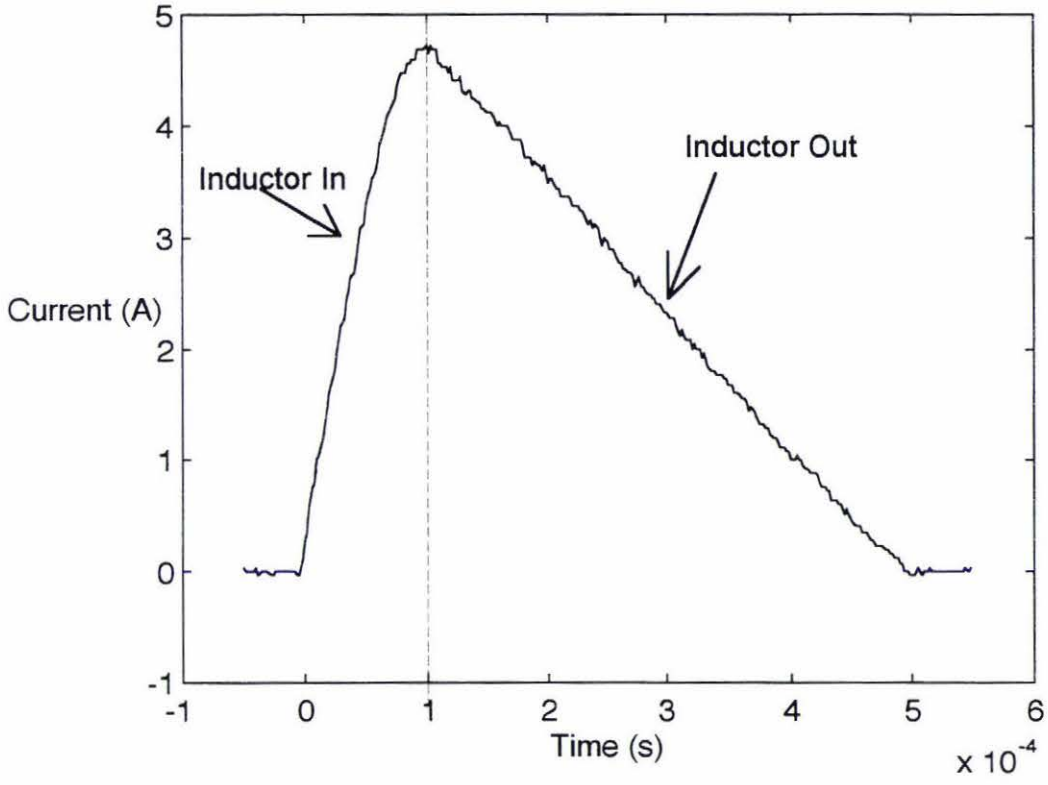


Figure 4.7 Transformer Current Out.

$E_{transformer}$ will consist mainly of Magnetising inductance and core losses. Matlab programs were written to calculate the results. A sample output is shown in Figure 4.8 for the Energy into the transformer for the standard lamination 2000 turn core. The total energy in is displayed in the graph title. The voltage and current waveforms are displayed as measured after adjustments for probes and resistor values. The resultant calculated power graph is displayed in the bottom third of the graph.

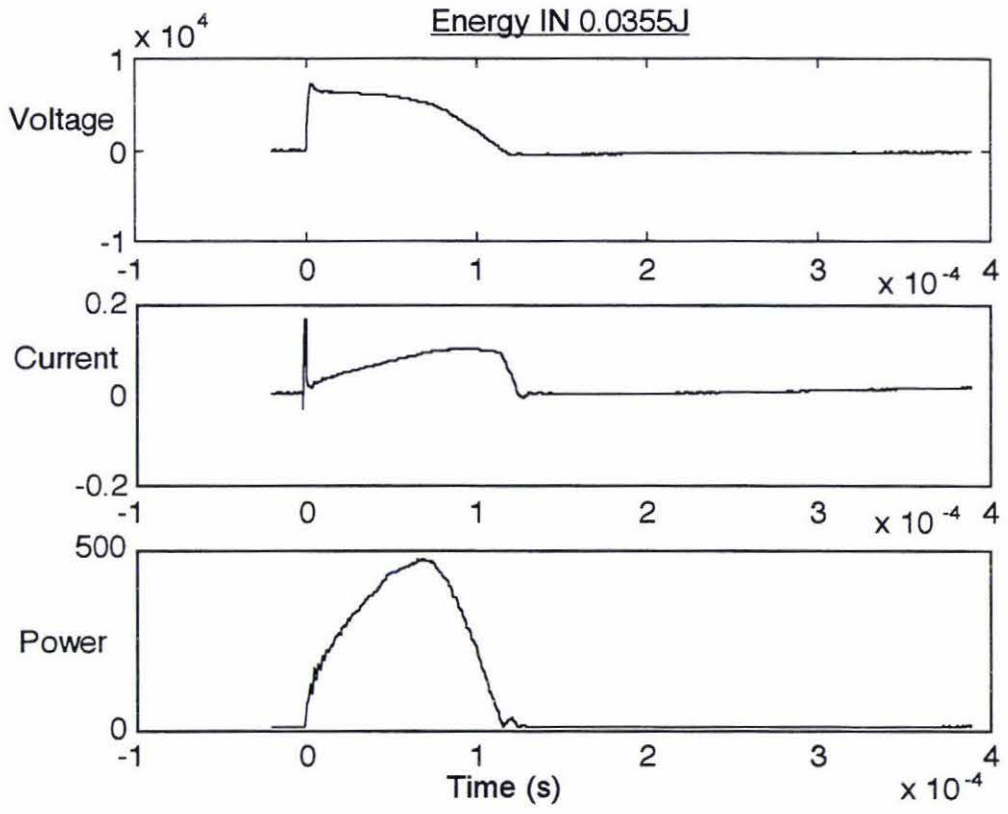


Figure 4.8 Matlab Energy Calculation Example

The energy balances are shown in Table 4.2 and Table 4.3

Table 4.2 Energy Balance Standard Laminations

Transformer	**1** : 2000 Turns			**3** : 3400 Turns		
E_{in}			35.5			21.810
$E_{measure\ 100\Omega}$			0.007			0.003
E_{cap}	1.988		35.493	2.600		21.807
$E_{copper.primary}$	0.018			0.013		
$E_{copper.secondary}$	0.926	2.932		1.570	4.183	
$E_{diode.rectifier}$	0.202					
$E_{diode.regulator}$	0.654			0.800		
$E_{inductor.wire}$	2.972	3.828		2.640	3.440	
E_{out}		11.050	17.810		9.629	17.252
$E_{transformer}$			17.683			4.555
Efficiency			31.1%			44.2%

Table 4.3 Energy Balance Grain Orientated Laminations

Transformer	2 : 2000 Turns			4 : 3400 Turns		
E_{in}			49.070			25.860
$E_{measure\ 100\Omega}$			0.013			0.004
E_{cap}	1.627		49.057	2.760		25.856
$E_{copper.primary}$	0.034			0.016		
$E_{copper.secondary}$	0.939	2.600		1.457	4.233	
$E_{diode.rectifier}$				0.199		
$E_{diode.regulator}$	0.854			0.598		
$E_{inductor.wire}$	2.931	3.785		2.619	3.416	
E_{out}		10.949	17.335		9.613	17.262
$E_{transformer.}$			31.722			8.594
Efficiency			22.3%			37.2%

These energy balances show that there is a significant reduction in the energy lost in the transformer when the turns in the transformer are increased. This is as explained by the increased inductance of the transformer, and the corresponding drop in magnetising current:

$$V = L\frac{di}{dt} \tag{4.19}$$

for constant voltage

$$I_{peak} = \frac{VT}{L} \tag{4.20}$$

$$E = \frac{1}{2}LI_{peak}^{2} \tag{4.21}$$

$$= \frac{V^2T^2}{2L} \tag{4.22}$$

$$L = \frac{\mu N^2 A}{l} \tag{4.23}$$

$$E = \frac{V^2T^2l}{2\mu N^2 A} \tag{4.24}$$

Equation 4.24 shows that the magnetising energy is proportional to the inverse square of the number of turns in the primary of the transformer if the inductance (more specifically μ) is considered constant.

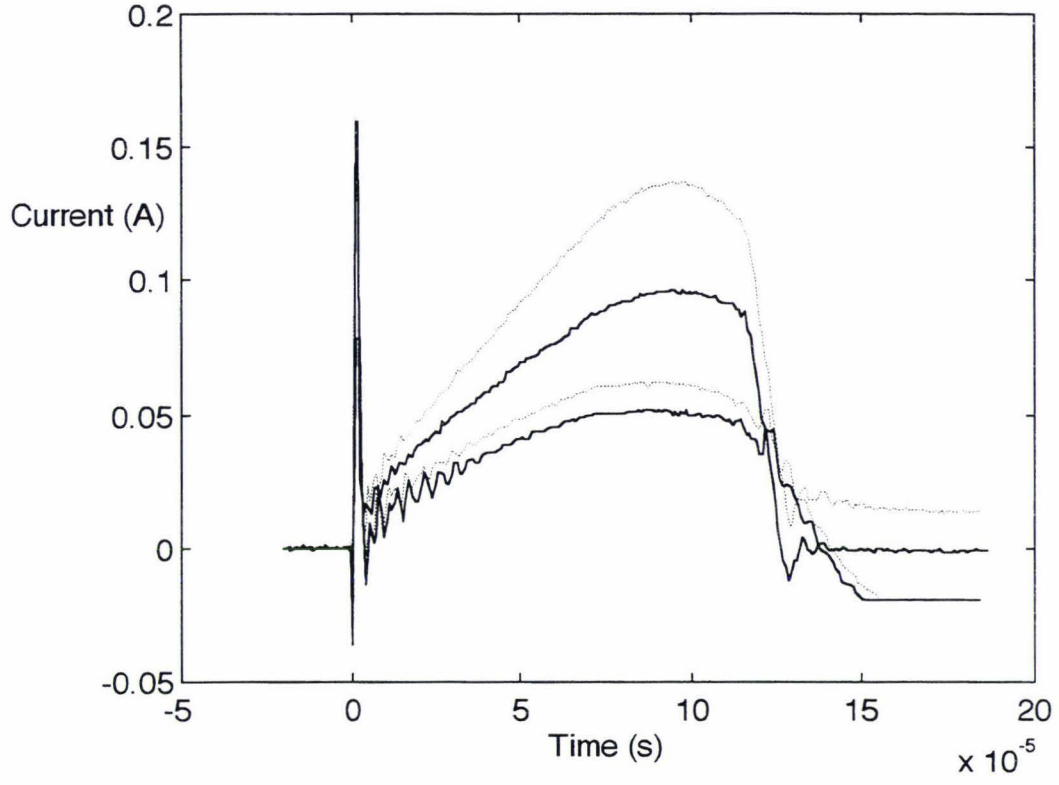


Figure 4.9 Transformer Current In (loaded), Standard (solid) vs Grain Orientated (dotted)

The experiment results measuring the magnetising current (shown in figure 4.9) also showed that the performance of the transformer deteriorates with the use of the grain orientated laminations instead of the standard laminations. This is somewhat unexpected since for grain orientated laminations μ should be a higher value, suggesting that the grain orientated laminations should perform better. The BH curves of the two materials were thus measured using the experimental setup shown in Figure 4.10.

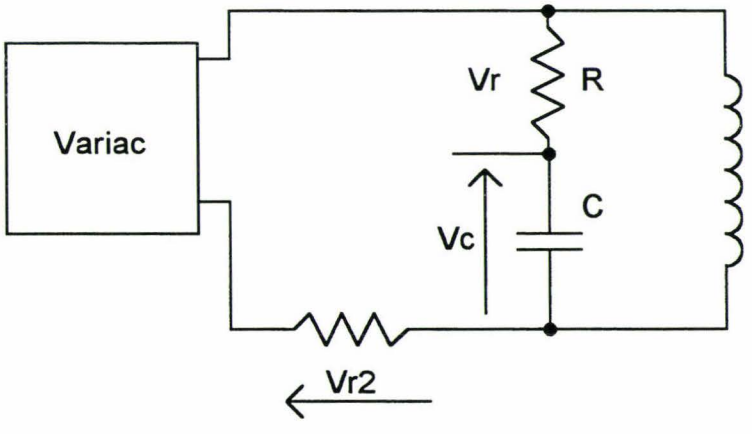


Figure 4.10 BH Curve Measurement Experiment Setup

$$H = \frac{NI}{l} \tag{4.25}$$

$$I = \frac{V_{R2}}{R2} \tag{4.26}$$

$$H = \frac{NV_{R2}}{R2l} \tag{4.27}$$

$$V_T = N\frac{d\theta}{dt} \tag{4.28}$$

$$\int V_T dt = N\theta \tag{4.29}$$

$$\int V_T dt = NBA \tag{4.30}$$

also

$$V_C = \frac{1}{C}\int i dt \tag{4.31}$$

$$V_R = iR \tag{4.32}$$

$$V_C = \frac{1}{CR}\int V_R dt \tag{4.33}$$

if $V_R \cong V_T$ then

$$B = \frac{V_C CR}{NA} \tag{4.34}$$

Thus B and H values for the core can be found by measuring the voltage at the indicated points in the Figure 4.10. Component values shown in equations 4.35 and 4.36.

$$P = \frac{V^2}{R} = 0.125W \tag{4.35}$$

$$R = \frac{230^2}{0.125}$$

$$\approx 0.47M\Omega$$

$V_R >> V_C$ then

$$R >> \frac{1}{C2\pi f} \tag{4.36}$$

$$C = \frac{100}{470000 \times 2 \times 50\pi}$$

$$C \approx 1\mu F$$

The measured voltages are then plotted against each other (on the XY setting on the scope) as the voltage traces out a working point on the curve. The variac voltage is then increased and a new set of points are gathered. Compiling a complete series of values will show the overall BH curve. The results for the two materials are shown in Figure 4.11.

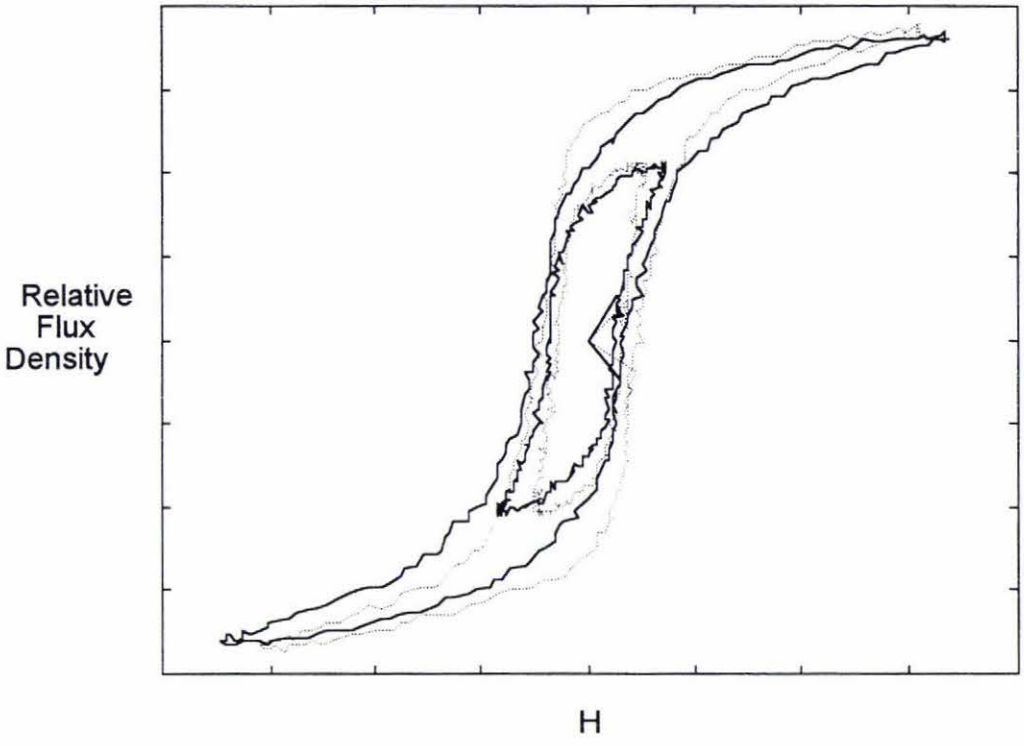


Figure 4.11 BH Curves Standard (solid) and Grain Laminations (dotted)

Only two sets are given for each of the cores, to avoid cluttering. The traditional BH curve can be obtained by connecting the end points of several of these curves. The curves of grain orientated laminations clearly show a much steeper and higher curve. Thus a lower current through the transformer would be expected.

If the energiser pulse is unipolar, the operating region of the grain orientated transformer will be much higher on the BH curve as the core remains magnetised after the pulse (Billings (1989), Slemon (1981) and Hughes (1973)). This effect of a higher residual flux is shown in Figure 4.12.

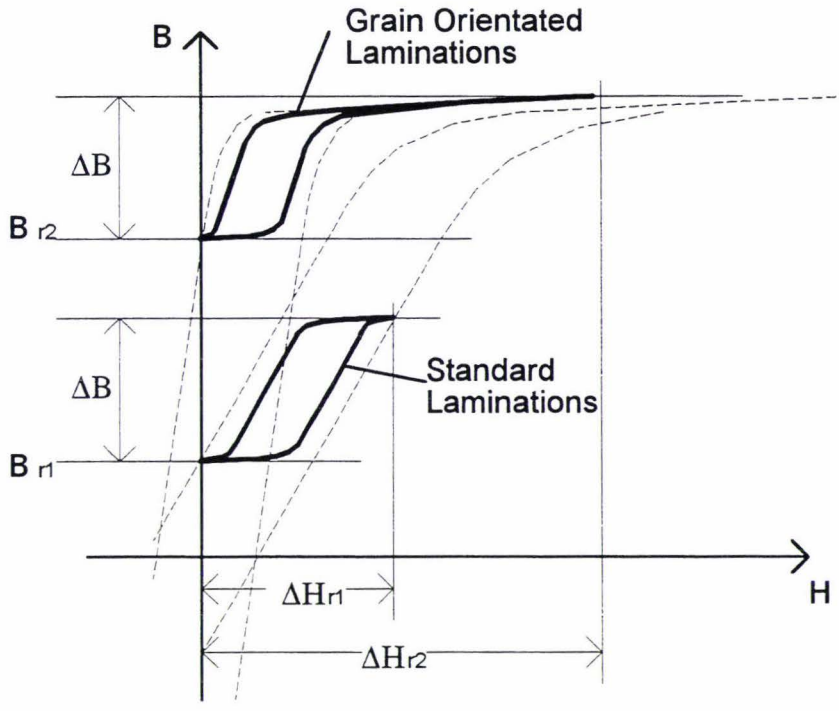


Figure 4.12 Transformer Residual Flux

In order to test this explanation, the power supply efficiency experiment for the open circuit load was performed again using a grain orientated core, and a result captured after a few seconds of operation of the power supply. The connections to the transformer were then reversed, thus ensuring that the core was magnetised in the reverse direction. The current was then captured in the first (and then succeeding) pulses. These results showed a increase in the current on each pulse. The first and final currents are shown in Figure 4.13.

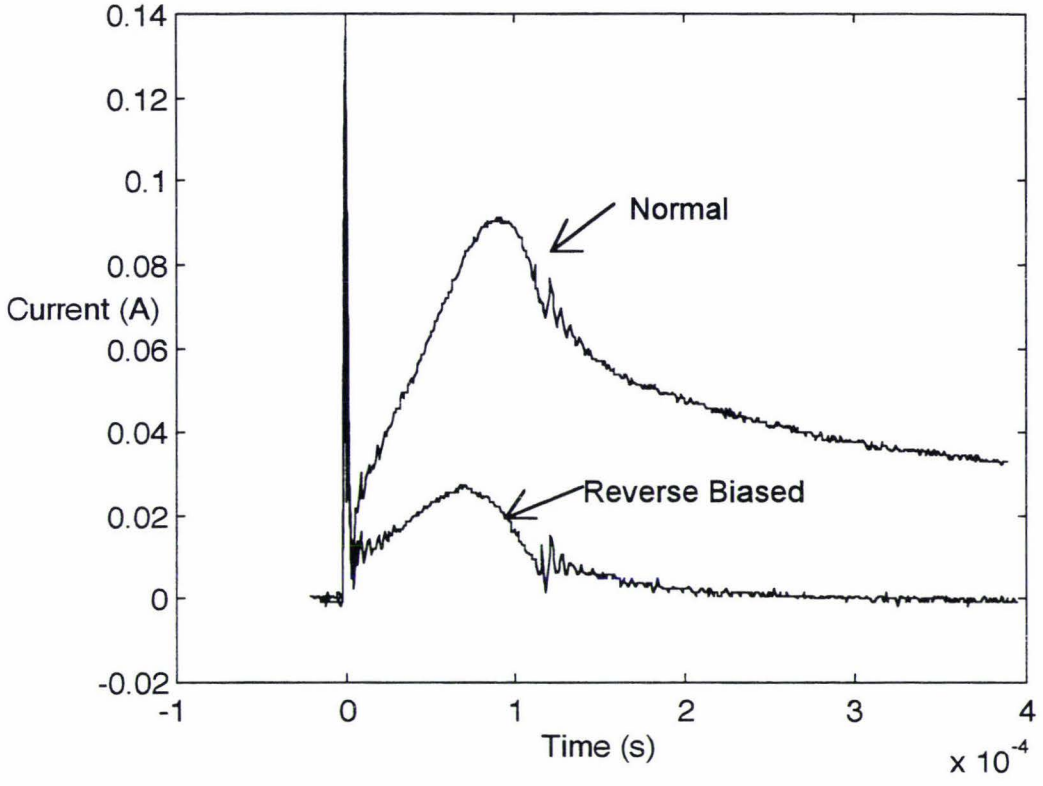


Figure 4.13 Transformer Current In, Open Circuit, Normal and Reverse Biased

Thus for this application the residual magnetism introduces inefficiency in the power supply. Grain orientated laminations are not appropriate due to the large residual flux in the core after the unipolar energiser pulse. An improvement to the standard laminations may be possible if they were gapped in order to reduce the residual magnetism (as shown in Figure 4.14) (Billings (1981), Slemon (1980), Lowdon (1981)). Yet this would also increase the current through the transformer.

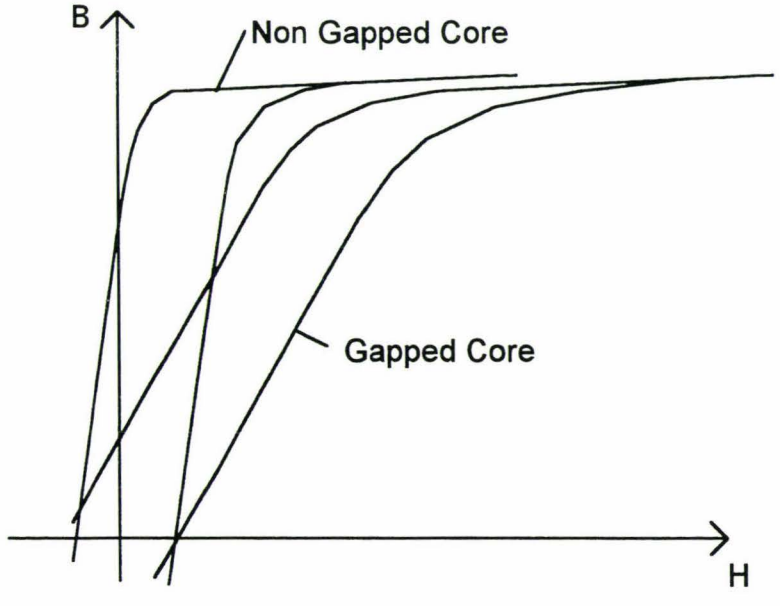


Figure 4.14 Gapping of Transformer Core

A simple test to see what improvements can be made is to perform the same experiment of reverse biasing the transformer as mentioned above, on the 3400 turn standard lamination transformer. This would indicate what could be gained if a gap was introduced in order to limit the residual flux in the transformer.

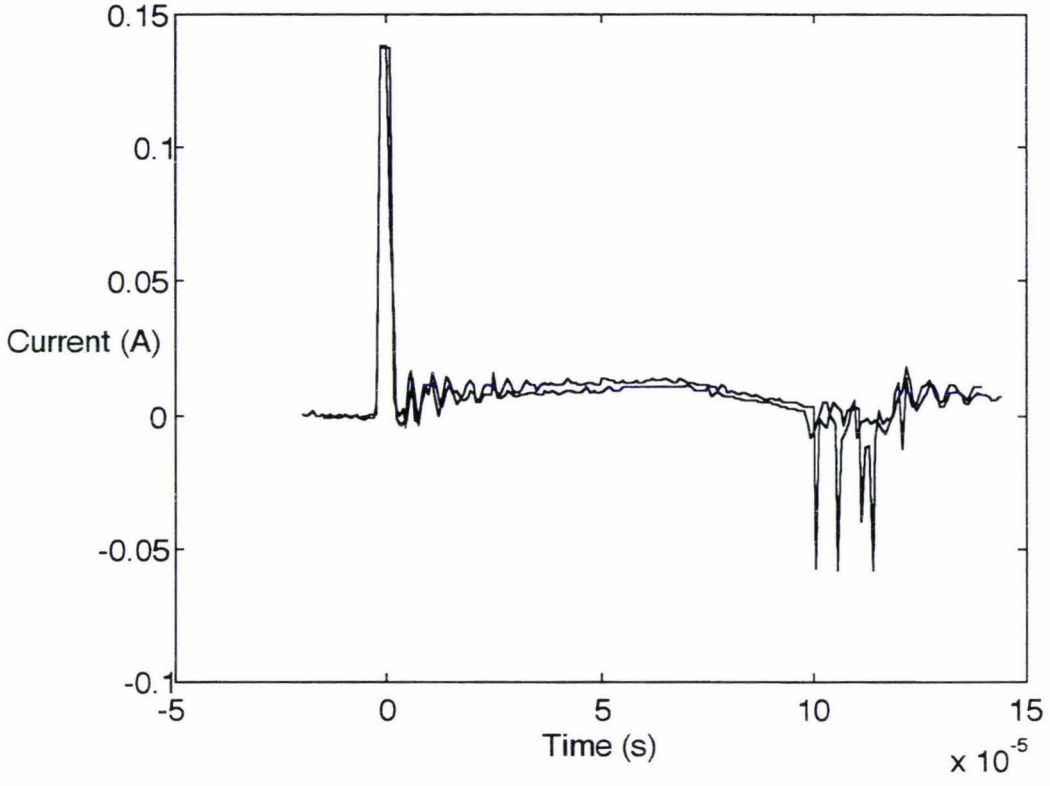


Figure 4.15 Transformer Current In, Open Circuit, Normal and Reverse Biased

The experiment shows the two currents as very similar, and thus the residual magnetism in the standard laminations is small compared with the magnetising inductance loss of the transformer. Note that the reverse biased current in Figure 4.15 shows slight gains could be made by reducing the residual magnetism, which gapping the core could achieve. However gapping the core will also cause a counteractive effect due to the reduced inductance of the transformer.

4.1.4 Transformer Specifications Summary

The efficiency tests show that the selection of the transformer core and number of turns on the primary are big factors in determining the efficiency of the total power supply.

The turns ratio should be 100:1 to step down the fence voltage to a manageable level for the slave unit. The core of the transformer should be a standard lamination instead of grain orientated lamination in order to reduce the residual magnetism of the core. There is no need for a gap in the core as the residual magnetism in the standard core is low enough compared to the inductive loss. The resistance loss in the primary is negligible compared to the magnetising inductance loss. Thus there should be as

many turns as is feasible on the primary while still maintaining the isolation requirements of an electric fence transformer. A value of 3400 turns is suggested.

Using the transformer specified an energy efficiency approaching 50% for the power supply is possible.

4.2 Measurement System:

The function of the slave unit is to measure the peak voltage of the electric fence pulse, and transmit this to a central monitoring point. Thus a measuring system is required by the unit. This must be able to measure upto a 10kV (100V after the transformer) pulse with an accuracy of a few percent. Several measurement alternatives were considered and these are described below.

All of the options presented are rated according to their cost, accuracy and power efficiency.

4.2.1 Measurement And Transmission Control Alternatives

Each of the options presented here must measure the fence pulse amplitude and determine if a transmission is required from the slave unit.

4.2.1.1 Dedicated A/D Based System

This method, as the name suggests, requires the use of a dedicated chip for the measurement process. The slave unit software will perform the function of transmission control.

The limited number of I/O pins on the microprocessor also limits the A/D chip to have a serial data bus rather than a parallel data bus. This chip will require an analogue voltage reference and will consume considerable more power than an analogue method. A solution to the large power requirement of a dedicated A/D chip is to turn the power to the chip off when the measurement is not required (only one measurement will be required after every pulse). This will then require a tri-state buffer between the two chips so that the data pins on the A/D chip do not sink current from the attached microprocessor I/O lines. This two chip design consumes considerable more space on the circuit board.

The inexpensive A/D devices require an external sample and hold circuit. This is accomplished in this application with a fast attack, slow decay filter. The system can be built with a very small capacitor to charge up to the scaled down fence pulse. A switch would also be required to discharge the capacitor between fence pulses, yet this switch could be replaced with a resistor provided the A/D can produce a result for the microprocessor fast enough. A switch may also be required to prevent the capacitor discharging into the A/D when the A/D is off. Figure 4.16 shows an example of this method.

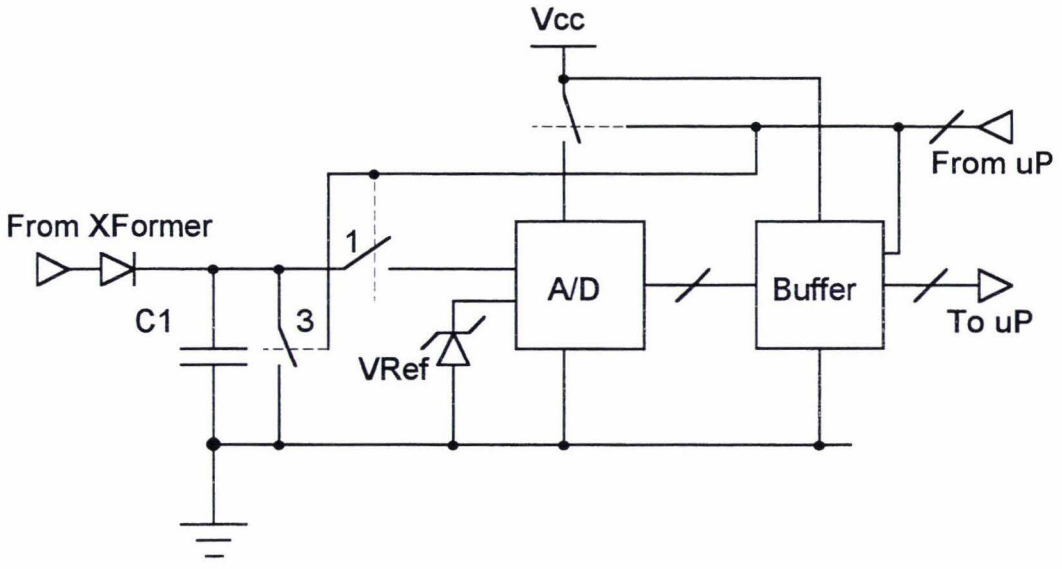


Figure 4.16 Dedicated A/D Measurement and Transmission Control System

4.2.1.2 Complete Analogue Measurement and Transmission Control

In this method the circuit measures the input voltage and determines when a transmission is required. The microprocessor need only poll a port pin to determine if it should transmit. The circuit is shown in Figure 4.17

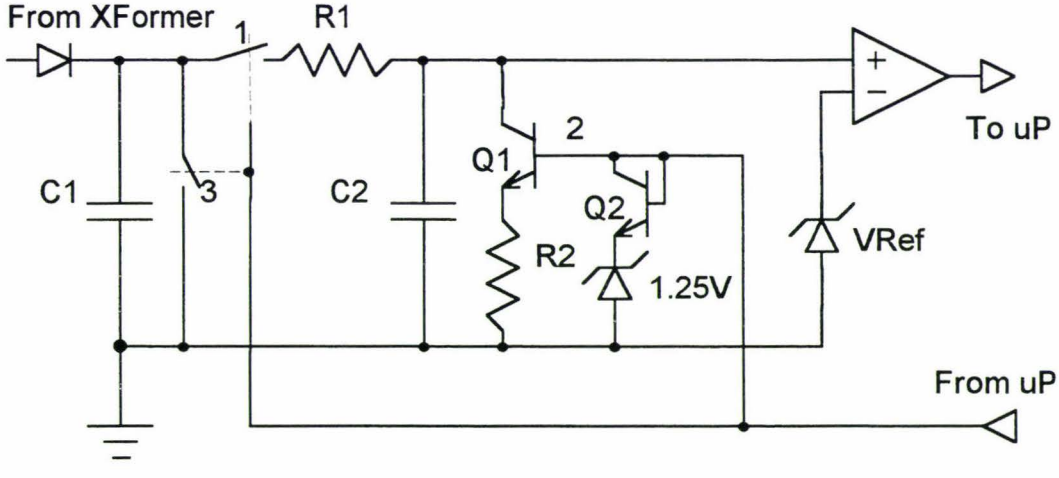


Figure 4.17 Complete Analogue Measurement And Transmission Control

The micro-processor is required to control the 3 switches, and to transmit when the voltage at C2 exceeds the reference voltage. C1 performs the task of holding the peak input voltage. C2 performs the task of aggregating the peak voltages. After every pulse switch 1 is closed for a fixed period of time to charge up C2 by a voltage proportional to the voltage at C1. It is necessary to control this time with a switch as the time between successive energiser pulses will vary between energisers. After charging C2 switch 1 is opened and switch 3 is closed to discharge C1. The voltage at C2 is sampled once every pulse to see if the aggregate has reached the threshold value. If so, switch 2 is closed for a set time in order to discharge the aggregate value

by a fixed amount and then opened to restart the process. A constant current discharge is required if C2 is to hold a residual voltage in order improve the accuracy of the measurement. The threshold voltage should be significantly lower than the minimum voltage at C1 such that the change in the voltage at C2 is as constant as possible regardless of the starting value at C2.

The equations for the voltage at C2 are (assuming V1×C1>>V2×C2):

$$V_{C2} - V_{C2}(0) = (V_{C1} - V_{C2})(1 - e^{-\frac{t_C}{R1C2}}) \quad \text{for charging} \tag{4.37}$$

$$V_{C2} = V_{C2}(0) - 2.0 \quad \text{for discharging} \tag{4.38}$$

From these equations it can be seen that if t_c is constant then if V1×C1>>V2×C2 then V_{C2} increases by a constant proportion of VC1 after each pulse. Thus an approximately linear increase in V_{C2} is possible which is directly proportional to V_{C1}, ie

$$\Delta V_{C2} = V_{C1}\alpha \tag{4.39}$$

$$\text{where } \alpha = (1 - e^{\frac{t_C}{R1C2}}) \text{ is a constant}$$

If C1=0.1uF and C2=47nF then a 100V input should not be enough to charge up C2 from 2.0V to 2.5V. Thus:

$$R = \frac{-0.366 \times 10^{-3}}{47 \times 10^{-9} \times \ln(1 - \frac{0.5}{98})} \tag{4.40}$$

$$\geq 1.53M\Omega$$

$$= 1M6$$

Matlab was used to simulate the operation of this circuit. Appendix 3 contains the code used for the simulation. The assumption that Q1>>Q2 was left in the Matlab program. The result for three cases (15V, 50V and 100V) running for 99 energiser pulses is shown in Figure 4.18.

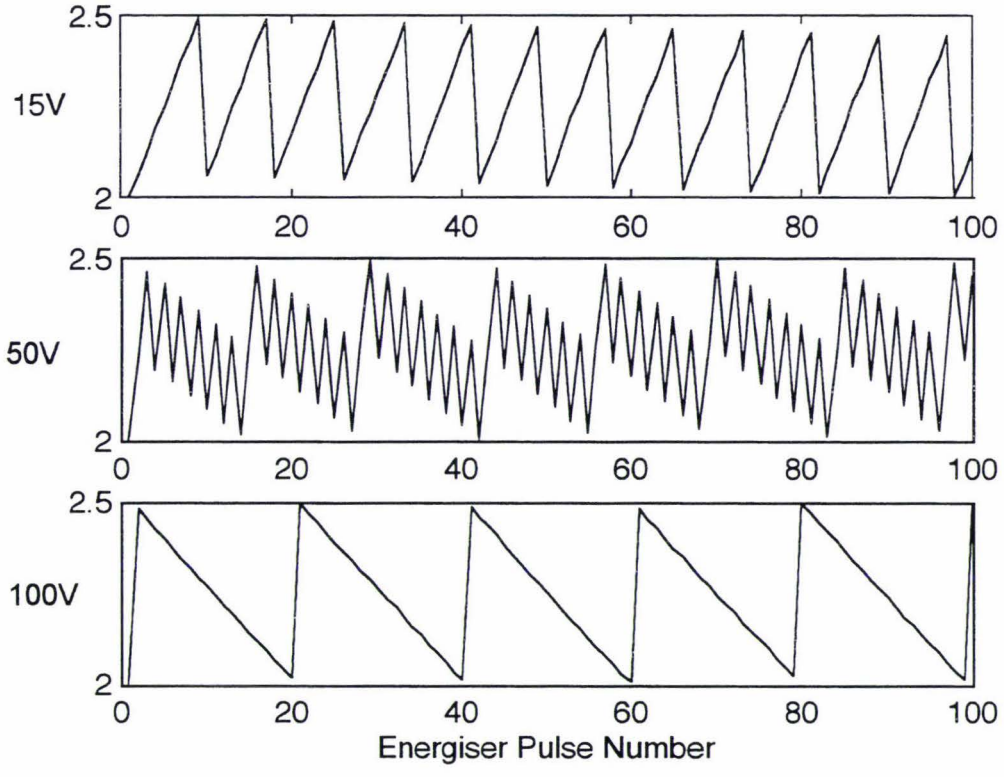


Figure 4.18 Complete Analogue System, V_{C2} Voltage Variation Over Time For Varying Input Voltage

A positive slope indicates one or more consecutive pulses are being aggregated at C2 without transmission. A negative slope indicates one or more consecutive pulse results in a transmission and partial discharge of C2.

The 15V case shows clearly when a transmission occurs, due to the discharge in V_{C2}. For this case there are 12 transmissions in all. For the 50V case again a transmission is easy to see, 45 in total. In this case it is also possible to see the charge balancing as the 50V system does not reply exactly once every two pulses. For the 100V case a transmission occurs at almost every energiser pulse, the six instances when there is no transmission are more obvious. In total there are 93 transmissions.

For the proposed system the output, number of pulses, vs the input voltage is shown in Figure 4.19.

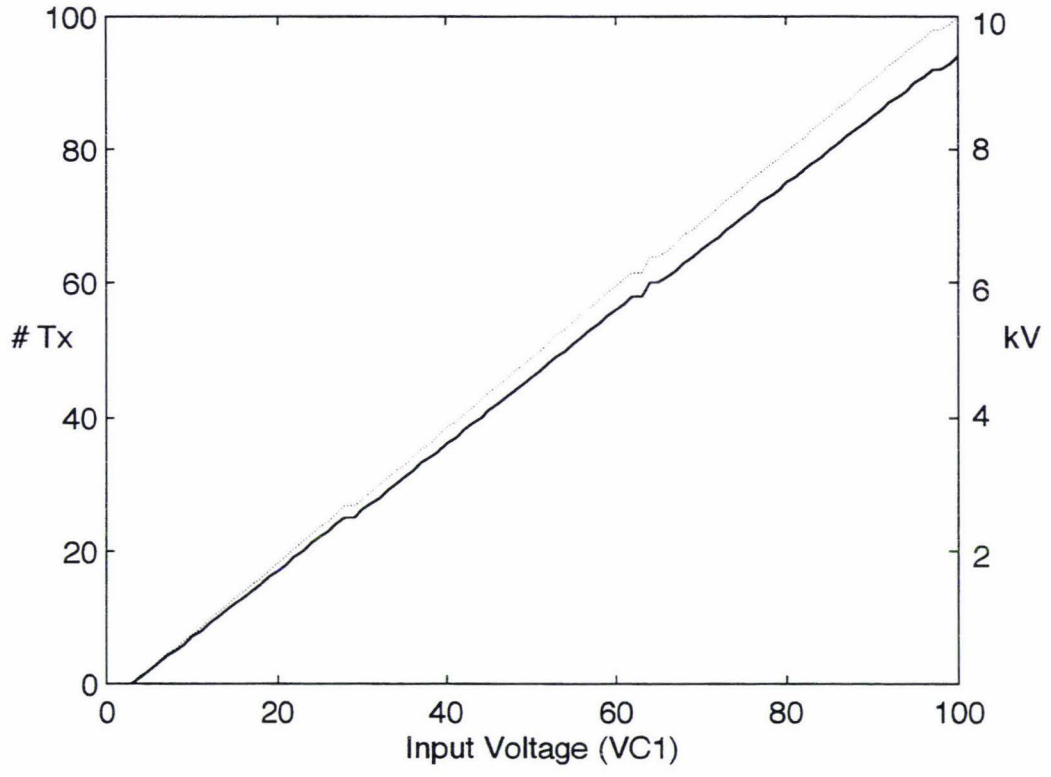


Figure 4.19 Complete Analogue System Responses (solid) vs Decoded Output (dotted)

The decoded output is found from the following equation

$$V_{fence} = \frac{\#\ Responses}{0.94} \times 100 \qquad (4.41)$$

The 0.94 accounts for the fact that the system cannot reply every time for an input of 10kV, and the 100 accounts for the transformer divider ratio. Note that there are some voltages that give the same number of responses in a given 100 response window.

This system has several limitations. Firstly the inaccuracies in the constant current discharge. This is minimised using identical transistors Q1 and Q2 and a precision voltage reference (which must work with very small currents). A large capacitor is required for C1 such that its discharge into C2 does not drop the voltage at C1. This takes more energy from the fence pulse. The minimum value for the residual voltage in C2 is 1.25V + 1 PN Junction drop. The 1.25V reference is required to ensure that the constant current sink is accurate. The maximum value for V_{C2} is the power supply voltage of the comparator. This is a minimum of 2.5V (Vref). This leaves a very low range of only 0.5V for any change in V_{C2}. Finally and most importantly, leakage currents in the circuit have to be kept to a minimum. At 100V input the charge current is only 64μA, thus leakages must be very small.

4.2.1.3 Multiple Capacitor Peak Detection And A/D Conversion

The circuit for this method is shown in Figure 4.20.

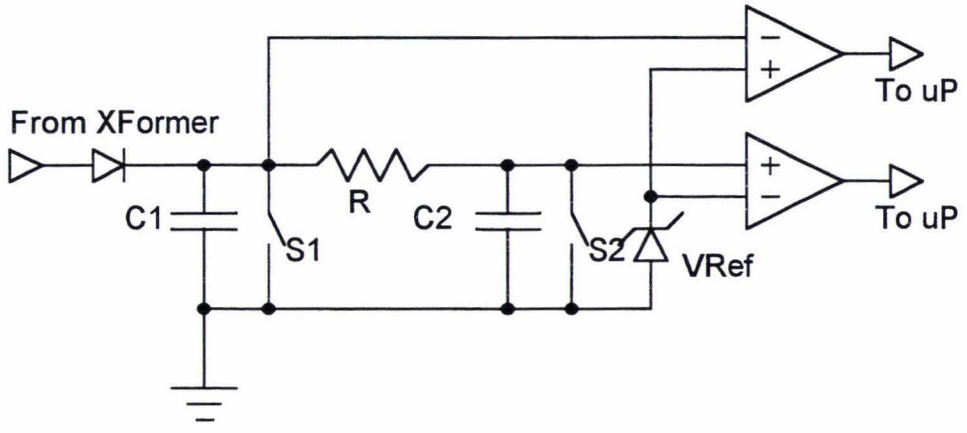


Figure 4.20 Multiple Capacitor Peak Detection and A/D Conversion Circuit

The Analog to Digital conversion is completed by the microprocessor measuring the time it takes for a known accurate capacitance (C2) to charge up to an accurate reference voltage (Vref). Allowing the microprocessor to handle transmission control simplifies the circuit and improves its accuracy.

This method requires the use of a comparator to tell the microprocessor that the capacitor charge has reached the destination value. The ATMEL 89C2051 does include a comparator, however the input leakage current for the comparator pins is specified as ±10μA. This means that to minimise leakage current errors the capacitor C2 would be very high, and the voltage source for the charging would have to be able to supply a very large current without deteriorating. In a corresponding manner C1 would also have to be very large. To overcome this problem an additional comparator with a low leakage input current may be used.

The voltage source for the charging needs to produce a voltage proportional to the peak voltage from the fence pulse. This is accomplished using a capacitor which is small enough such that it charges up to the scaled down peak value of the fence pulse within the duration of the pulse.

If it is assumed that: Q1>>Q2 (ie C1*V1>>C2*V2) and that the leakage into the comparator is negligible then the voltage at capacitor C2 is governed by equation 4.42.

$$V_{C2} = V_{C1}(1 - e^{-\frac{t}{RC2}}) \tag{4.42}$$

Thus the microprocessor could use a lookup table with a rearranged form of equation 4.43.

$$t = -\ln(1 - \frac{V_{C2}}{V_{C1}}) \tag{4.43}$$

This function is graphed in Figure 4.21.

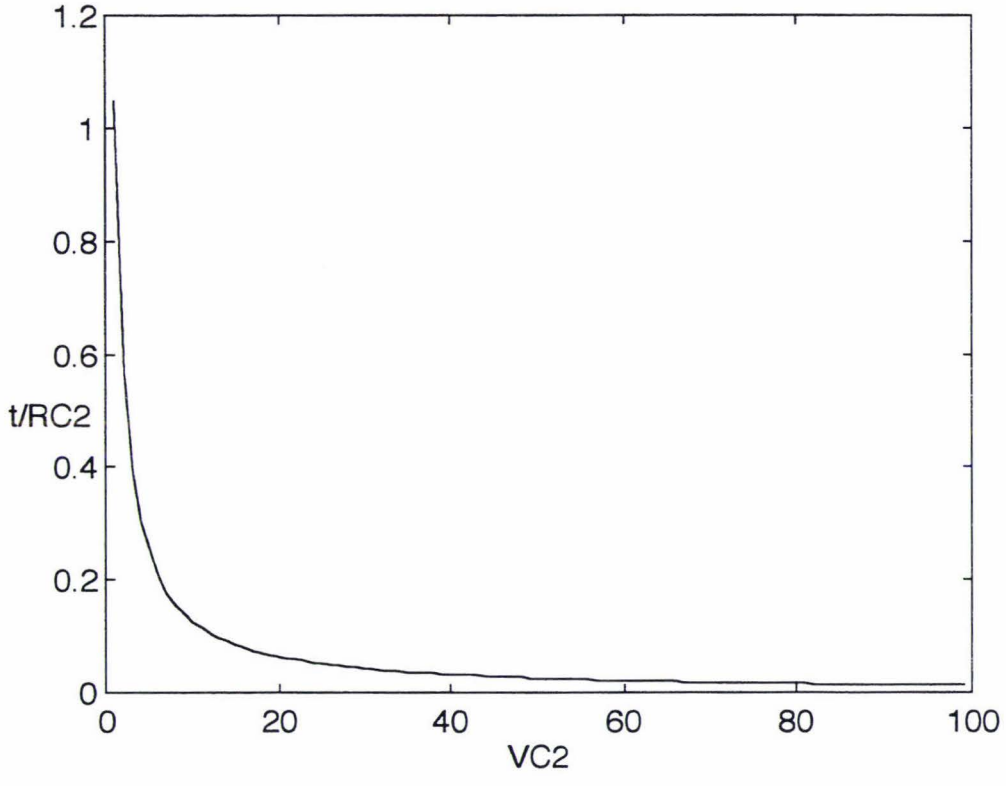


Figure 4.21 Multiple Capacitor Input Voltage vs Measurement Time

As can be seen from Figure 4.21 the conversion would take a lot longer for the lower input voltages, than the higher voltages. Thus the smallest time resolution from the microprocessor and the desired resolution at the highest voltages will be used to determine the time constant for the charging. Note the circuitry required to discharge the two capacitors before the next fence pulse arrives (S1 and S2) and the second comparator to signal to the microprocessor that the capacitor has started charging.

This system has the advantage that the greatest accuracy is obtained in the lower voltage regions, and lesser accuracy is available for the higher voltages. Thus the accuracy can be dropped in the higher voltages in order to allow a faster conversion if required. While C2 must be an accurate capacitor for an accurate reading, C1 may be of low accuracy as Q1>>Q2.

4.2.1.4 Single Capacitor Peak Detection And A/D Conversion

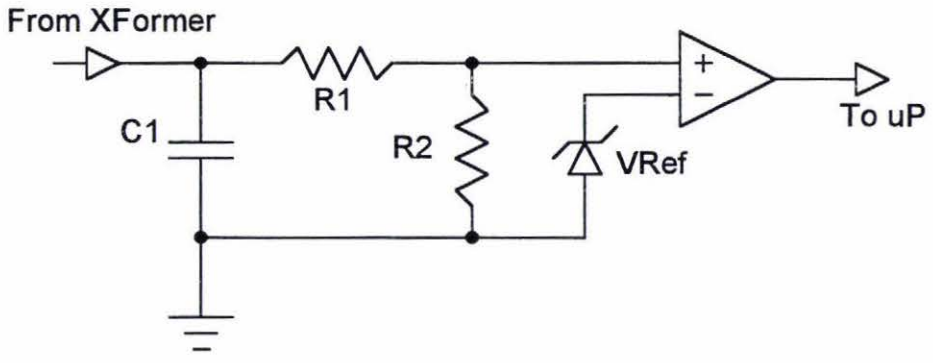


Figure 4.22 Single Capacitor Peak Detection And A/D Conversion.

The previous A/D circuit can be improved if instead of timing the charge of a second capacitor, the discharge of the first capacitor is measured. This capacitor may be a lot smaller as it does not need to be held at a constant voltage. This capacitor is part of the RC timing circuit so it must be accurate. Discharge circuitry is not required as the capacitor will be discharged before the next pulse comes along. Also circuitry is not required to signal to the microprocessor that the pulse has arrived.

The equation for the voltage at the comparator is given in equation 4.44

$$V_+ = V_0 e^{-\frac{t}{(R1+R2)C1}} \times \frac{R2}{R1+R2} \tag{4.44}$$

where V_0 is the original voltage on C1 (the scaled down peak fence voltage). This can be rearranged equation 4.45.

$$t = -\ln\left(\frac{V_+}{V_0} \times \frac{(R1+R2)}{R2}\right) \times (R1+R2)C1 \tag{4.45}$$

Again it is desirable that the input current for the comparator be as low as possible thus enabling a smaller (and accurate) capacitor C1 to be used. A comparator with internal 1.3V reference is available in the power supply circuit. The minimum working voltage for the slave unit is 1.5kV, which corresponds to 15V at C1. At this voltage there should be a minimum of 2.5% error, thus the time difference between 15V and 15.375V (2.5% larger) should be greater than or equal to the minimum time step that the microprocessor can distinguish (0.366ms as shown in section 4.4.2). Also 15V should convert to a voltage greater than 1.3V through the divider such that the microprocessor has time to jump to the interrupt routine, and then differentiate between 15V and 15.375V.

R1 must also be large. Figure 4.23 shows how the comparator protection circuitry will conduct and try to pull up the power supply if the V+ voltage is a diode drop greater than the power supply voltage. If R1 is large enough then the current will be limited out of C1, and there will be no damage to the IC, and the resulting error in the discharge will be small. An external schottky diode is included so that the external

power supply is pulled up and the IC continues to function properly. The current should also have only a very small effect on the external power supply itself. Any error in the change in resistance due to the effective short created by the diode can be accounted for in the microprocessor look-up table used to provide the A/D result.

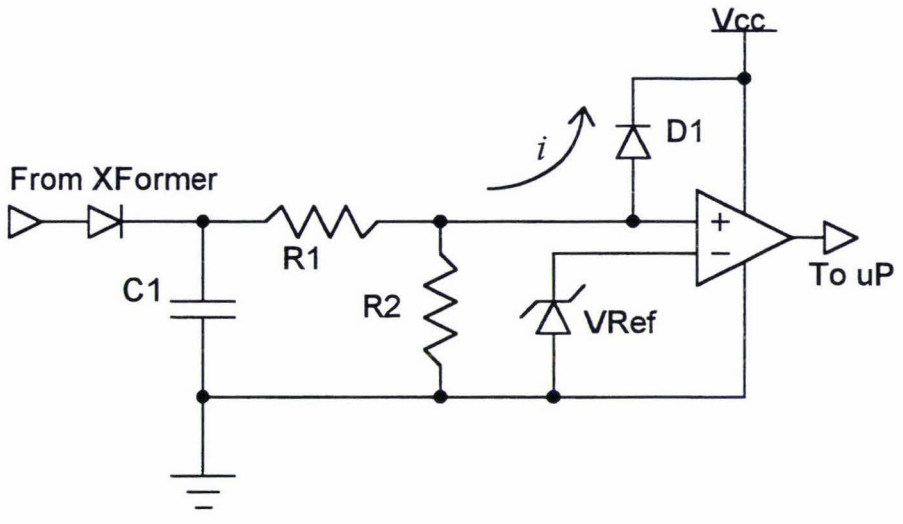


Figure 4.23 Comparator Protection Circuitry

Divider resistors are selected such that R1=1MΩ and R2=120kΩ. Thus 15V converts to 1.607V at the comparator input. Thus:

$$t_{15} = -RC \times \ln(\frac{1.3}{1.607}) \tag{4.46}$$
$$= RC \times 0.2121$$
$$t_{15.375} = -RC \times \ln(\frac{1.3}{1.6407}) \tag{4.47}$$
$$= RC \times 0.2368$$
$$t_{15.375} - t_{15} = 0.366 \times 10^{-3} \tag{4.48}$$
$$RC = 0.0151$$

Thus if R=1.120MΩ then C=13.5nF. A value of 15nF is chosen, thus the maximum and minimum time conversions are:

$$t_{15} = 3.56\text{ms}\ (9.7\ \text{macs})$$
$$t_{100} = 35.4\text{ms}$$

The microprocessor should have ample time in 9.7 machine cycles in order to jump to an interrupt in order to start measuring the time for the C1 to discharge. 35.4ms is also lower than a slave transmission timeslot (≈60ms shown in Chapter 3), thus allowing calculations before a transmission may be required.

The microprocessor requires a look-up table in order to convert the time into a fence voltage. Matlab was used to produce this table. This table has the advantage that systematic errors can be taken if required. For example the error introduced by the

capacitor C1 discharging into the power supply is predictable and can be calculated into the look-up table if required. The function is graphed in Figure 4.24.

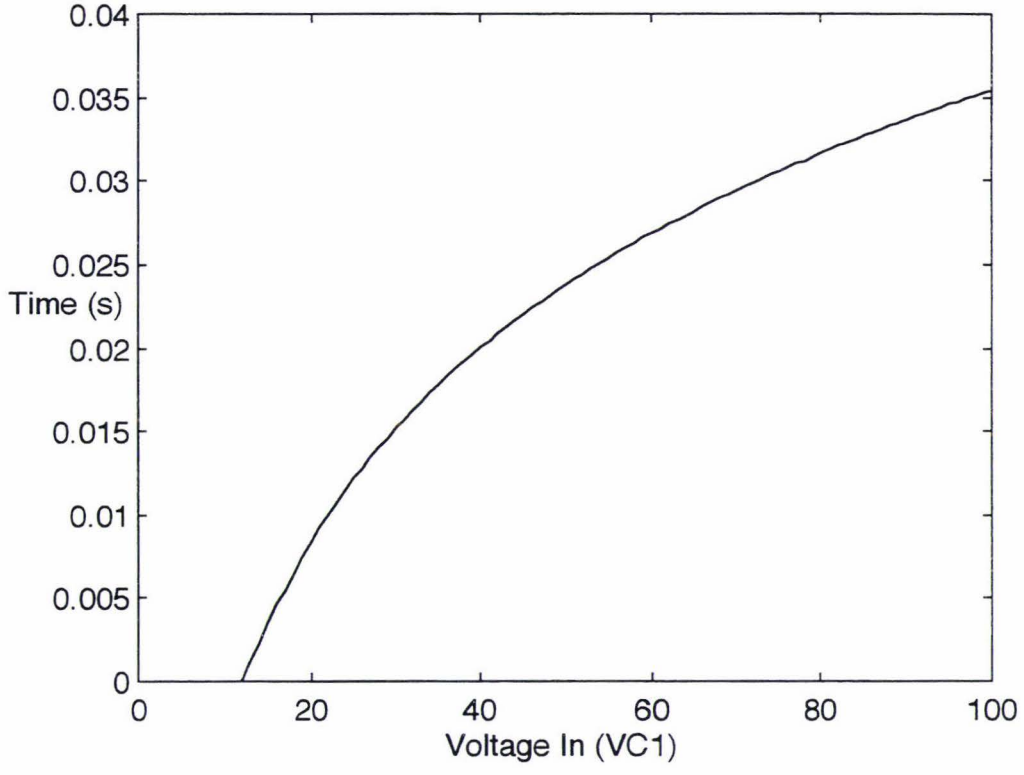


Figure 4.24 Conversion Times For Single Capacitor A/D Method

Figure 4.24 shows that the faster conversion times are obtained for the lower voltages and the absolute accuracy drops as the input voltage increases, thus a percentage error can be used with the output reading.

This method will rely on the microprocessor to calculate the times when a transmission is required.

4.2.2 Measurement Method Chosen

The single capacitor peak detection and A/D conversion method was chosen for implementation in the slave unit, as this method was simple to implement, was cost-effective, could use standard components, and was sufficiently accurate.

4.3 Transmission Circuit

A circuit is required to transmit to the master unit using the fence line. This signal must propagate down the fence easily and be detectable at the master unit. Two methods are presented to do this, the first is a pulse in the slave timeslot, the second is a frequency burst in the timeslot. This section outlines the circuit chosen and presents results of a Matlab simulation for the circuit.

4.3.1 Transmission Alternatives

4.3.1.1 Pulse

This method is the simplest transmission method. It consists of placing a pulse on the fence line when the slave unit transmits. Figure 4.25 shows an example of the operation of this method.

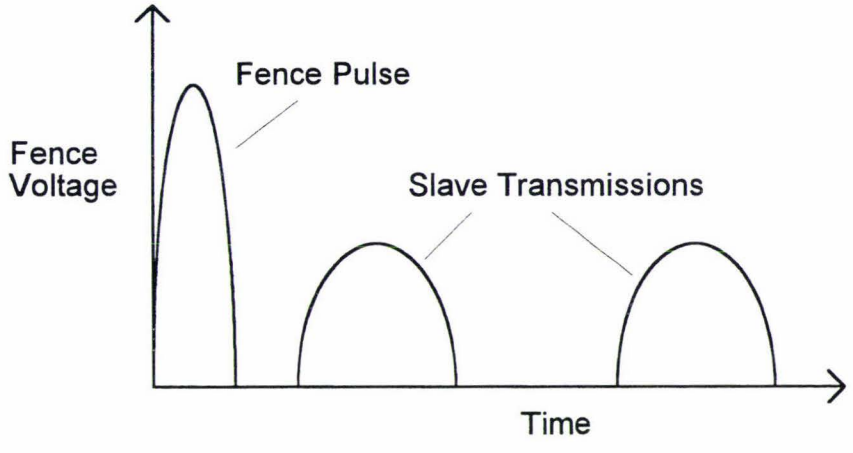


Figure 4.25 Pulse Slave Unit Transmissions

While it is simple for the slave to put these pulses on the fence their detection could be complicated by the fact that another energiser operating near this electric fence may couple its pulse onto the fence, and thus cause false detection at the master unit. More importantly, a large amount of energy is required to make the pulses detectable, ensuring that the slave units could not be powered from the fence pulse.

4.3.1.2 Frequency Burst

While this method is more complicated than the pulse method, if the frequency is chosen carefully, then the transmission can be unique, in that there are no other significant signals of that frequency on the fence line. A precise frequency can be generated by the microprocessor. Section 4.4.3 details the choice of frequency for the transmissions at 273.07Hz. Using a microprocessor two methods for the frequency burst are available, a square wave and a sine wave.

4.3.1.2.1 Square Wave

The microprocessor output will be a square wave. The simplest transmission circuit would involve switching the power supply directly into the transformer using the microprocessor digital output signal, using a series impedance to limit the current through the transformer. In effect this would end up switching a current through the transformer.

The output of the transformer is proportional to the change in current at the input of the transformer. When the microprocessor switches from one power supply to the other, the output of the transformer will be very large, but when the switching is not in progress the output of the transformer is zero.

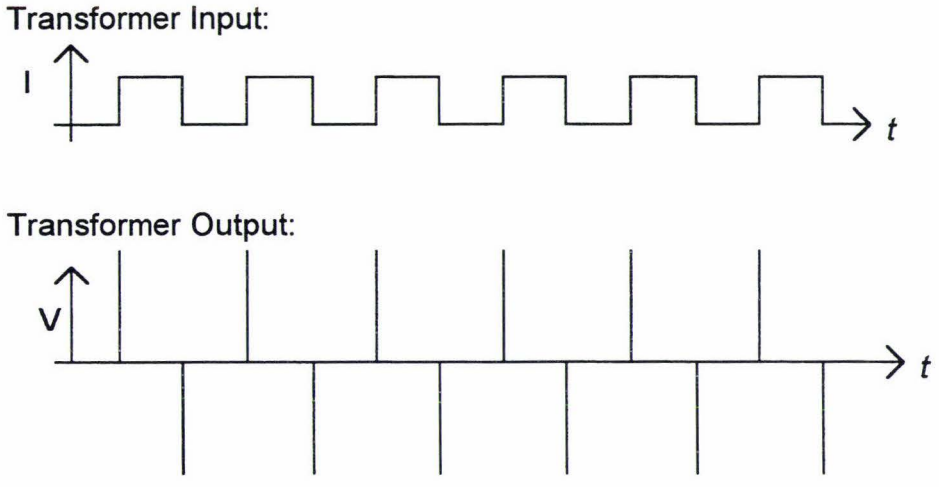


Figure 4.26 Transformer Switching Response

Figure 4.26 shows the transformer output is a accurate repetitive waveform at a frequency of 273.07Hz due to the accuracy of the switching input. Yet it is more difficult to detect than a simple sine wave as the fraction of output power at 273 Hz is very low when compared to a simple sine wave output. Thus this method is not as efficient as a sine wave.

The output switches and current limiting circuit must also be able to withstand or block the input fence pulse, without breaking down or taking power out of the pulse.

4.3.1.2.2 Sine Wave

The simplest way to convert the square wave to a sine wave is to put the digital output through a 273Hz resonant circuit (or filter tuned to 273Hz) shown in Figure 4.27.

This method will produce a sine wave output at the transformer input and therefore requires extra circuitry but is far more efficient and easier to detect at the far end of the fence.

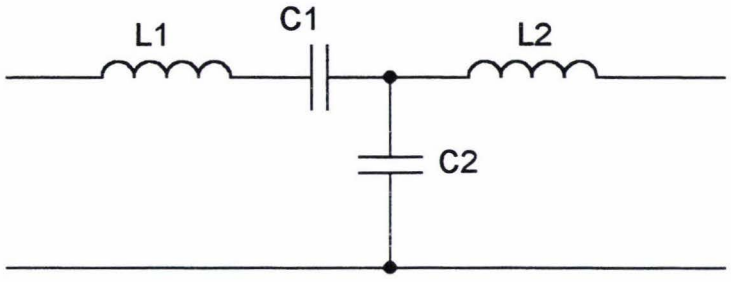


Figure 4.27 Transmission Resonant Circuit

The inductance in the filter can perform a second function. L2 is large and provides a high series impedance to the fence pulse. This means the transmission circuit is protected and the pulse energy is not bypassed. L2 is not so large as to prevent the low frequency slave unit reply from being coupled onto the fence.

A series capacitor is required to block any DC voltages from the slave transmission getting onto the fence.

L1 and C1 should form a series resonant circuit with a resonant frequency of 273.07Hz and L2 and C2 should do the same. The impedance of the second circuit is lower than that of the first in order to obtain an impedance transformation. Thus the drain from the power supply will be restricted by the first circuit, yet the output impedance of the filter will be low. To achieve this L1 should be greater than L2 and C1 should be lower than C2, while L1C1 and L2C2 still have then same resonant frequency.

The circuit is further complicated by the requirement that the microprocessor and the transmission output come from different power supplies. The microprocessor should run off its own supply circuit which is derived from the main supply through the power up circuit. The transmission output should come from the main supply itself in order to provide enough power to the fence.

Although slightly more complex to implement than the pulse or square wave alternatives, its greater noise immunity and efficiency means that this circuit is appropriate for the slave unit transmission circuit.

4.3.2 Modelling Of The Transmission Circuit

Matlab was used to create a ordinary differential equation model for the transmission circuit. This model was used to evaluate the response of the circuit to input waveforms, the performance of the circuit, the response to different terminating impedances and the response from circuits with component errors. Appendix 4 contains the Matlab code for the transmission circuit model.

The modelling method requires breaking down the equations for the circuit into a series of ordinary differential equations. Matlab then uses the Ranga-Kuuta method to produce a numerical solution to the equations.

The modelled circuit is shown in Figure 4.28.

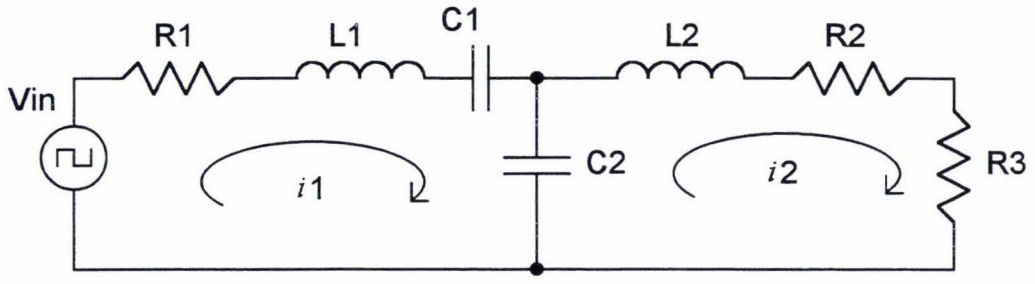


Figure 4.28 Modelled Transmission Circuit.

The generator at the left is the digital input voltage from the microprocessor and driver circuit. The generator switches between 3V and 0V. R3 is the terminating impedance and represents the impedance of the energiser and fence referred through the slave transformer.

The model equations are given in equations 4.49 to 4.52.

$$\frac{di_1}{dt} = \frac{V_{in} - i_1 R_1 - V_{C1} - V_{C2}}{L_1} \tag{4.49}$$

$$\frac{dV_{C1}}{dt} = \frac{i_1}{C_1} \tag{4.50}$$

$$\frac{dV_{C2}}{dt} = \frac{i_1 - i_2}{C_2} \tag{4.51}$$

$$\frac{di_2}{dt} = \frac{V_{C2} - i_2 R_2 - i_2 R_3}{L_2} \tag{4.52}$$

A sample output from the Matlab model is shown in Figure 4.29. This simulation was run with L1=35mH, R1=80Ω, C1=7.5μF, L2=14.5mH, R2=3.5Ω, C2=22μF and R3=0.01Ω representing the design minimum of 100Ω energiser impedance.

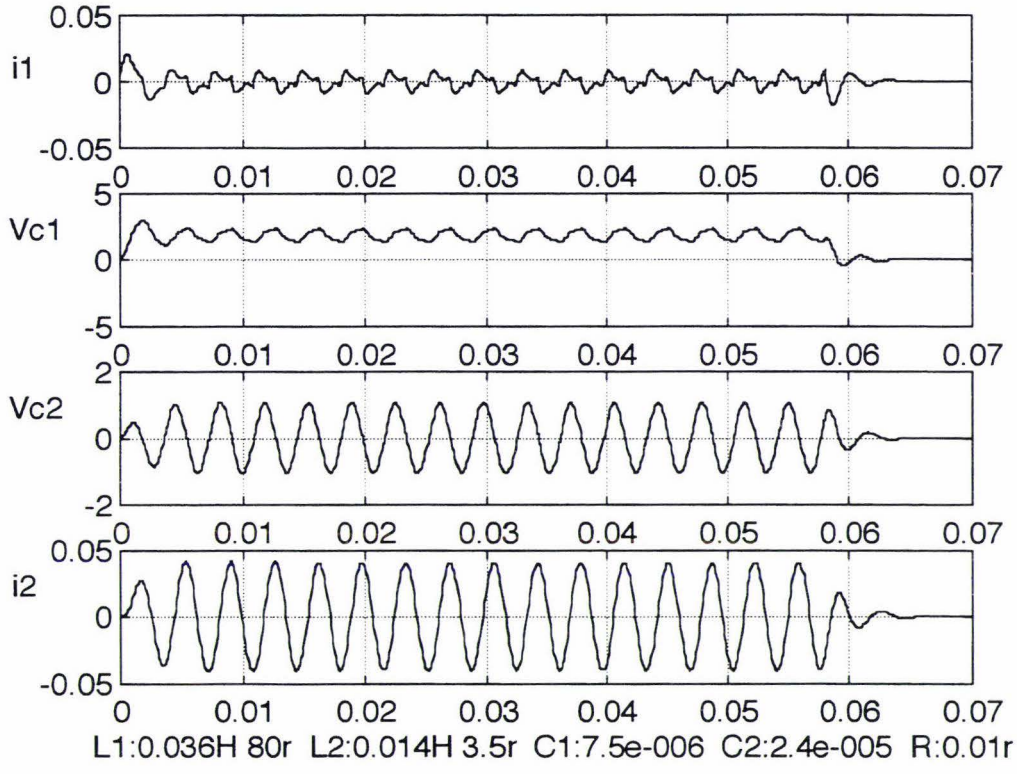


Figure 4.29 Slave Transmission Simulation

The sample output of Figure 4.29 shows that with a energiser impedance of 100Ω the voltage across the energiser would be 100*0.01*i2 (due to the 100:1 transformer ratio). Thus a amplitude of less than 100mV peak to peak is available as a slave output. Options for changing the energiser impedance, and transmission detection are presented in chapter 5.

The simulation showed that the capacitors may need to be bipolar. Bipolar capacitors of the required values are expensive. Circuit changes can be made to use electrolytic capacitors. The section 4.3.3 outlines how this is accomplished.

The simulation also showed that the filter output will decay quickly once the digital input is removed (at 58ms in the above example) and there is no need to leave large guard bands between slave timeslots.

The simulation showed that the resistance of L2 was critical as R3 is very small. Thus the resistance of the second inductor must be minimised in order to maximise the voltage across the energiser. Figure 4.30 shows the drop in output current when the inductor resistance of L2 is increased to 10Ω

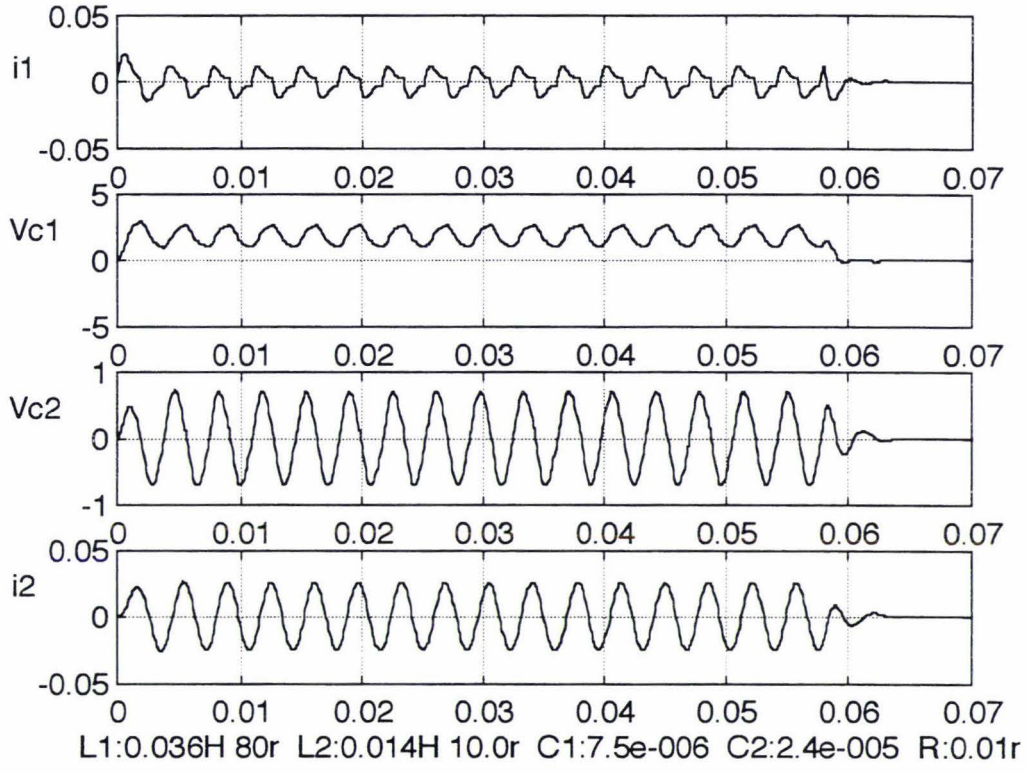


Figure 4.30 Slave Unit Transmission Simulation With Large Inductor Resistance.

The simulation also showed that the output frequency was not sensitive to small variations in the component values of the circuit. Figure 4.31 shows the drop in amplitude of i2 for very large component errors in the output circuit. In the final circuit if the errors in the capacitance values will be ±20% and in the inductor values approximately 10%, and thus the effect will be far less significant.

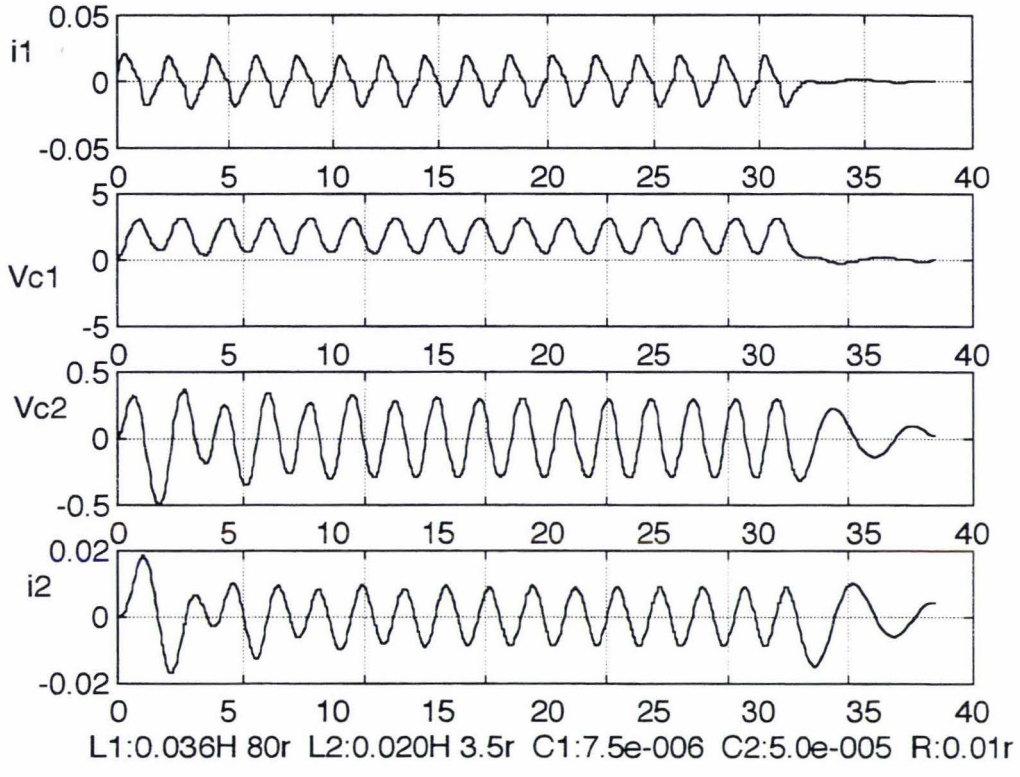


Figure 4.31 Slave Unit Transmission Simulation With Errant Components

4.3.3 Transmission Circuit Components

As mentioned in the previous section both the capacitors must be bipolar, yet their capacitance values dictate that they be electrolytic capacitors. By connecting two capacitors in series, of twice the desired capacitance, this can be accomplished as shown in Figure 4.32. Both remain charged up to the power supply voltage, and either discharge to accomplish bipolar operation.

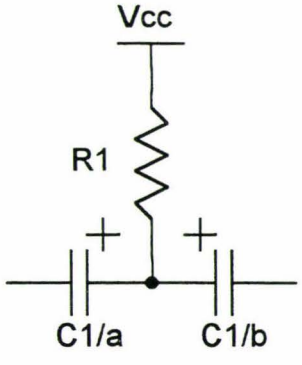


Figure 4.32 Electrolytic Capacitors Replacing Bipolar Capacitors

L2 of Figure 4.27 has to be big enough to block the incoming energiser pulse. If the transmission circuit is to take 5% of the energy of the power supply the inductor value can be calculated by equations 4.53 to 4.55.

$$I = \frac{V\tau}{L} \tag{4.53}$$

$$L2 > 20 \times L_{\text{Power Supply}} \tag{4.54}$$
$$> 16\text{mH}$$

$$C2 = \frac{1}{4\pi^2 f^2 L} \tag{4.55}$$
$$= 21.2\mu\text{F}$$

2×C2 must take on a standard value, thus the capacitors for C2 are 47μF capacitors, and thus C2 becomes 23.5μF and L2 becomes 14.5mH in order to keep the same resonant frequency.

The values of the first resonant circuit set the amount of energy that is released in each transmission. Thus C1=11μF (0.5×22μF) and L1=31mH

4.3.4 Transmission Driver Circuit

A driver circuit is required for the transmission as the microprocessor output pins cannot supply enough current for the transmission circuit.

The simple emitter follower output driver (shown in Figure 4.33) is not possible in this case due to the base emitter voltage drops of the transistors.

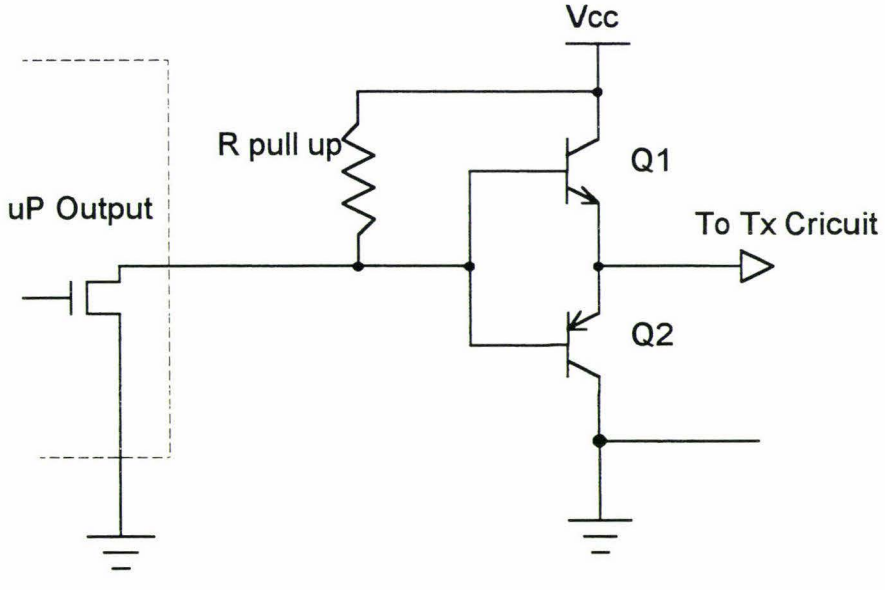


Figure 4.33 Emitter Follower Output Driver.

With a 3V power supply voltage the transmission output voltage range is shown in equation 4.58.

$$V_{max} = Vcc - V_{Pull\ Up} - V_{be} \tag{4.56}$$
$$= 3 - 0.3 - 0.7$$
$$= 2.0V$$
$$V_{min} = Gnd + V_{FET\ saturation} + V_{be} \tag{4.57}$$
$$= 0 + 0.8 + 0.7$$
$$= 1.5V$$
$$V_{max} - V_{min} = 0.5V \tag{4.58}$$

As an improvement common emitter configurations are used giving only small collector emitter saturation voltage drops and diodes are placed across the transistors to prevent collector base breakdown, when the fence line rises above the power supply or falls below the slave ground voltage. Two microprocessor output pins are now required to drive the transmission circuit, but also three output states can be achieved.

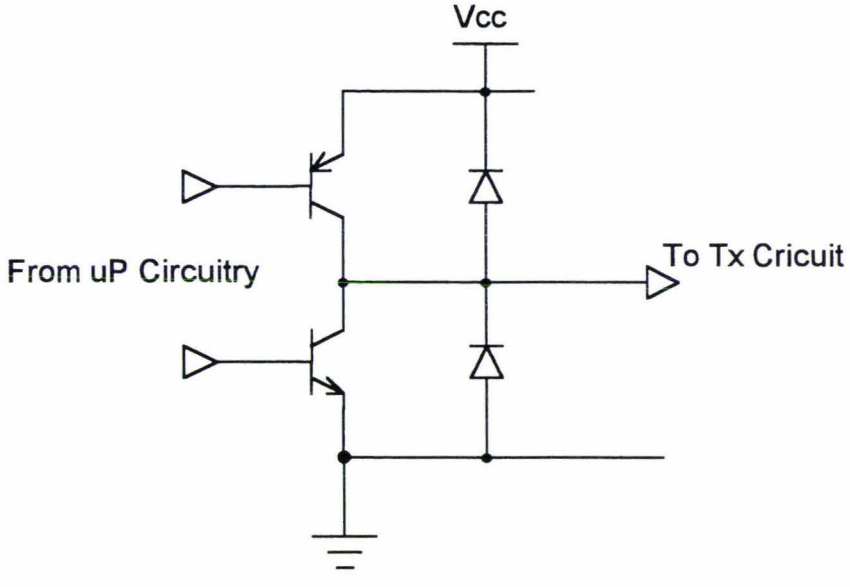


Figure 4.34 Slave Unit Transmission Driver Transistor Configuration

This driver circuit must now also isolate the microprocessor power supply from the transmission circuit power supply, such that when the microprocessor is off, the microprocessor power supply is not pulled up from the energiser pulse passing through the transmission circuit. The energiser pulse should be directed into the main power supply only. Thus the high PNP transistor base needs to be pulled up by the main power supply, and not the microprocessor.

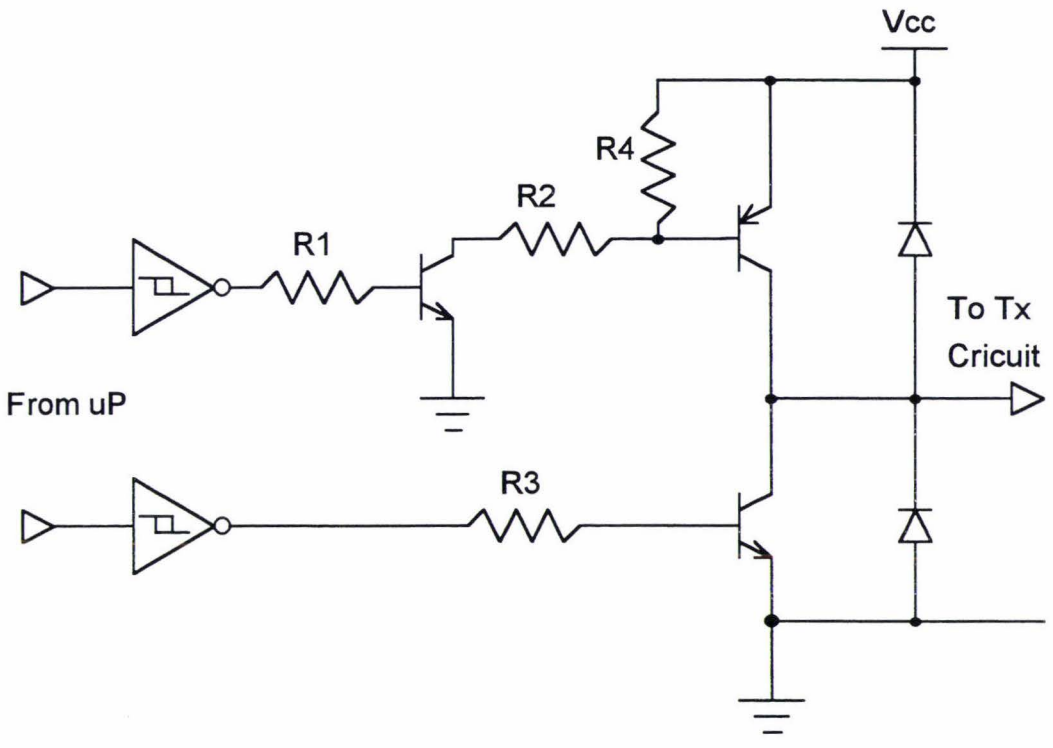


Figure 4.35 Slave Unit Transmission Driver Circuit.

The invertors used ensure that the off state at the I/O pin which is a high and the on state is a low voltage taking less power as the transmission circuit is off more often than it is on. The invertors are incorporated in the package also used in the microprocessor oscillator (section 4.4.2.2)

4.4 Microprocessor

This section details the issues surrounding the use of the microprocessor in the slave unit. For this application very little energy is available, so the microprocessor circuit must be very energy efficient.

The power consumption of the microprocessor is dependant upon: microprocessor power up, power supply voltage ($E \propto V^2$), oscillator circuit, oscillator speed ($E \propto f$), microprocessor sleep time. These are examined throughout this section with a mind to reducing the total power consumed in the microprocessor circuit.

4.4.1 Power Up Circuit

The microprocessor will only operate above 2.7V. Below this voltage level (when the oscillator does not work) the microprocessor will consume a large current when compared to normal operating conditions. Hence circuitry is required to prevent the microprocessor from powering up until there is sufficient voltage in the power supply. This voltage is accumulated over several pulses.

Several different options for the slave power up circuit are presented, and discussed.

The foundation of all of the circuits presented is a pass transistor. Additional circuitry is required to control this device

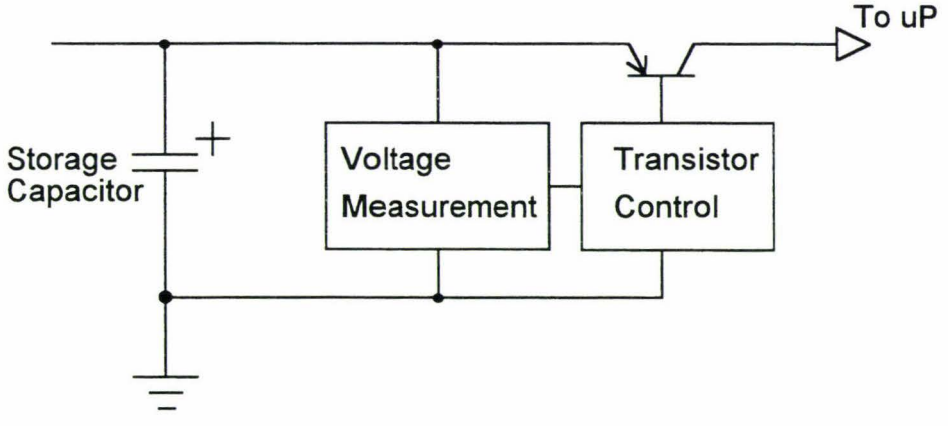


Figure 4.36 Microprocessor Power Up Circuit.

The circuit must also include some latching or hysteresis, as the microprocessor will consume a significant current on power up and drop the power supply voltage.

4.4.1.1 Reference Diode With Positive Feedback For Turn On

Here three similar circuits are presented together.

All of these circuits rely on a 2.5 zener conducting to turn on the microprocessor when the power supply rises above 2.5V. In these circuits hysteresis and switching is

accomplished by the use of positive feedback from the pass transistor output for rapid latch on.

In the first circuit feedback is provided by a resistor connection to the base of the pass transistor driver as shown in Figure 4.37.

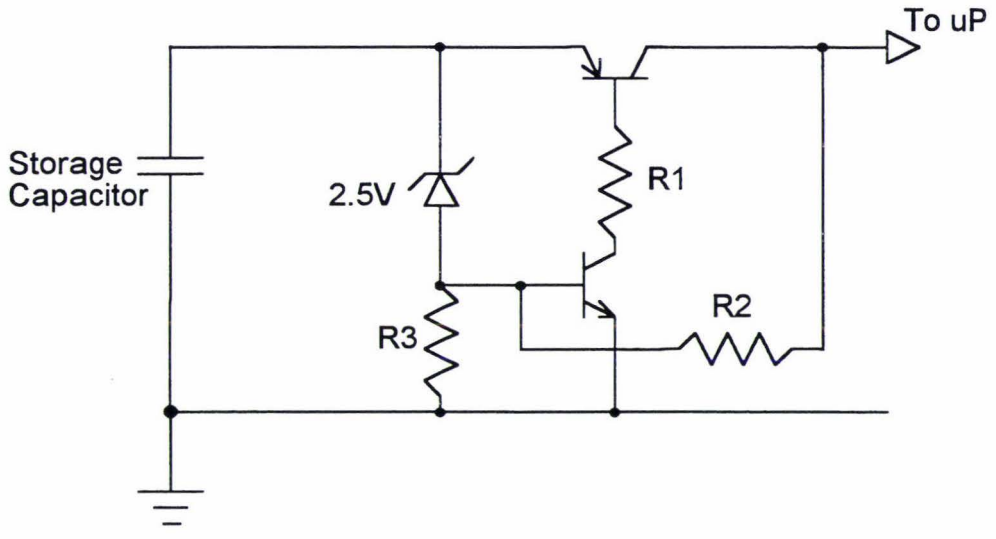


Figure 4.37 Reference In Series With Control Switch

In the second circuit the positive feedback is achieved by shunting out the 2.5V reference with a transistor.

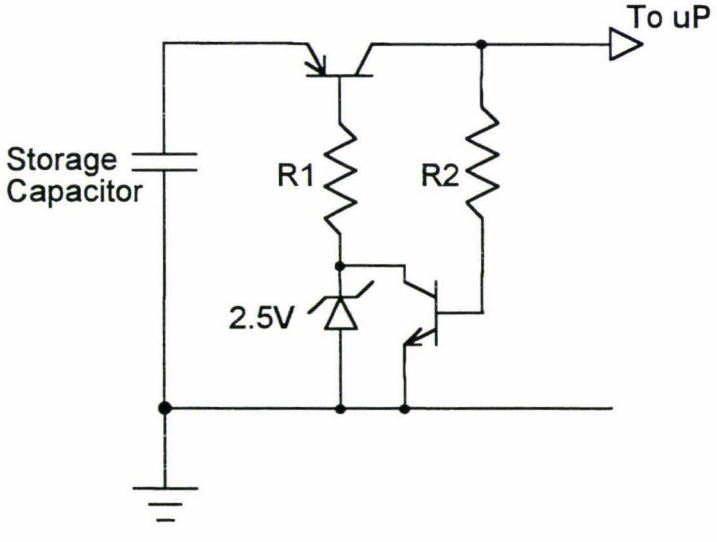


Figure 4.38 Reference In Pass Transistor Base Current Path

This circuit is similar to the previous, yet here the reference diode is now in parallel with the pass transistor.

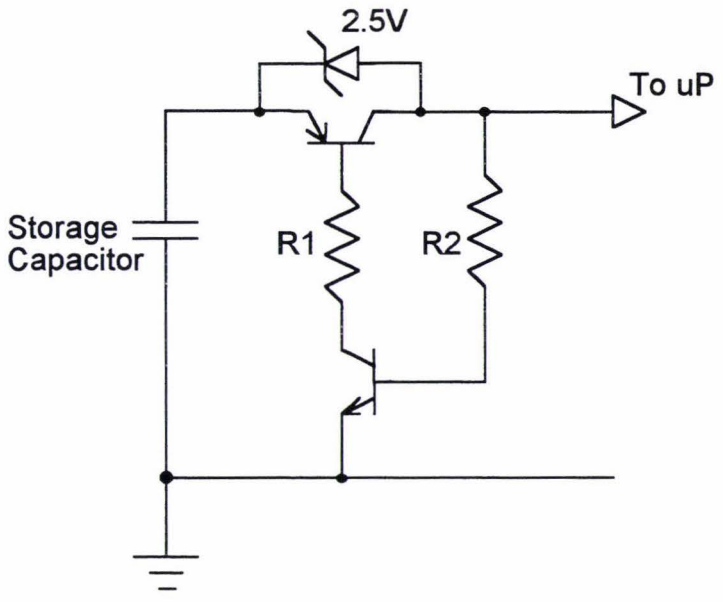


Figure 4.39 Parallel Reference Diode With Hysteresis.

Considering the circuit shown in Figure 4.39 in more detail. Approximately 40μA is adequate as a base current for the pass transistor when the circuit is on. This sets base resistor R1 to 68kΩ. Resistor R2 need only provide a very small current due to the gain provided by two transistors. Making this resistor smaller will mean more power is lost, yet noise pickup could be a problem if this resistor gets too large. For testing the circuit a value of 4.7MΩ was used.

In order for this circuit to work a load is required at the microprocessor. Experiments were performed on this circuit, the results are summarised in Table 4.4

Table 4.4 Circuit Switch On Voltages

Load Resistance	Switching Voltage
2k2	3.0
470k	1.2

These show that if the feedback resistance is close to the load resistance any leakage currents through the parallel reference diode will partly go down the feedback branch. This will then turn the circuit on (at a much lower than desired voltage).

This circuit would work in the slave unit, as the load the microprocessor presents when not running (at low voltages) is in the order of 1 mA and thus enough to ensure the circuit will not falsely trigger.

4.4.1.2 TL431 With Hysteresis

The circuit for this method is shown in Figure 4.40.

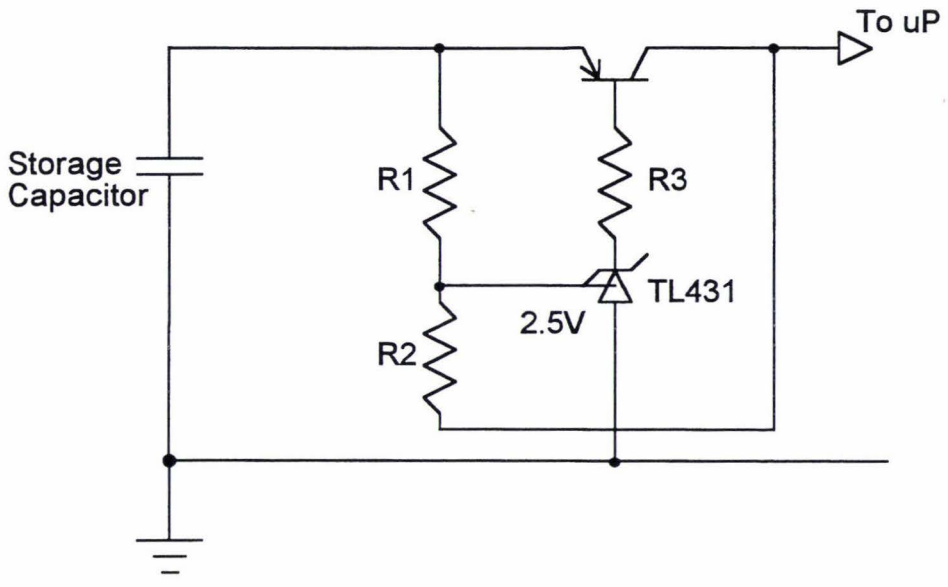


Figure 4.40 TL431 With Hysteresis

The TL431 contains an internal 2.5 V reference and comparator. It will conduct like a zener, when the input reference is >2.5V relative to the device anode. The reference input current is specified as ±4μA and at least 40μA must go down the voltage divider R1R2 to keep divider errors less than 10%. When not operating this current is not a problem, however under normal operating conditions, this divider current is comparatively large. This can be reduced by connecting the bottom of the divider to the pass transistor output. Here when the circuit turns on, this divider bottom will rise up providing positive feedback keeping the pass transistor on.

The failing of this method is the cathode to anode leakage current of the TL431. This leakage is may cause the pass transistor to turn on early.

4.4.1.3 PUT

The Programmable Unijuction Transistor (PUT) would seem a device ideal for this application. It is a four layer silicon device similar to the SCR, yet the gate is brought out on a different layer.

When the gate voltage exceeds two diode drops the PUT will turn on, and once on will stay on until the current through the device falls below the device holding current.

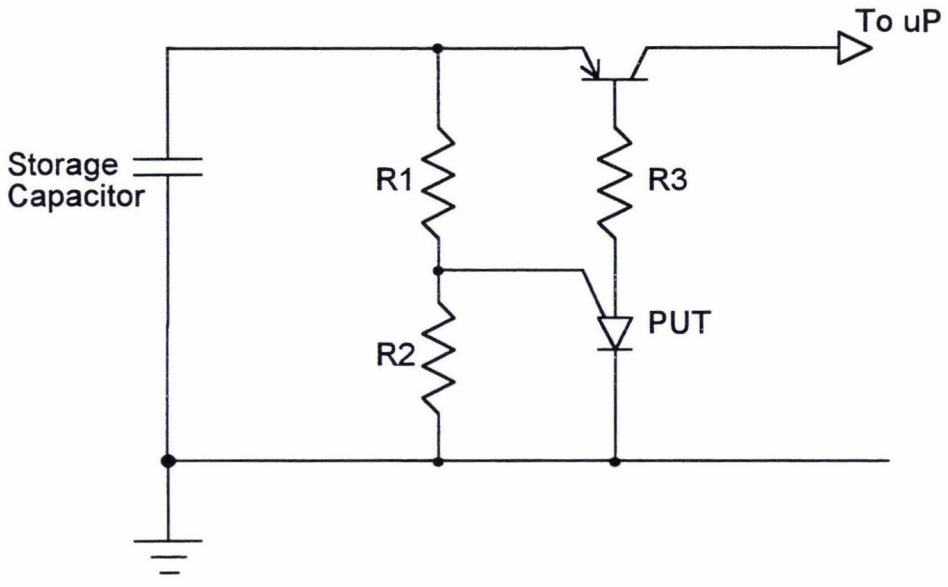


Figure 4.41 PUT Power Up Circuit

The method shown in Figure 4.41 has the advantage that the positive feedback latch on is not influenced by the leakage through the pass transistor. The PUT itself provides this function.

Unfortunately PUTs are not very common, and manufacturers prefer them to be designed out if possible. This, and the inaccuracy in the device switching point rules this method out as an option.

4.4.1.4 Power Supply Monitor IC

This method relies on a dedicated IC to perform the task of measuring the power supply voltage and turning on the pass transistor. Devices such as the ICL7665 are intended for this use and are readily available (Harris Semiconductor (1990)). The ICL7665 has a built in voltage reference and comparator, along with FETs to switch external circuits and permit the use of hysteresis. It is available in an option with a comparator reference accuracy of 2%. The circuit is shown in Figure 4.42.

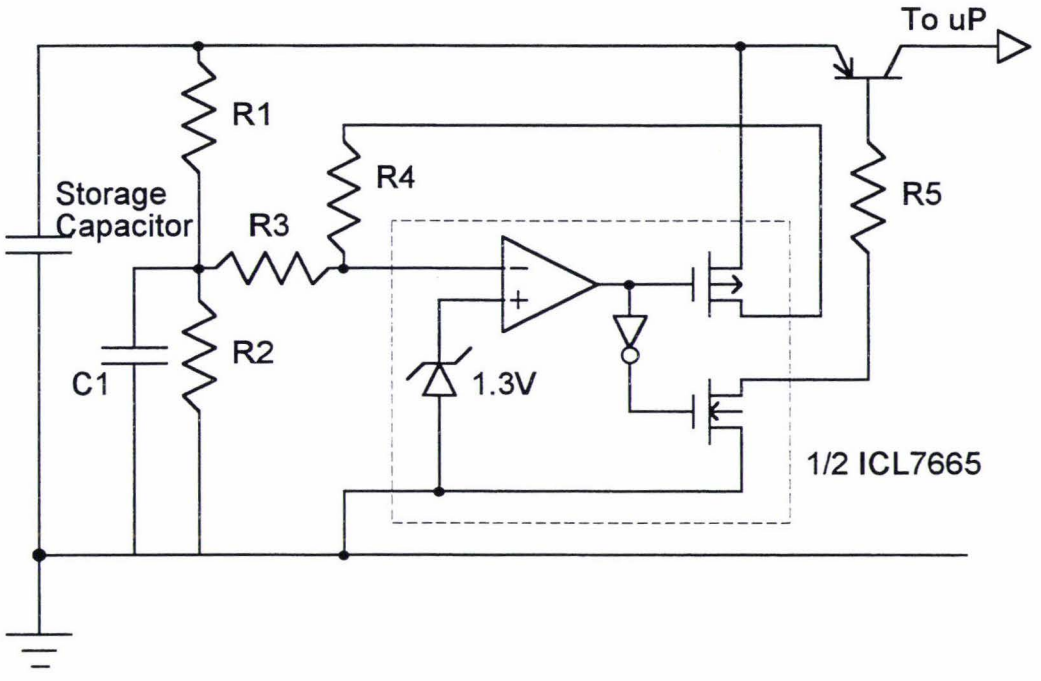


Figure 4.42 Power Supply Monitor IC Power Up Circuit.

To find the resistor values for the circuit the thevenin equivalent circuit is used shown in Figure 4.43.

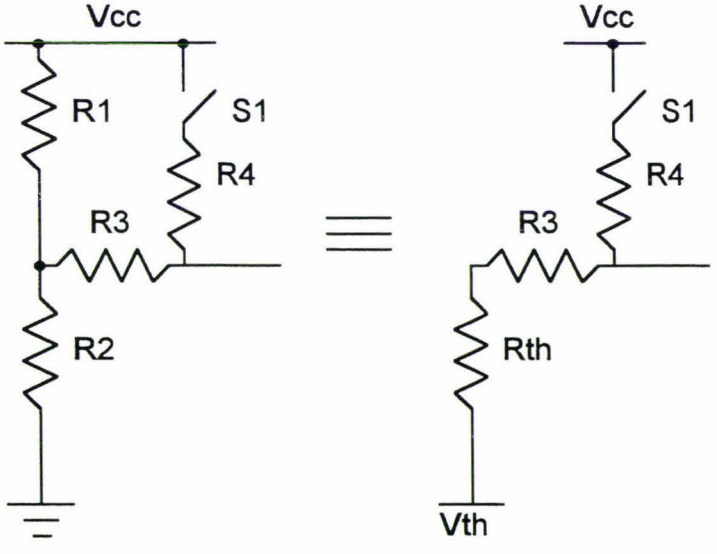


Figure 4.43 Thevenin Equivalent Resistor Circuit

The equations for this circuit are shown in equations 4.59 to 4.62

$$V_{TH} = V_{CC} \frac{R2}{R1 + R2} \tag{4.59}$$

$$R_{TH} = \frac{R1 \times R2}{R1 + R2} \tag{4.60}$$

When hysteresis switch is off:

$$V_{TH} = 1.3 \tag{4.61}$$

When hysteresis switch is on:

$$1.3 = (V_{CC} - V_{TH})\left(\frac{R_{TH} + R3}{R_{TH} + R3 + R4}\right) + V_{TH} \tag{4.62}$$

The leakage current into the comparator is specified as ±10nA maximum. Thus at least 500nA down the divider is desired. Hence R1=1.5MΩ, R2=680kΩ, R3=100kΩ and R4=2.7MΩ.

The power supply chip will now switch on at 4.17V and switch off at 3.0V.

A decoupling capacitor is added to the resistor divider to reduce divider errors caused by noise pickup. A series resistor is also added to the power supply line of the ICL7665 again to reduce noise pickup from the Vcc supply rail. For R5 a base resistor current of approximately 40 - 50μA should be enough to provide sufficient current for the microprocessor circuitry. Thus a value of 68kΩ is adequate for R4.

4.4.1.5 Power Up Method Chosen

The inexpensive power supply monitor IC (ICL7665) method was adopted. The greater accuracy offered by the IC and the unreliable switching of the other circuits due to leakage currents, made the power supply monitor IC option the best alternative. The second half of the ICL7665 was used in the measurement circuit.

4.4.2 Microprocessor Oscillator

Early experimentation with the power consumption of the microprocessor showed that the oscillator was an important part in the overall power consumption of the microprocessor. Minimising the power consumption of the oscillator circuit would greatly improve the performance of the slave unit, by allowing it to effectively measure lower voltages on the fence line.

4.4.2.1 Oscillator Speed

The relationship between oscillator speed and microprocessor power consumption has already been shown elsewhere (Atmel (1994)).

In this application where power consumption has to be kept to a minimum, a slow microprocessor speed is essential. The power consumption for the microprocessor using its own internal oscillator in idle mode is shown in figure 4.44.

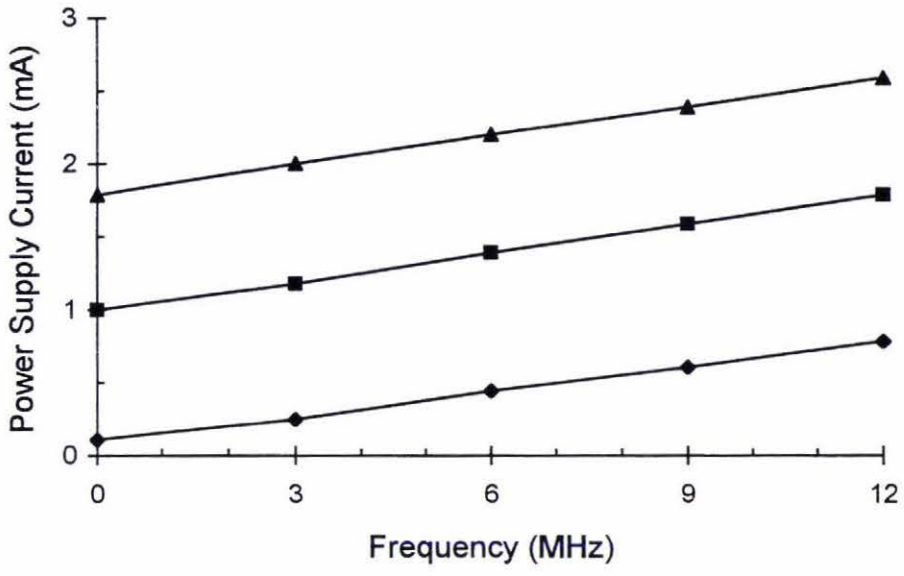


Figure 4.44 Microprocessor Power Consumption Idle Mode

Accuracy in clock speed is important for this application with the microprocessor controlling the transmission frequency, and timeslot discrimination and consequently a crystal oscillator is used.

A 32.768kHz crystal, commonly found in digital watches is a suitable low cost device.

4.4.2.2 Oscillator Circuit

The microprocessor internal oscillator circuit was trialed using the standard oscillator configuration suggested.

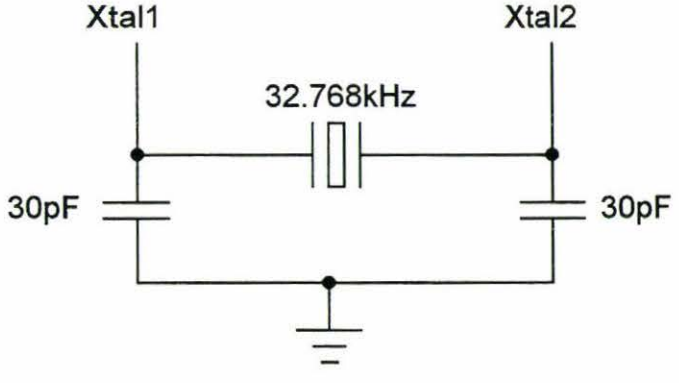


Figure 4.45 Standard Oscillator Circuit.

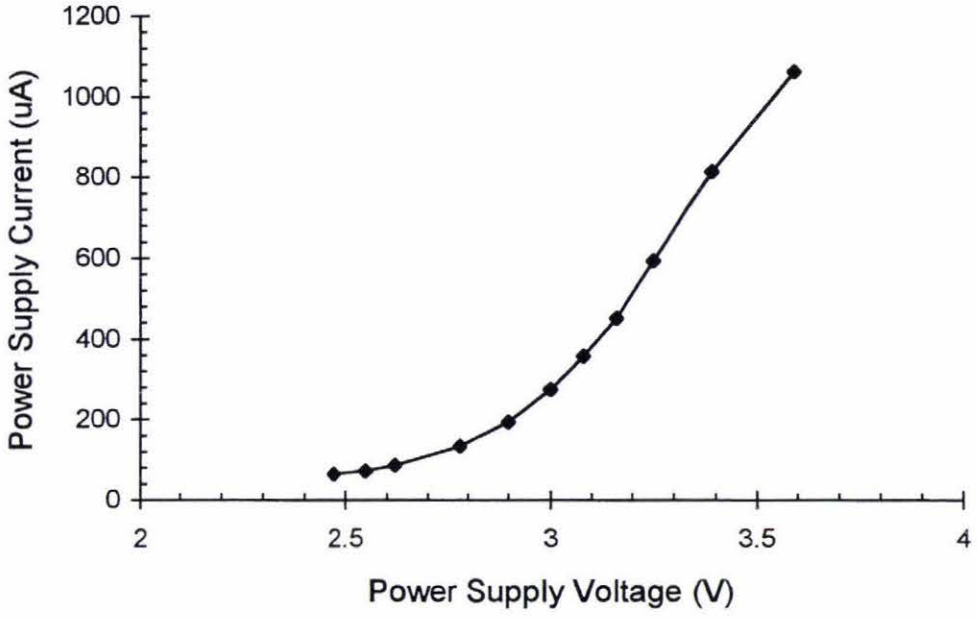


Figure 4.46 Microprocessor Idle Power Consumption (Internal Oscillator)

Figure 4.46 shows that the Icc idle current for 3V is at best 300μA with the standard circuit shown in Figure 4.45. The circuit shown in Figure 4.47 was suggested in order to minimise the oscillator power consumption.

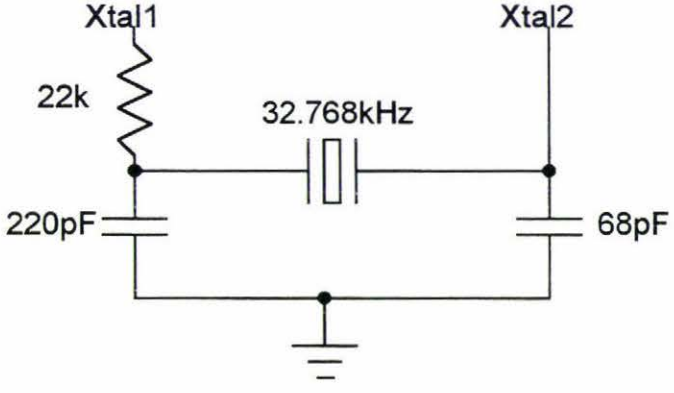


Figure 4.47 Microprocessor Oscillator Circuit (Internal Oscillator).

This circuit has an IDLE current of 75μA. Yet the power consumption for the 89C2051 microprocessor running at 32kHz with an external oscillator drive at 3.0V is specified at 4.3μA (square wave) and 77μA (sine wave) (Atmel (1994)). This implies that if a square wave external oscillator which uses very little current was constructed, the overall current consumption figure could be reduced greatly from 100μA.

The external oscillator circuit chosen is shown in Figure 4.48.

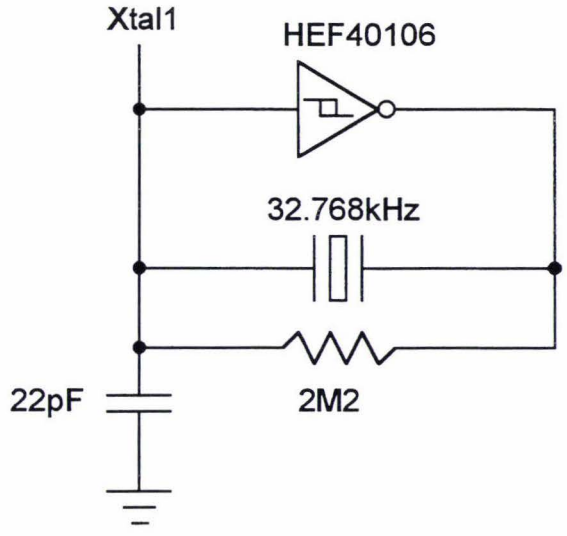


Figure 4.48 Microprocessor Oscillator Circuit (External Oscillator)

The total Power Supply Icc (including HEF40106) at 3.0V is 8.3μA. A vast improvement from 100μA.

4.4.3 Frequency Synthesis

This section outlines the selection of the transmission frequency of the slave unit.

The microprocessor is running from a 32.768kHz clock. Each instruction takes 12 clock cycles to complete and two instruction cycles are required to produce a frequency. Thus the maximum frequency the microprocessor could produce is

$$\frac{32768}{12 \times 2} = 1365\text{Hz}$$

The microprocessor could also produce integer factors of this frequency (ie 1365Hz divided by 1, 2, 3.. etc).

The typical fence line is very similar to a transmission line in electrical characteristics, with distributed inductance, resistance, and capacitance.

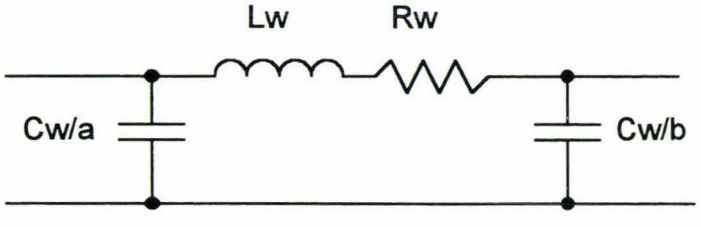


Figure 4.49 Typical Fence Line Equivalent Π Circuit.

The Π equivalent circuit shown in Figure 4.49 for the fence is a low pass filter (McCutchan (1980), Stevenson (1962)). Thus high frequencies will be attenuated much more than low frequencies on the fence line. Chapter 2 contains a more detailed study of the characteristics of the typical electric fence line.

It is also essential that the frequency chosen is not common, such that there is no interference on the line from sources other than the slave units. The main concern here was noise pickup from mains. The system is expected to be sold in the USA and NZ thus 50Hz and 60Hz mains sources must be considered. It is desirable that the frequency chosen is as far away from harmonics of either of these two possible mains frequencies.

A table was constructed to show all of the possible frequencies available from the microprocessor and the distance that each of these was from a harmonic of 50 or 60Hz. This is shown in Table 4.5.

Table 4.5 Transmission Frequency Selection

Mac Cycles	Output Frequency	Closest 50 Hz	Closest 60 Hz	Min Distance	Q
1	1365.33	1350	1380	14.67	46.55
2	682.67	700	660	17.33	19.69
3	455.11	450	480	5.11	44.52
4	341.33	350	360	8.67	19.69
5	273.07	250	300	23.07	5.92
6	227.56	250	240	12.44	9.14
7	195.05	200	180	4.95	19.69
8	170.67	150	180	9.33	9.14
9	151.70	150	180	1.70	44.52
10	136.53	150	120	13.47	5.07
11	124.12	100	120	4.12	15.06
12	113.78	100	120	6.22	9.14
13	105.03	100	120	5.03	10.45
14	97.52	100	120	2.48	19.69

The Q column of Table 4.5 shows the selectivity of a simple bandpass filter centred at the microprocessor output with a 3dB bandwidth twice the minimum distance frequency. The higher the Q the more difficulty in realising the filter. The amplitudes of the mains frequency harmonics would be expected to drop as their frequency increases. These two factors together suggest that the table should be weighted in favour of the higher frequency, low Q values. The low pass frequency characteristics of the fence line have little effect below a few hundred Hz.

Thus the transmission frequency of the slave unit was chosen at 273.07Hz.

4.4.4 Software

The slave unit software has several functions to preform. It must convert the analogue input (time) into a digital value for the measurement method. It must calculate when a transmission is required, and at what time to start this transmission. It must also synthesise the transmission frequency when a transmission is needed.

The software is written in assembler, due to the critical timing nature of the program and the simple nature of the tasks required. Appendix 5 contains the assembler code.

4.4.4.1 Overall Program flow

Figure 4.50 shows the flow diagram for the slave unit software. The program is interrupt based, where an energiser pulse should cause an external interrupt. The software then measures the peak fence voltage and calculates the aggregate values required to determine if transmission is necessary. If transmission is not required, then the software returns from the interrupt and goes back to idle mode. If a transmission is required, the software calculates when that transmission should occur, loading the timer with this value, and enabling the timer interrupt. The software will then return from the interrupt and the microprocessor switch to idle mode. When it is time to transmit, the timer interrupt causes a switch to run mode and the software does the transmission and then returns to idle mode. This ensures that the microprocessor is in idle mode for as much time as possible.

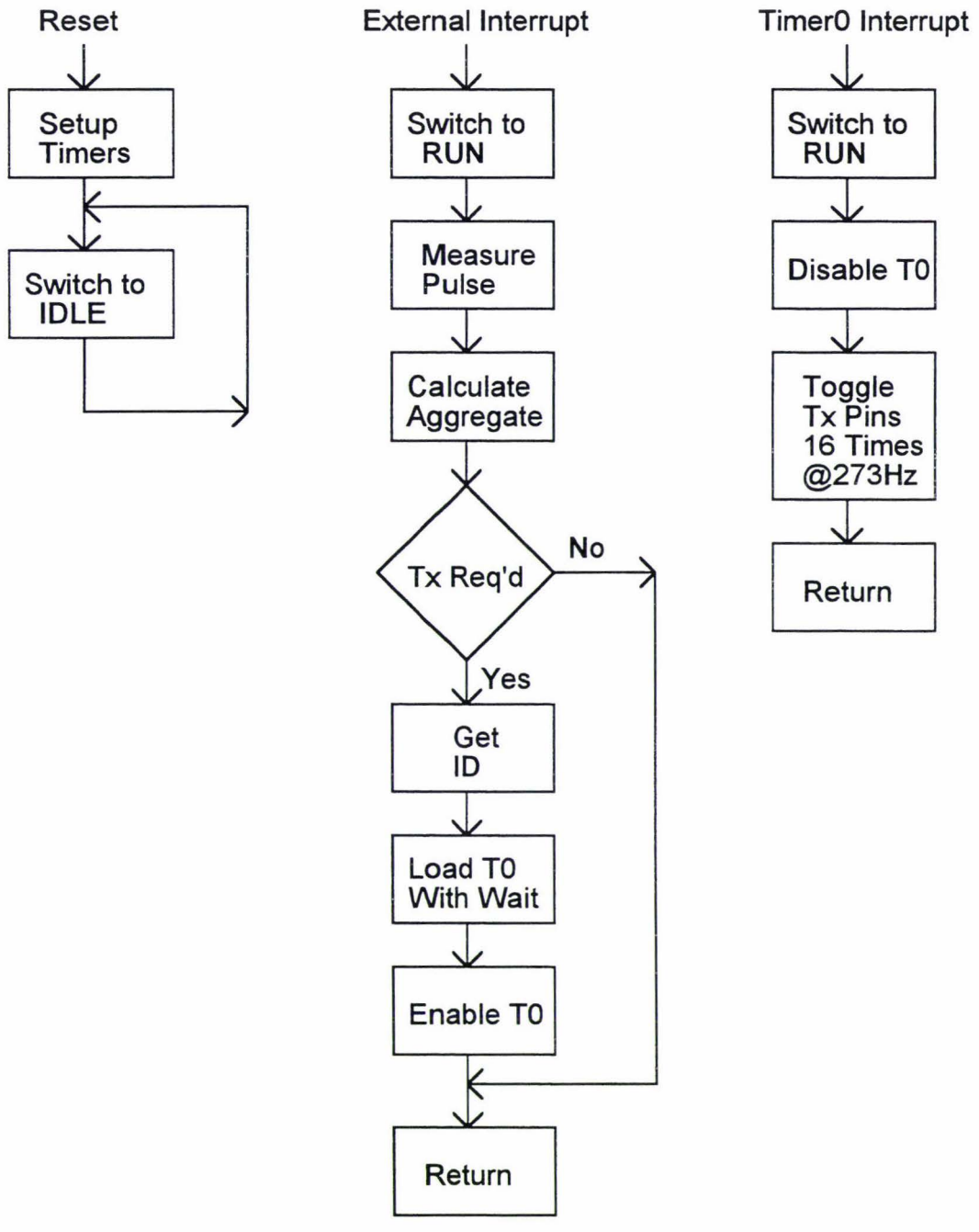


Figure 4.50 Slave Unit Software Flow Diagram

4.4.4.2 Energiser Detection

As the output of the measurement circuit changes as soon as an energiser pulse arrives, it is used to inform the slave that an energiser pulse has arrived. The output of the measurement circuit is connected to the external interrupt 1 I/O pin of the 89C2051. Figure 4.51 shows the measurement / pulse detector circuit connection to the microprocessor.

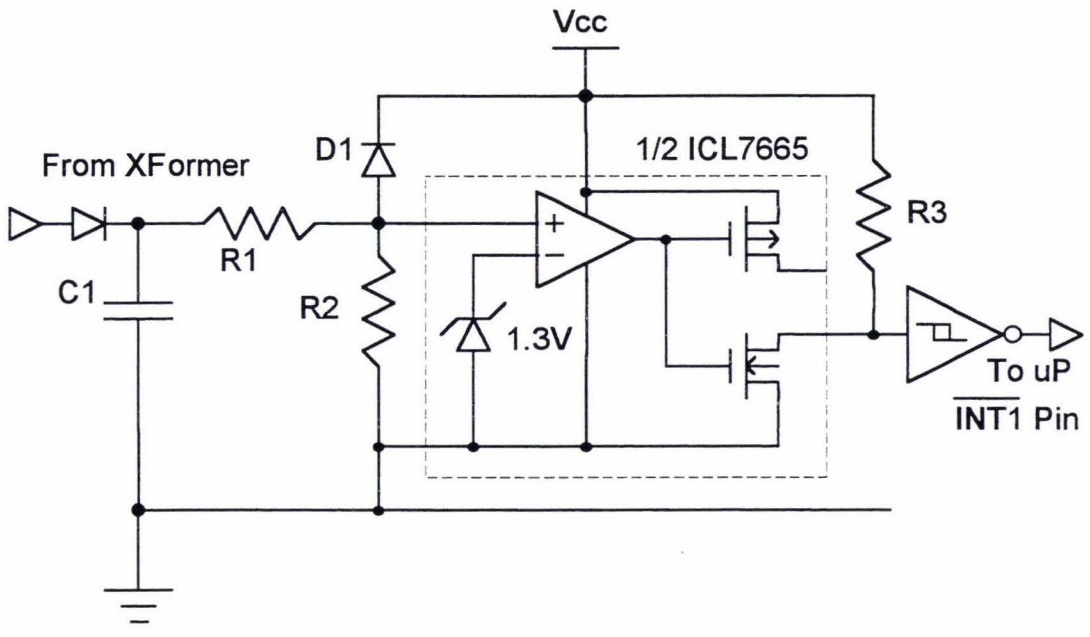


Figure 4.51 Energiser Pulse Detector Circuit.

4.4.4.3 Pulse Amplitude Measurement

To measure the pulse amplitude the software must count the number of machine cycles that the interrupt pin stays low. Thus before jumping to the interrupt subroutine the timer is started to start the count. When the interrupt pin goes high (C1 has discharged) the timer is stopped and the timer value is used as an index to a lookup table to find the result. The reading from the lookup table (named PEAK in the code) is a 8 bit value corresponding to the fraction of threshold that the reading represents (see in Chapter 3). Thus if a 10kV pulse was measured with a 10kV threshold, the result of the table would be FF hex (the threshold). If the pulse had been 5kV the result of the lookup table would be 80 hex.

4.4.4.4 Transmission Calculation

The aggregate of peak fence values is stored in the variable SUM. After adding the peak fence voltage (PEAK) to the aggregate (SUM). A test is preformed to see if SUM has overflowed. If so then a transmission is required. If a transmission is required then the threshold (FF hex) is subtracted from SUM.

4.4.4.5 Slave Unit Identification

In order to calculate in which timeslot each slave unit should transmit, each slave requires a unique identification. This Identification could be directly programmed into each microprocessor as 16 different slave programs, hard wired into the circuit board, or selected by a 4 way DIP switch as shown in Figure 4.52.

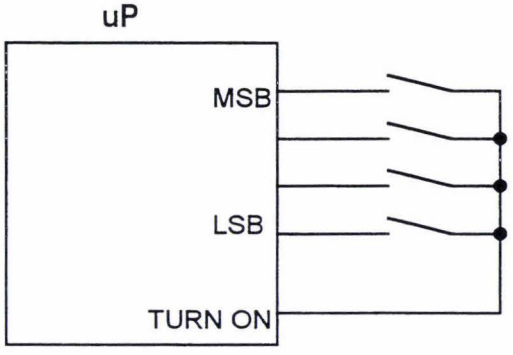


Figure 4.52 Slave Unit ID Selection.

The 4 way DIP switch was chosen allowing only one slave variation and relying on the farmer to change the ID of slaves. To save power the software turns the DIP switch on (by pulling the port pin low) when required to read its ID.

4.4.4.6 Timeslot Discrimination

The 1 second gap between energiser pulses is divided into 17 timeslots. The first timeslot is used to measure the peak fence voltage and each of the final sixteen are allocated to a slave unit of transmission (if required). The microprocessor must calculate the time at which it should transmit according to its ID number. Here it multiplies the duration of one timeslot by the ID+1 and then subtracts the time of the A/D conversion along with a constant for the time to process the code. The result is then loaded into Timer 0 and the timer started. This is the time that must pass until the transmission is required. The microprocessor will then go into idle mode to save on power consumption.

4.4.4.7 Transmission

When timer0 interrupt is called the microprocessor starts toggling the transmission I/O pins at 546.14Hz (2*273.07Hz). This is done for the duration of a timeslot, and then the microprocessor goes back into idle mode with the Timer 0 interrupt disabled.

4.5 PCB Layout

The environment in which the slave unit has to operate is a harsh one. It is important in designing the slave unit layout that interference from electric fence pulses have as little effect on the slave unit as possible. This can be accomplished by surrounding the edge board with ground plane, with a function similar to a faraday cage. Decoupling capacitors are also used wherever possible in order to minimise the effect of noise on the power supply. Decoupling is also included on the resistor divider for the power up circuit.

A single sided board is used for the slave unit to keep costs down.

It is also important to keep the high current power supply part of the slave as far away from the low current measurement and microprocessor part of the slave. The transmission circuitry is placed between the two to separate them.

The measurement circuit is laid out to provide for the possibility of several types of different measurement components. If 100V capacitors are unavailable two 63V capacitors can be used instead, and the divider resistors changed to evenly spread the voltage across the divider resistors.

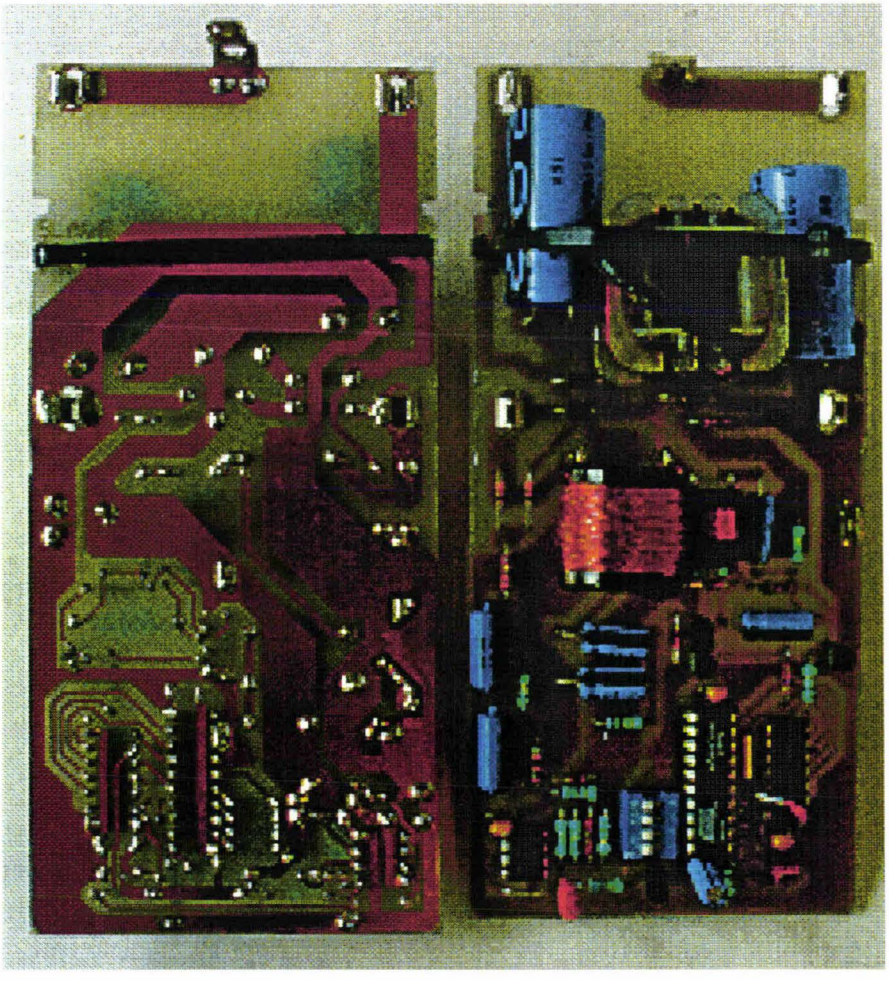


Figure 4.53 Slave Unit PCB

4.6 Packaging

A Speedrite SP4 case that is already used in other Speedrite products that are placed on the fence line is used. This case clamps onto the fence line, and an earth cable is provided to complete the circuit to ground.

The chassis is built to hold a transformer in the back, while allowing a circuit board to be placed on top as well as components.

The outer case is a one piece construction and thus the slave unit is splash proof from above, and rain should not be able to get inside the case. The outer case is solid (3mm thick), and hence stock should not be able to damage the unit easily.

Cable ties are used where required in the unit to hold down components that would otherwise be free to move and fatigue at the joints.

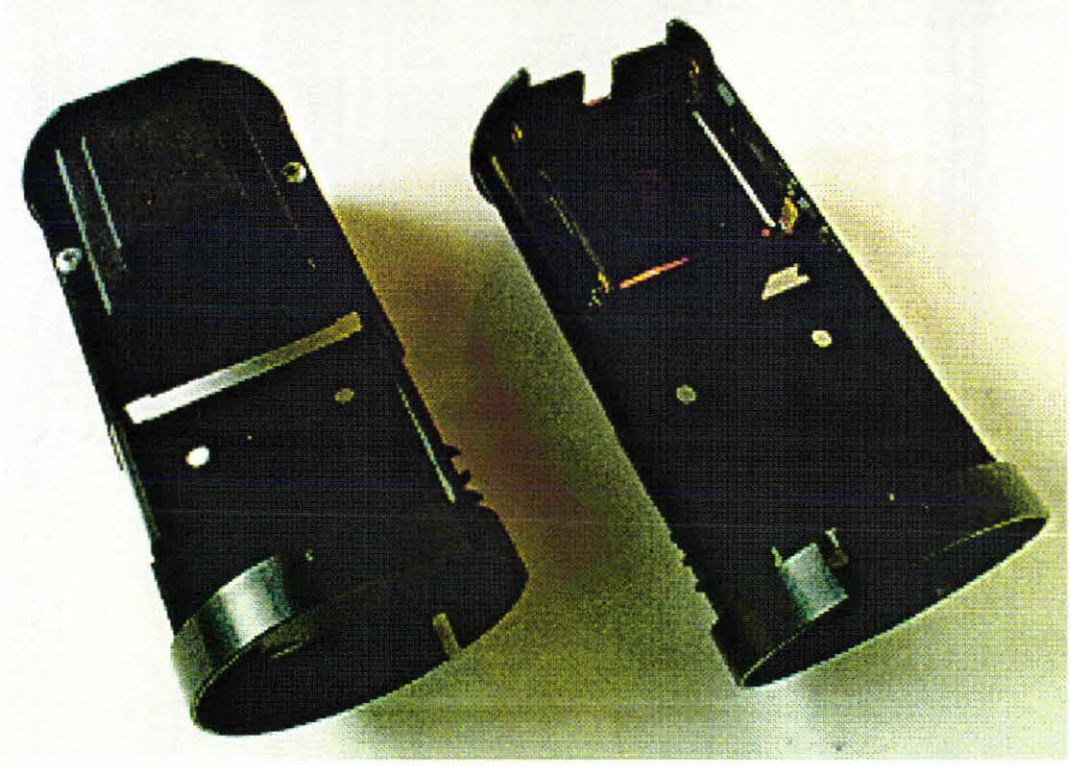


Figure 4.54 Slave Unit Chassis Packaging

Figure 4.55 Slave Unit Case Packaging

Chapter 5
The Master Unit

This chapter is concerned with the construction of the master unit for the Electric Fence Monitoring System (EFMS) described in Chapter 3. The master unit is connected to the fence line and detects transmissions from the slave unit to decode the voltage at the slave unit. The slave unit transmissions are a 273.07Hz frequency burst in a 16th of a second timeslot allocated according to the slave unit identification number.

The decoded voltages are displayed on the master unit screen for the user to view. The master unit includes a keypad through which the user can specify the voltage level to be monitored at each of the slave points. If the voltage falls below the specified value the master unit indicates an alarm condition.

Major aspects of the master unit design are summarised below.

If the impedance of the energiser is low then the slave output amplitudes will be low and may vary between different slaves. Section 5.1 investigates the effect of the energiser impedance on the slave unit transmissions and the possibility of removing variation in amplitude between slave units.

Fundamental to the effective operation of the master unit is a filter to detect the slave unit transmissions. This filter must reject all of the noise surrounding the 273Hz slave unit frequency bursts. Filters with narrow passbands are common, but necessarily have a large rise time associated with the narrow passband (Haykin (1989)). In this application if the rise time is too long, the frequency burst will be over before the filter output has risen to a detectable level. A novel approach using switched capacitor filters and multiple frequency clock speeds is presented in Section 5.2.

Section 5.3 details a gain control system for the filter, that will automatically adjust its setting depending upon the operating environment of the EFMS.

Section 5.4 details the design of the power supply for the slave unit. It discusses the isolation requirements for various parts of the circuit and how these are met with the power supply.

Sections 5.5 to 5.8 detail the user interface, microprocessors, software, alarms, PCB layout and the master unit packaging.

A block diagram of the master unit is shown in Figure 5.1.

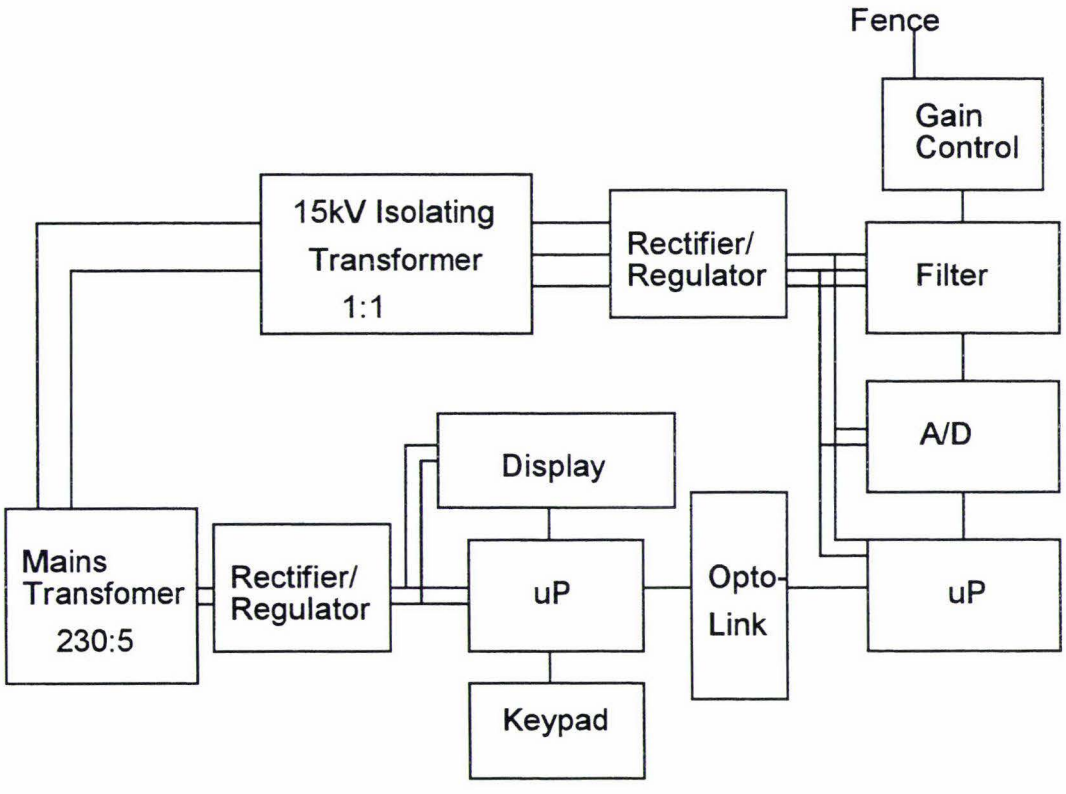


Figure 5.1 Master Unit Block Diagram.

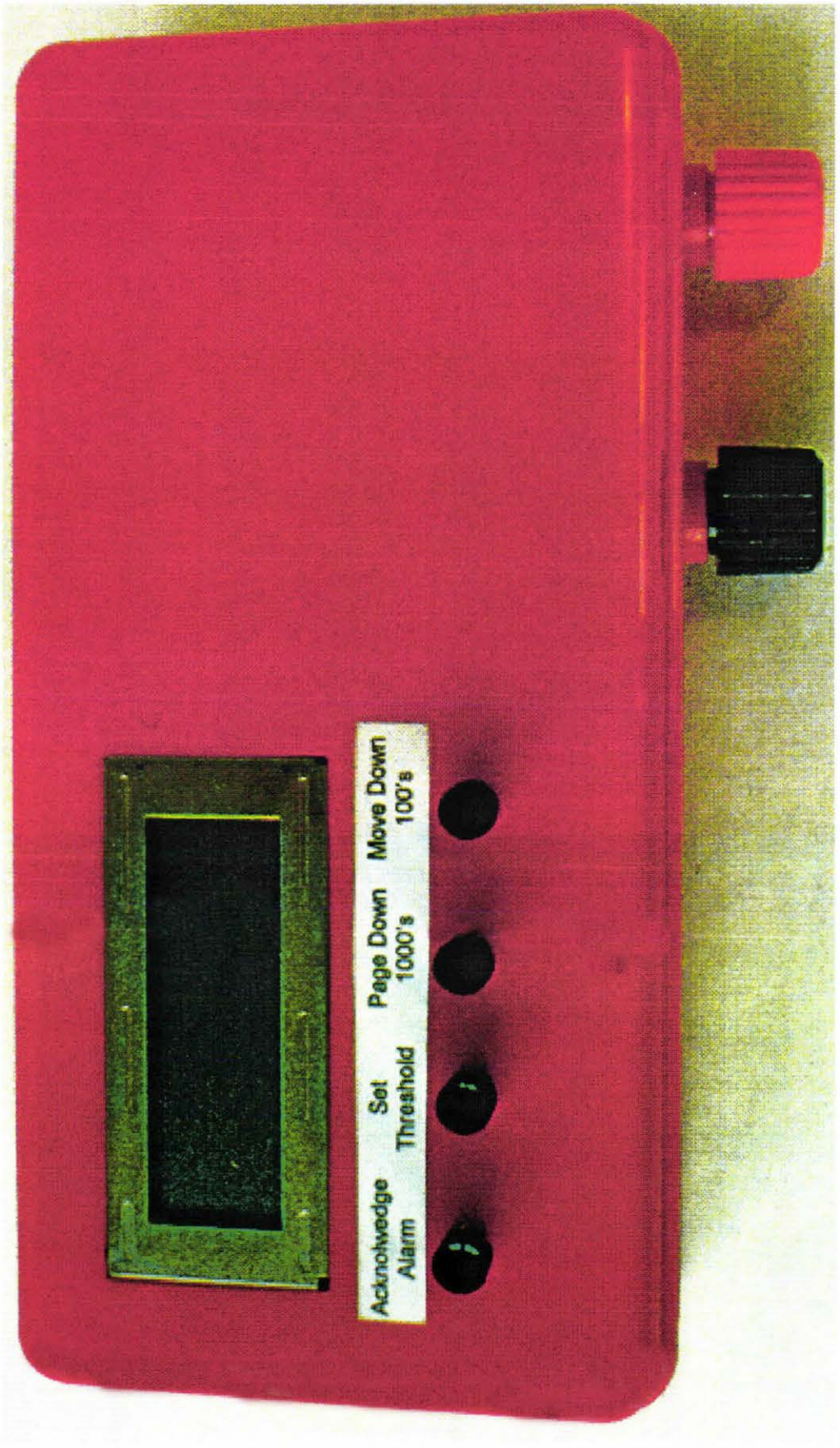


Figure 5.1 The Master Unit

5.1 Attenuation Of The Slave Transmissions

The impedance of the energiser at low frequencies may be too small for the slave unit to produce a large transmission signal at the master unit. Fence energisers that include RFI suppression components may have a capacitor as low as 7μF across the output transformer. When referred through the energiser output transformer and onto the fence, at 273Hz this impedance would be as low as 6000Ω. More significant in some cases is the inductive impedance of the large output transformer in the large energisers, which may be as low a 100Ω (shown in section 5.1.1). Figure 5.3 shows the transmission output circuit and how the low impedance of the energiser is a major factor in the amplitude of the received transmission. The impedance values are those when referred outside the slave unit transformer onto the fence.

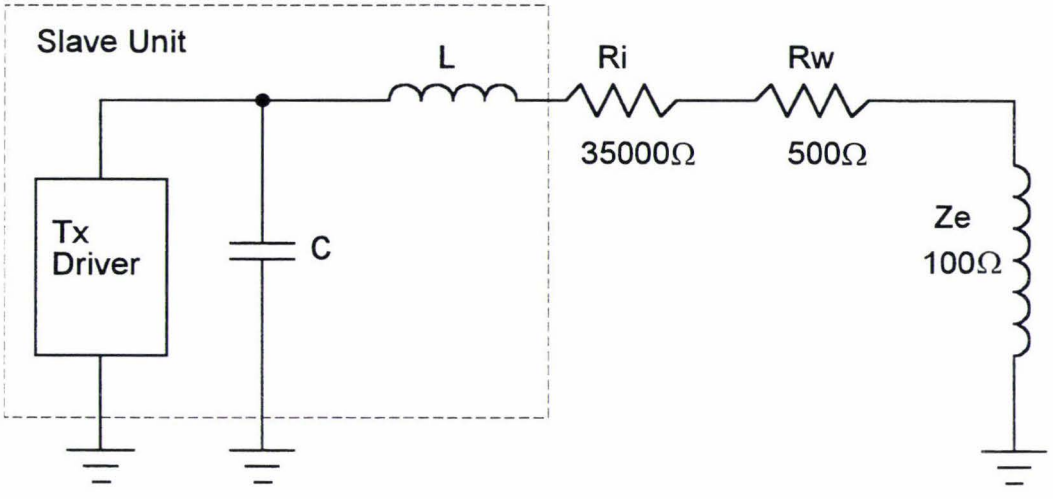


Figure 5.3 Slave Unit Transmission Equivalent Circuit.

It is thus important to consider methods for increasing the energiser impedance and to measure the impedance of various energisers.

5.1.1 Impedance Measurements

Experiments were carried out on a selection of transformers to measure their impedance at 273Hz. This will indicate the expected amplitudes of slave responses. Figure 5.3 shows the setup used.

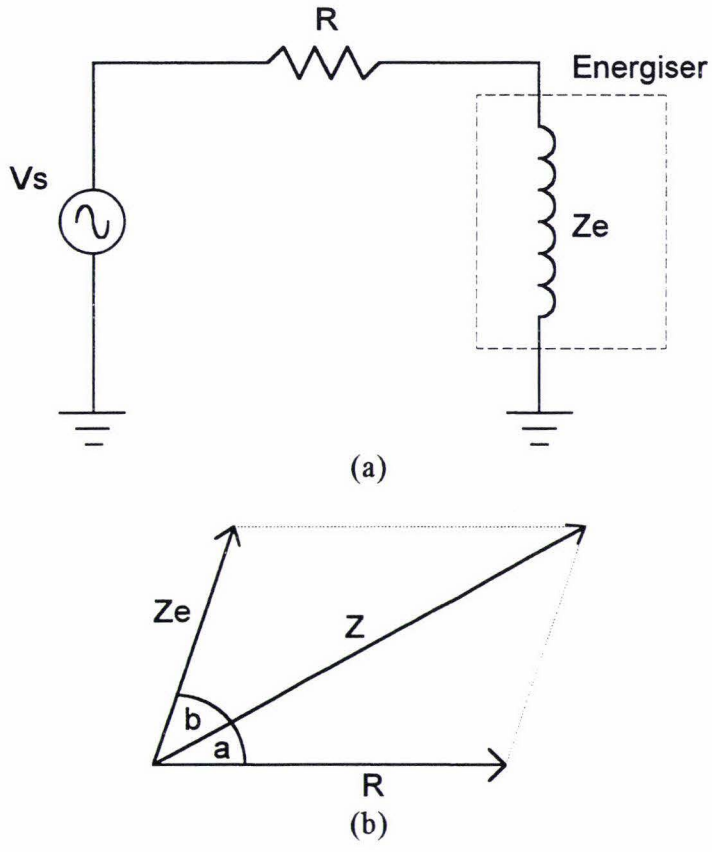


Figure 5.4 Energiser Impedance Measurement.
(a) Experiment Circuit
(b) Phase Diagram

where a is the phase angle between the Supply voltage and the current, and b is the phase difference between the energiser voltage and the supply voltage. The energiser impendence is given by equation 5.1.

$$|Z_e| = \frac{R \times |V_{Ze}|}{|V_R|} \tag{5.1}$$

Four different energiser configurations were tried. The first was the Speedrite SM12000 (a high power energiser) on a low power setting, the second a Speedrite AN90 (a very low power energiser), the third a Speedrite HB150 (a low power energiser) and the fourth the SM12000 on a high power setting. The results are shown in Table 5.1.

Table 5.1 Energiser Impedance.

Energiser	Impedance
SM12000 Low Power	1974Ω
AN90	1650Ω
HB150	1200Ω
SM12000 High Power	223Ω

The smallest of these impedances (223Ω) is very much smaller than the source impedance of the slave unit transmitter and thus options for increasing the energiser impedance should be considered.

5.1.2 Impedance Adjustment Options

The master unit could be constructed such that it is placed in series with the energiser and the rest of the fence. Thus when the pulse was present a switch could be closed to drop the series impedance, and the switch opened when the pulse was not present in order to increase the impedance, such that the transmission signal is detectable.

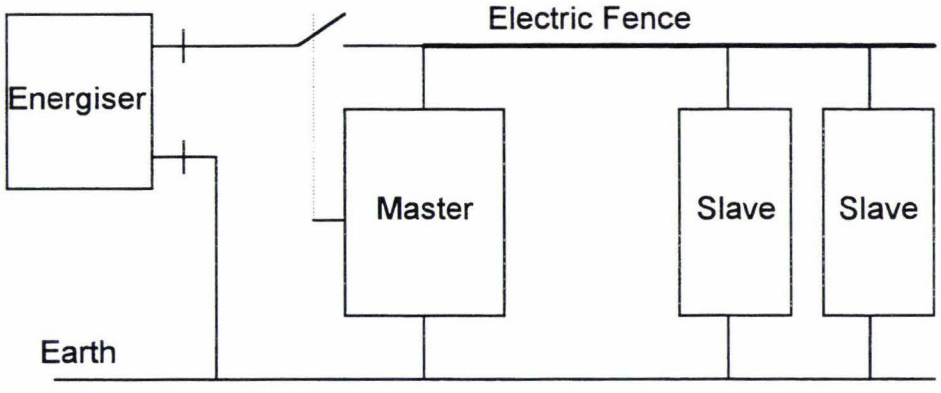


Figure 5.5 Master Unit Impedance Switch Configuration.

This configuration (shown in Figure 5.5) has the restriction that the master unit must be located between the energiser and the electric fence which is being monitored. The switch would have to be able to repeatably withstand 10kV without deterioration. This system would have the advantage that the energiser impedance could be monitored and adjusted when required, yet it places restrictions on the location of the master in the electric fence system.

If an automatic switch could be created that would do the impedance switching according to the voltage on the fence, this would allow the master unit to be placed at any location on the fence, however it would require another unit for the system's operation. Figure 5.6 shows this system configuration.

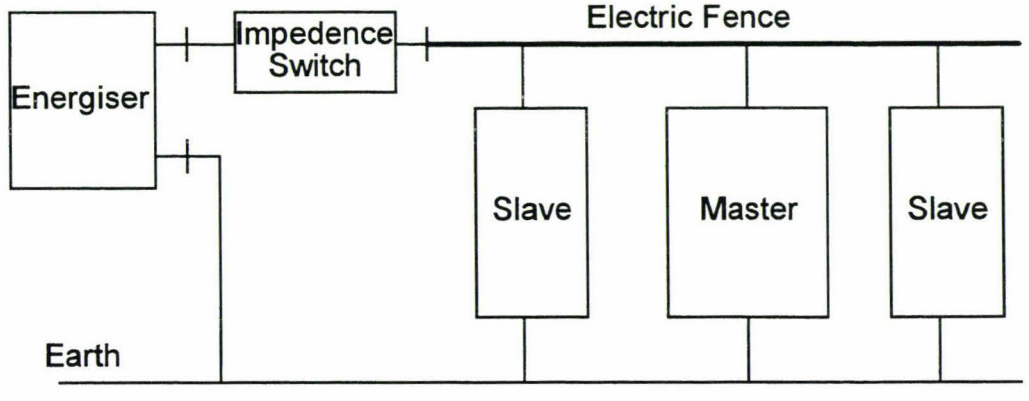


Figure 5.6 Series Fence Impedance Switch.

An impedance switch circuit must be able to dissipate the entire fence pulse in the case of a short circuit from the fence to ground. As the most powerful energisers are 25 Joule energisers, operating at less than one pulse a second, the impedance circuit must be able to dissipate 25 Watts.

The impedance circuit must not significantly drop the voltage on the fence under normal operating conditions. Thus during the fence pulse the device must have a low impedance, and it increase the impedance to ground for voltages <100V when the fence pulse is not present.

Two possibilities exist for the impedance circuit: varistors and diodes.

5.1.3 Slave Transmission Amplitude Measurement

Experiments were also performed using a slave unit and the energisers to measure actual slave responses, rather than rely on the results of simulation to gauge the effect of different impedances. Figure 5.7 shows the setup used.

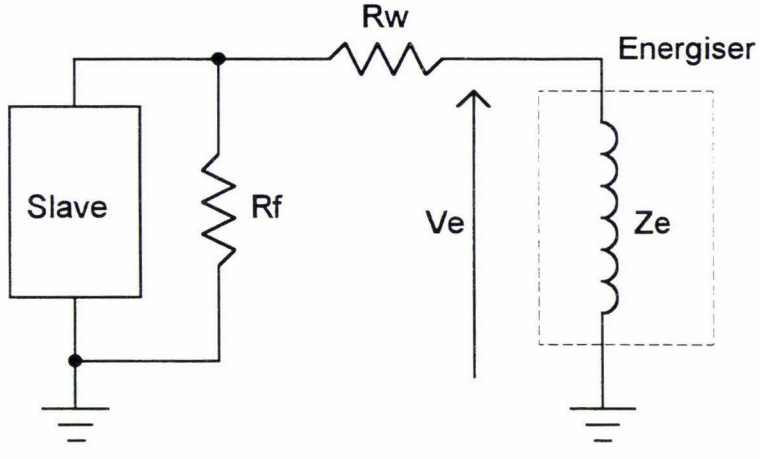


Figure 5.7 Slave Unit Transmission Amplitude Measurement.

Rs is included to represent a short fault resistance on the fence. The slave unit is expected to continue to operate when the fence voltage is reduced to 1.5kV at the slave. Thus Rs is such that it divides the energiser pulse with the resistance of the wire. The response amplitude was also measured under normal operating conditions.

Table 5.2 Slave Unit Transmission Amplitude.

Energiser	Rw Ω	Rf (Ω)	Ve (mVpp)
SM12000 Low Power	0	∞	2500
	470	∞	2500
	470	150	150
HB150	0	∞	1600
	470	∞	1600
	470	150	160
SM12000 High Power	0	∞	300
	470	∞	300
	470	150	60

The results in Table 5.2 show that R_W has no practical effect on the transmission amplitude unless a fault resistance is present. When the fault resistance is included the transmission amplitude can drop as low as 60mV.

The slave unit transmission amplitude was also measured for various terminating resistance. The results are shown in Figure 5.8. These show that the maximum slave transmission amplitude approaches 100V peak to peak, yet the source impedance of the slave unit is so high that in practice the low energiser impedance permits a slave unit transmission amplitude of only a couple of volts peak to peak maximum.

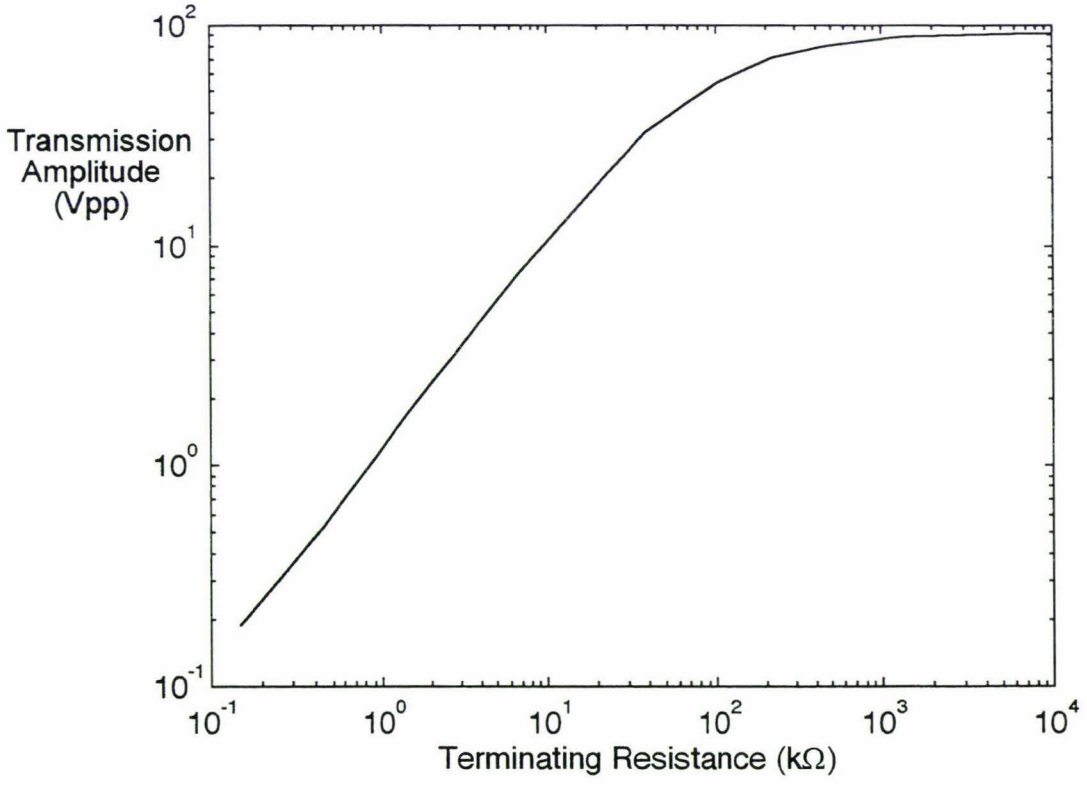


Figure 5.8 Slave Transmission Amplitudes

In order to increase the transmission amplitude to 1V peak to peak the impedance of the energiser would have to be a minimum of 1kΩ. This would make the signal easily

detectable with a master unit filter. It would mean that any fault resistance to ground, while it may drop the fence pulse should not be less than approximately 1kΩ.

If it is desired to measure the transmissions at the lower amplitudes, and thus avoid the requirement of increasing the energiser impedance, a more complicated filter is required.

5.1.4 Final Design Criteria

The impedance switch means that the amplitude of the pulses will vary very little in amplitude from the different slaves. The major difference between any two transmissions will depend upon the slave unit power supply voltage and not the electric fence configuration. Because of this the filter in the master unit will not have to detect pulses of varying amplitude, and can be much simpler. The amplitude of the pulses will also be much larger and thus the bandwidth of the filter may be increased.

Problems may arise with the impedance switch under non-normal operating conditions. If the line becomes partially shorted (by grass or trees etc), then the amplitude of all of the signals will drop. The signals may also start to vary slightly in amplitude between slaves. An impedance switch also introduces another unit to the system, which from a users point of view is detrimental to the product desirability.

In the final design no impedance switch is used. Thus this requires a much more complicated filter in the master unit. Here the master unit is expected to detect the slave unit pulse regardless of the impedance of the energiser. A design limit for the energiser impedance has to be selected, and the following conservative limit was chosen. Minimum energiser impedance of 100Ω and maximum wire resistance (includes ground resistance) of 500Ω.

An example of this operating condition is shown in Figure 5.9.

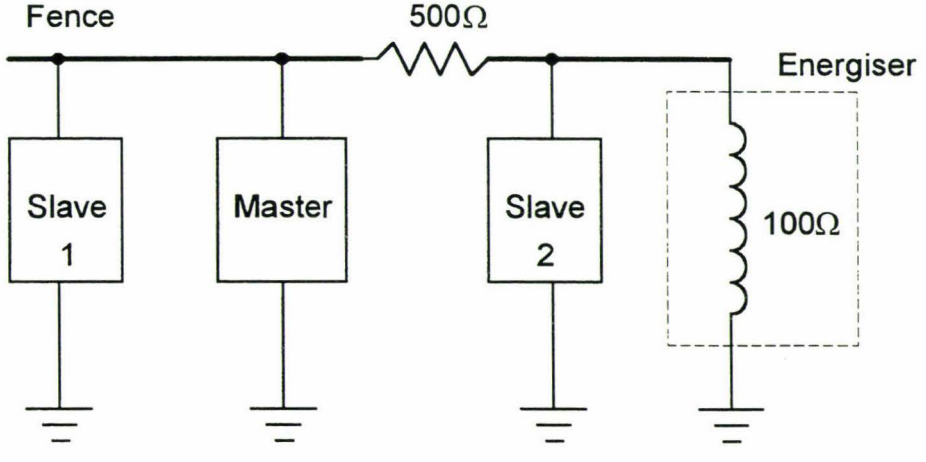


Figure 5.9 Impedance Design Criteria.

In the situation shown in Figure 5.9 the signal produced by slave 2 will be low as the output impedance is only 100Ω. When referred back into the slave unit this is an

output impedance of only 0.01Ω. For slave 1 the output impedance is much higher and thus the signal is also higher. Thus the two signals produced will vary in amplitude significantly.

This system is more customer friendly, as no separate device is needed and the energiser can be placed anywhere in the system, however the filter required in the master unit is more complex.

5.2 Signal Filtering.

A filter is required so that the slave unit transmissions can be distinguished from the noise on the fence line. This filter is tuned to the slave transmission frequency of 273.07Hz and should be narrow enough to reject harmonics of mains noise. This section outlines the problems of using a traditional filter and how the problem of the slow rise time of the traditional filter is overcome in this application.

Firstly, the requirements of the master filter are presented and how this filter is constructed traditionally. Following this the problems with this traditional filter are discussed, and solutions are given. Filter experiments are presented and a final configuration with reduced residual noise pickup outlined.

The post filter circuit needed to interface to the microprocessor is given.

5.2.1 Filter Requirements

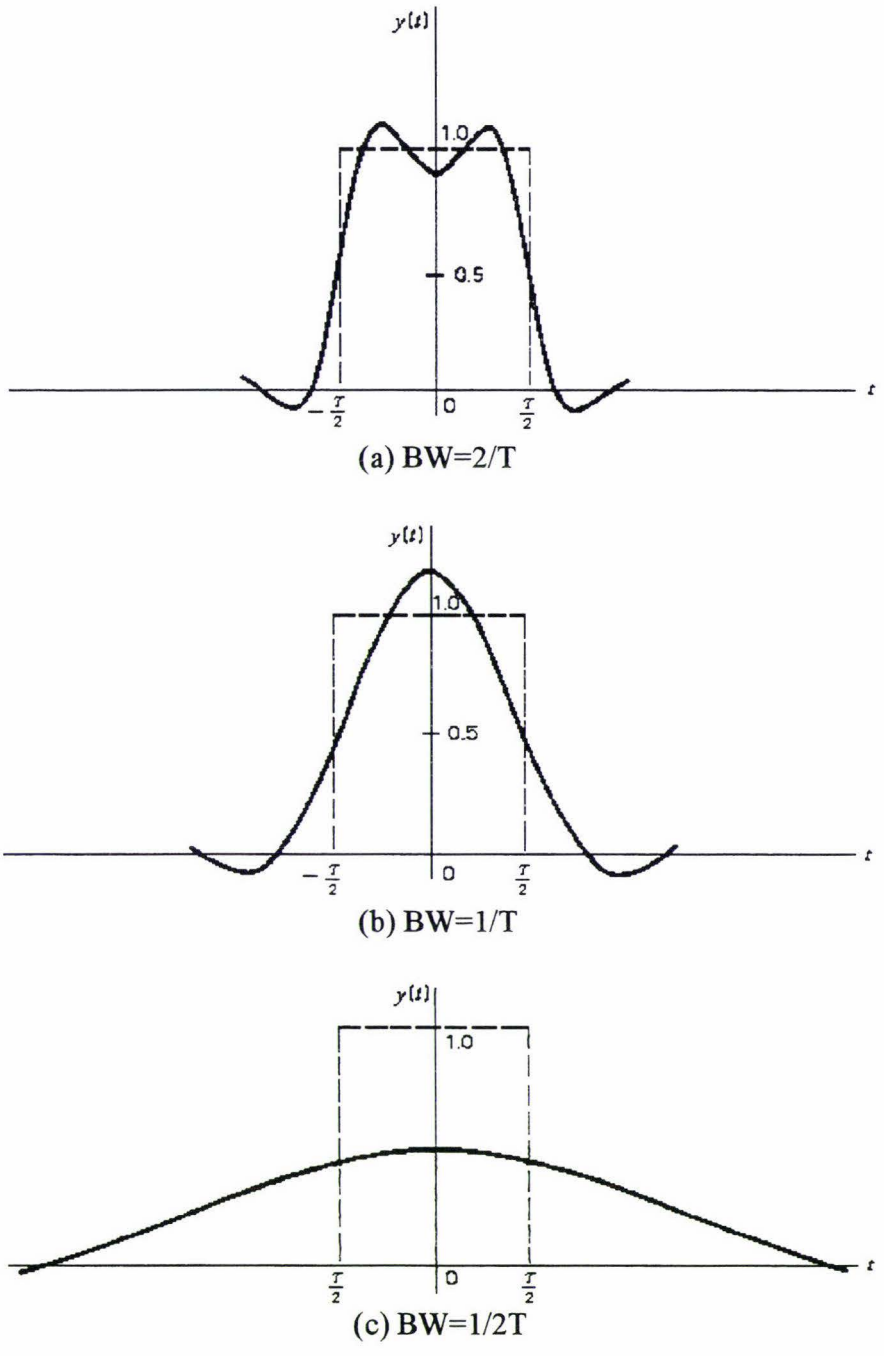
The filter is a band pass filter centred at 273Hz. The filter must be able to detect 273Hz slave transmissions with an amplitude as low as 50mV and must be able to withstand variations in amplitude between slave transmissions of up to six times.

The filter must block mains pickup of both 50Hz and 60Hz of upto 100V peak to peak, along with the harmonics of these mains frequencies. The two closest harmonics are 250Hz and 300Hz. The amplitude of these are a maximum of 5V peak to peak. This assumes a 5% of the fundamental amplitude for the 5th harmonic, which is considered very much a worst case scenario.

The filter must block and be able to withstand the energiser pulse itself. The frequency of the energiser pulse is generally around 10kHz.

The filter must also allow timing distinction of the signals that it passes. This ensures correct identification of slave unit timeslots.

In order for the filter to detect a tone burst in a timeslot, the bandwidth of the filter must be at least as large as the inverse of the length of the timeslot. Three cases are shown in Figure 5.10 of the relationship between filter bandwidth and signal duration (Haykin (1989)). These show the envelope of the filter output. Note that for the bandpass filter the true output would be this envelope multiplied by cosine($2\pi ft$).



(a) BW=2/T

(b) BW=1/T

(c) BW=1/2T

Figure 5.10 Slave Unit Transmission Amplitude Measurement.
From (Haykin (1989)).

The duration of the 273Hz frequency bursts are of 1/17th second each, thus the bandwidth of the filter must be at least 17Hz. This is possible as there is a 50Hz band gap for the filter from 250Hz to 300Hz. A series of simple bandpass filters all tuned to 273Hz with a attenuation of 34dB at 250Hz does not allow a 17Hz bandwidth.

The filter requirements are shown in Figure 5.11.

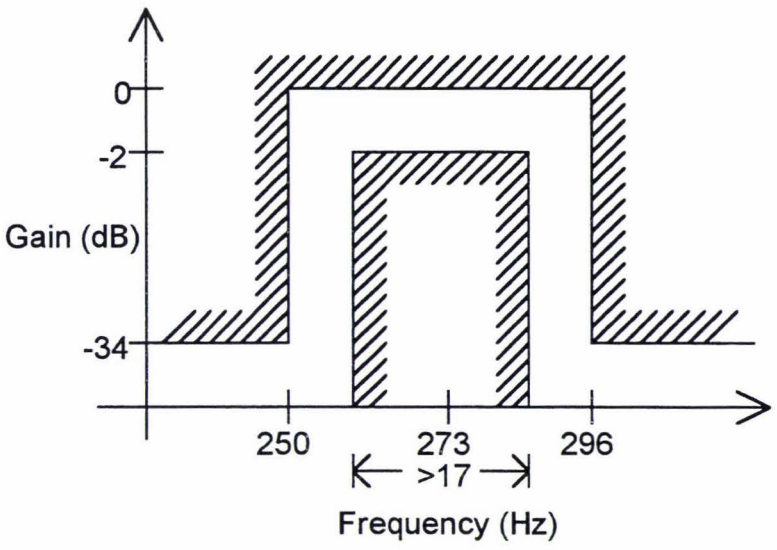


Figure 5.11 Master Filter Requirements.

5.2.2 Traditional Filter Design

A chebyshev filter (constructed of several bandpass filters each at slightly different frequencies) allows a step rolloff, and a wide passband, such that the bandwidth can be larger than 17Hz and the attenuation at 250Hz can be 34dB. An example is shown in Figure 5.12.

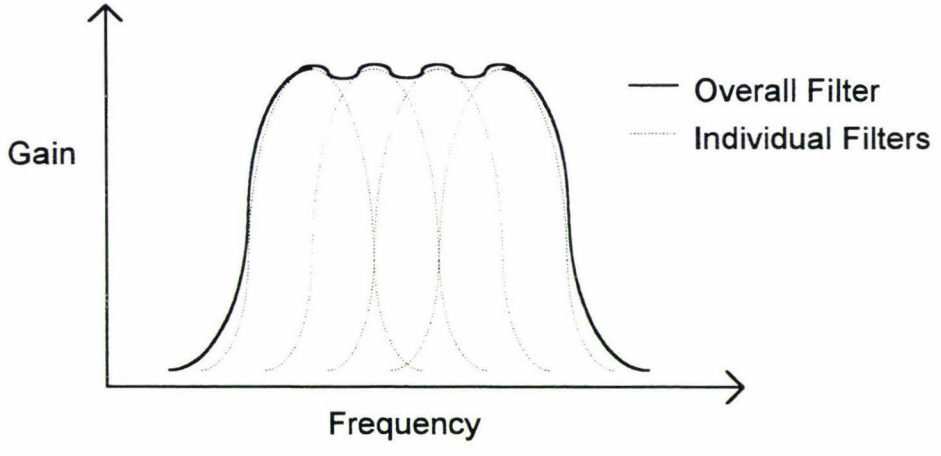


Figure 5.12 Chebyshev Filter.

An eighth order filter was selected for economic and performance reasons. The MF10 Universal Monolithic Switched Capacitor Filter provides a cheap solution to creating a high order bandpass filter. Each MF10 contains two second order switched capacitor filters, thus two MF10 ICs are required.

A tabulated design procedure was used to define the filter characteristics (Williams (1990)). A filter with a 2dB passband ripple and an attenuation of 34dB at 250Hz was selected. The resulting theoretical filter is specified by:

$Q_1 = 46.768$

$f_{01} = 268.5$ Hz

$Q_2 = 46.768$

$f_{02} = 277.7$ Hz

$Q_3 = 112.95$

$f_{03} = 262.0$ Hz

$Q_4 = 112.95$

$f_{04} = 287.1$ Hz

where f_{0n} is the center frequency of the n'th filter and Q_n is the selectivity of the n'th filter. The result is a chebyshev filter with a bandwidth of 23.5Hz.

5.2.3 Traditional Design Problems

In this filter the bandwidth is approximately equal to the inverse of the duration of the frequency burst. A mathematical package was used to predict the response of an ideal band pass filter with bandwidth 23.5Hz to a 17th of a second 273.07Hz frequency burst. The result is shown in Figure 5.13.

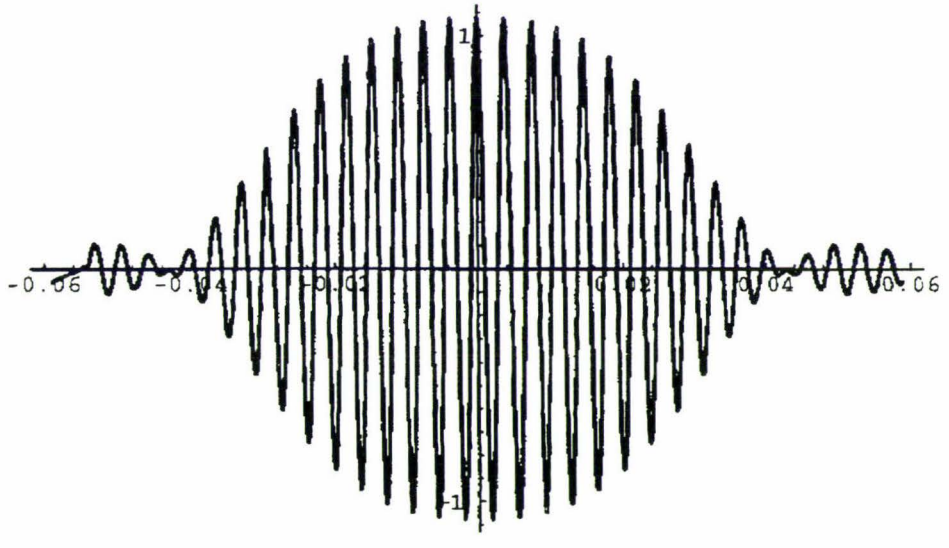


Figure 5.13 Theoretical Filter Output.

The filter then is similar to that shown in Figure 5.10(b). The rise time for the output of the filter is approximately half the duration of a slave transmission timeslot. If the amplitude of all of the slave transmissions, when viewed at the master unit were constant, then this filter would work. The design criteria shown in Figure 5.9 show an energiser impedance of 100Ω at 273.07 Hz and 500Ω wire resistance. These specifications mean that the slave replies could vary in amplitude by up to six times with certain combinations of energiser impedance, wire impedance and system layout. As shown in the filter response graph in Figure 5.13, the input signal from one timeslot is spread out over more than one timeslot at the output. This can cause

problems in detection of small amplitude signals relative to adjacent timeslot amplitudes. The worst case is shown in Figure 5.14, with measurements from the actual filter.

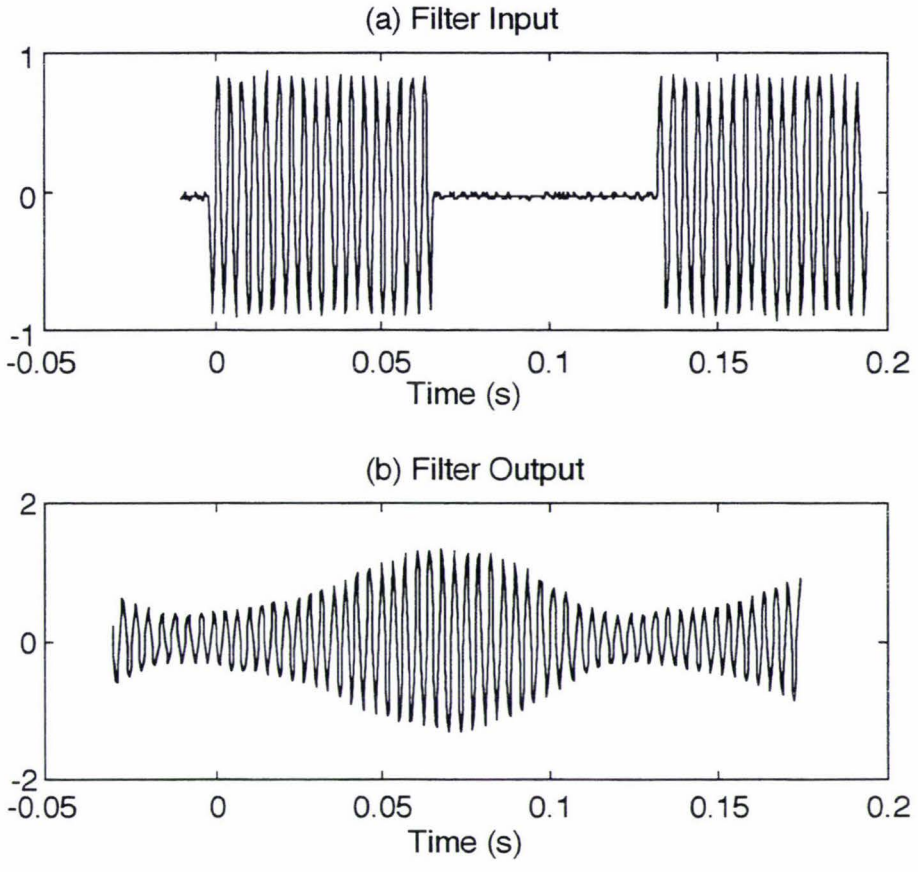


Figure 5.14 Timeslot Filter Response.

In the case shown in Figure 5.14, the filter input is shown in the first graph and the output of the traditional chebyshev filter is shown in the bottom half of the graph. The case shown is for alternate slave unit transmissions. The first transmission causes the 'bulb' in the output. After the transmission finishes, the output starts to decay, but does not disappear quickly. The decay will take as long as a timeslot before the transmission two timeslots away starts to cause a rise in the output. Thus the presence of a low amplitude transmission between the two higher amplitude transmissions may be indistinguishable from ringing and therefore is not detectable at the filter output.

The solution to this is to 'reset' the filter after each timeslot. If the output of the filter is set to zero at the end of a timeslot, then a rise in the filter output can only be the result of a transmission in that timeslot, or system noise. Thus a small portion of the end of a timeslot would be dedicated to resetting the filter output.

5.2.4 Filter Solution Options

Several options for resetting the filter were considered. Three options are presented here and their merits discussed

5.2.4.1 Use Of Multiple Filters

This method would use more than one filter in the circuit, yet only one would be connected to the fence during any one timeslot. This would allow the output of the filter to decay in the timeslot directly after it was used. Alternate filters could then be viewed in alternate timeslots. If required this system could be extended to include several filters (even up to one filter for each timeslot!) allowing more than one timeslot for the filter output to decay.

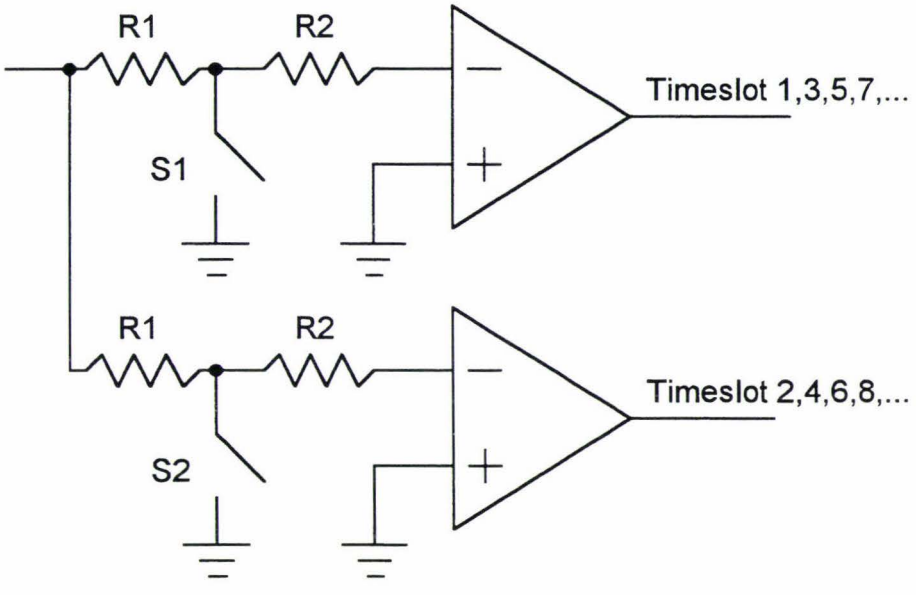


Figure 5.15 Multiple Filter Switching.

In Figure 5.15 Switch S1 would be closed during the even timeslots, and S2 would be closed during the odd timeslots. If the timeslot contains a transmission the output would always present a rising envelope (of varying amplitude), regardless of the contents of the neighbouring timeslots.

There is an obvious cost disadvantage with this method, as now two identical filters are required. Accurate switching is also required at the inputs to the filters.

5.2.4.2 Adjust Filter Clock Speed

It is possible to increase the clock speed by up to 50 times (hardware specific (National Semiconductor (1986)). This would move the passband of the filter upto the 13kHz range while keeping the Q's of the filter stages constant. This result is a 50 times increase in bandwidth and thus as shown in Figure 5.9 the decay time of the filter would be greatly reduced. The signal stored in the filter at the point of changing

the clock speed would then behave as a 13kHz signal and decay quickly. The filter would still pass any inputs in the 13kHz range. This could be avoided by blocking the inputs while the clock is sped up.

Thus a small portion of the end (or beginning) of the timeslot would have to be dedicated to speeding up the filter clock and clearing out the filter.

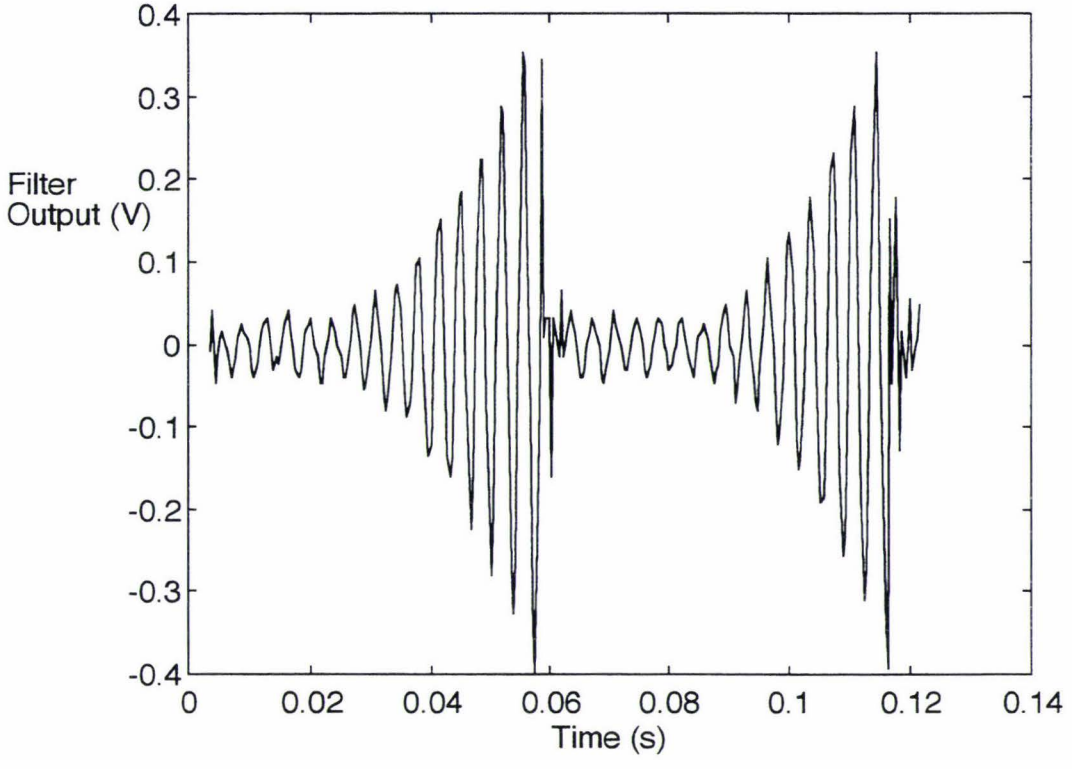


Figure 5.16 Switched Clock Filter Output.

Figure 5.16 shows the response of this method to two slave transmissions in row. The output continues to rise until the end of the timeslot. At the end of the timeslot the clock is sped up (and inputs earthed) and the output decays very quickly ready for the next timeslot. Notice the delay through the filter (approx 20ms), which increases the filter response time.

Unfortunately if there is a constant delay through the filter this is not taken into account, and means effectively that the bandwidth of the filter must be wider to get the same response time as if there were no internal delay through the filter. If each of the filter stages were reset individually (requiring 4 separate filter clock signals from the microprocessor) then a progressive resetting could be implemented whereby the delay of each filter could be taken into account. This would remove the effect of the filter delays from the system, yet the added complexity is currently not required.

This method has the disadvantage, that the timeslots are being reduced even further in duration and that switching is again required. However, a faster clock can easily be made available from the microprocessor, and no additional filter is required.

5.2.4.3 Change Filter Parameters

If the bandwidth of the filter was increased the rise and decay times would decrease. This method aims to dynamically change the parameters of the filter at the end (or beginning) of the timeslot to reset the filter. To do this the Q can be reduced by shorting one of the feedback resistors in the filter.

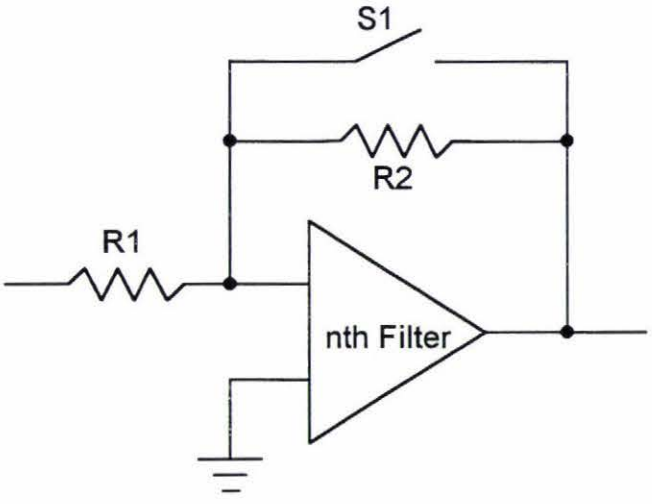


Figure 5.17 Reducing Filter Q.

In Figure 5.17 S1 should be closed when the filter is to be reset. Under experimentation this method did not work well. It appeared that noise associated with opening the switch S1 caused the filter to self oscillate at the filter center frequency.

5.2.5 Noise Pickup

Noise pickup stimulated the high Q filter to self oscillate. Under testing on a breadboard, the filter give a significant output when no input was present. When a small signal was placed at the input, a beating was seen at the filter output indicating the self oscillation frequency and the signal were slightly different frequencies.

The amplitude of the noise output was seen to be proportional to the amplitude of the clock signal.

Two solutions to reducing the noise pickup were implemented in the maser unit. The first was reducing the clock signal amplitude and the second was redistributing the gains of the filter stages.

5.2.5.1 Reducing Clock Amplitudes

The microprocessor output voltage amplitude varies from the power supply voltage to ground (CMOS 5.0V, 0V), yet the filter chip requires that the voltage only range from 0.8V to 2.0V (TTL).

The clock amplitude was reduced to the minimum required using an open drain I/O pin on the microprocessor to drive the clock and an external voltage divider network shown in Figure 5.18.

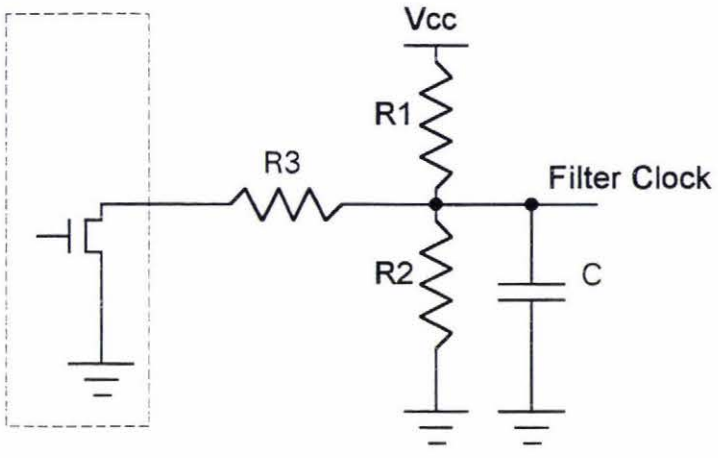


Figure 5.18 Reducing Filter Clock Amplitude.

Some filtering on the filter clock will also lower the high frequency components of the clock waveform. Provision was left in the circuit board for the addition of a capacitor for this purpose. In the breadboard circuit, the stray capacitance performed this function.

5.2.5.2 Adjusting Filter Gains

The noise pickup enters the system at the input of any of the four filter stages, as shown in Figure 5.19. Thus any noise entering the system at the first filter will propagate through all four filters.

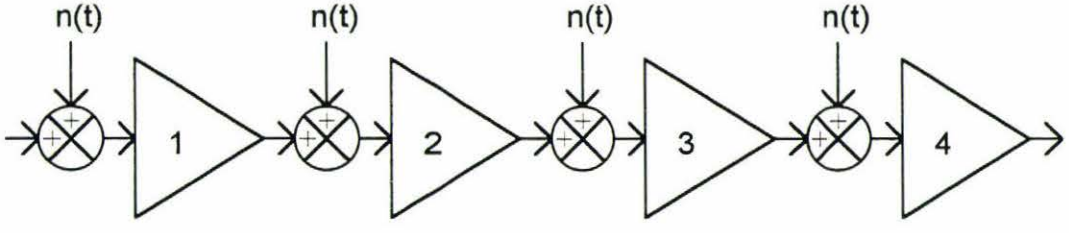


Figure 5.19 Filter Noise Pickup.

Noise entering the system at the final stages is not effected by the gains at the preceding stages of the filter. Thus in creating a filter where the minimum gains are at the final stages reduces the noise pickup as much as possible.

The filter gains must be high enough so that clipping at one stage causes clipping at the output of the filter. Thus a minimum gain of 1 for each filter stage at the slave unit transmission frequency is required. (Note this is not the gain at the center frequency of the individual filters).

5.2.6 Final Circuit

The filters can be programmed with three resistors. There are several modes of operation (methods of arranging the filter components) each with slightly different operating characteristics (National Semiconductor (1986)). In this case the mode with the least sensitivity to resistor values (and thus the most accurate was chosen). The schematic for each of the four individual filters is shown in Figure 5.20.

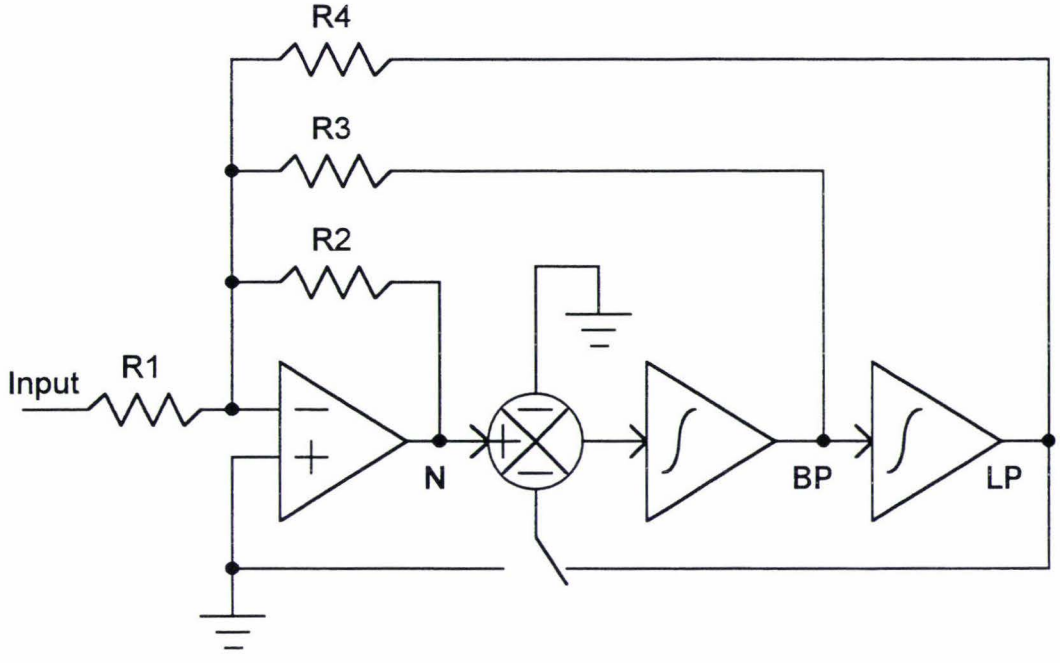


Figure 5.20 Filter Stage Schematic.

The equations that accompany the schematic are shown in equations 5.2 to 5.6.

$$f_0 = \frac{f_{CLK}}{50}\sqrt{\frac{R2}{R4}+1} = \text{Center Frequency} \tag{5.2}$$

$$Q = \frac{\sqrt{\frac{R2}{R4}+1}}{\frac{R2}{R3}} = \text{Quality Factor of Complex Pole Pair} \tag{5.3}$$

$$H_{OLP} = -\frac{R2/R1}{R2/R4+1} = H_{ON} \quad (\text{as } f \rightarrow 0) \tag{5.4}$$

$$H_{OBP} = -\frac{R3}{R1} \tag{5.5}$$

$$H_{ON2} = -\frac{R2}{R1} \quad (\text{as } f \rightarrow f_{CLK}/2) \tag{5.6}$$

The filter clock was sped up to 500kHz and the inputs were blocked at the start of each timeslot in order to reset the filter. Thus the filter output envelope rises up the

amplitude of the transmission pulse during the timeslot, regardless of the neighbouring timeslot content. The amplitude to which the filter rises will depend upon the amplitude of the slave transmission (as seen at the master unit).

5.2.7 Post Filter Signal Conditioning

This section outlines the circuitry that is required between the filter output and the microprocessor.

The less information that is lost in this process, the more information that is available for the microprocessor to use in determining if a transmission is in any given timeslot.

The signal rectifier is presented along with options for interfacing the filter to the microprocessor. A peak detector, an integrator and a simple A/D conversion, are presented and the benefits of each method are discussed. Common to each method is an A/D converter.

5.2.7.1 Signal Rectifier

To simplify the signal processing a low voltage rectifier is required to convert the bipolar filter output signal to a bipolar signal.

A conventional operational amplifier precision full wave rectifier was used. The rectifier circuit is shown in Figure 5.21. A series capacitor is included to remove any DC offset from the output of the filter.

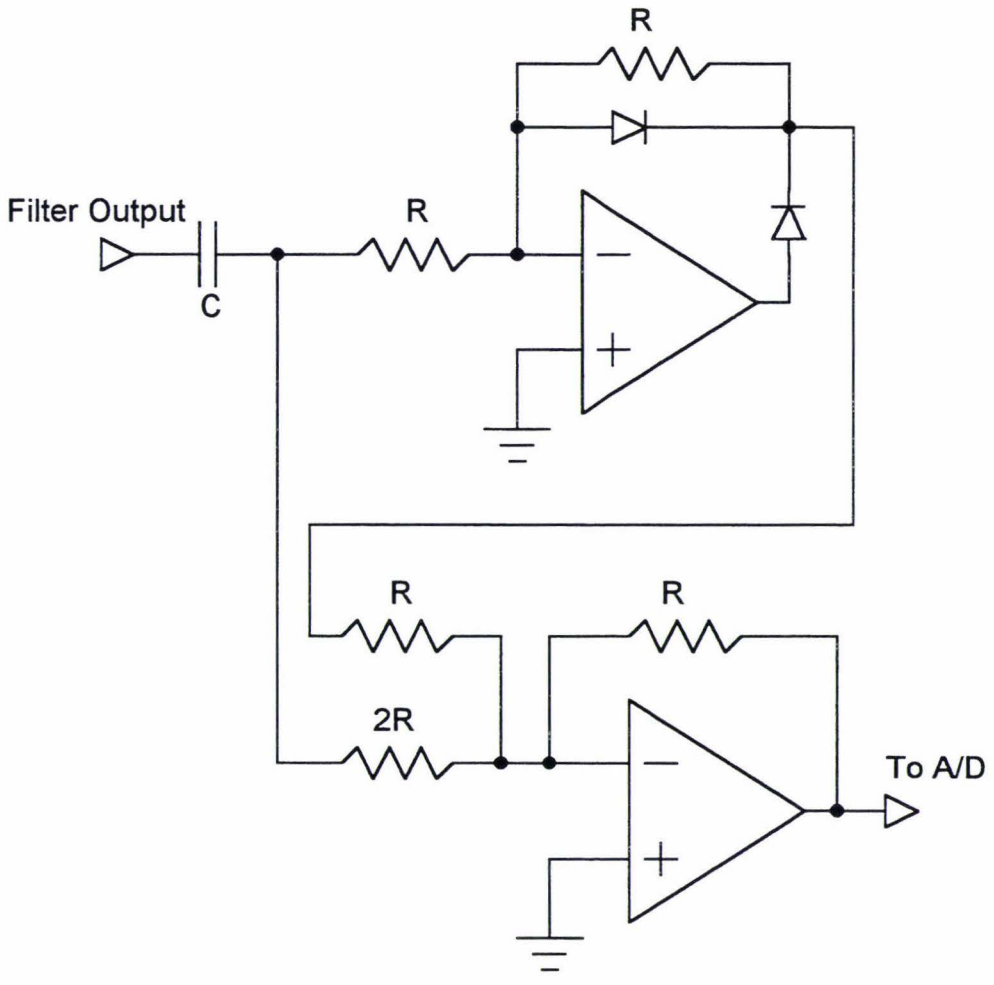


Figure 5.21 Signal Rectifier

5.2.7.2 Integrator

An integrator can still be used to show information about the envelope of the input if required, and it has the advantage that the A/D converter does not have to be fast enough to read in the result, in that it will not be required to pick out the peaks of the 273Hz signal.

In this method the second Op-Amp in the rectifier circuit can also be used to create an integrator circuit to simplify the operations of the microprocessor.

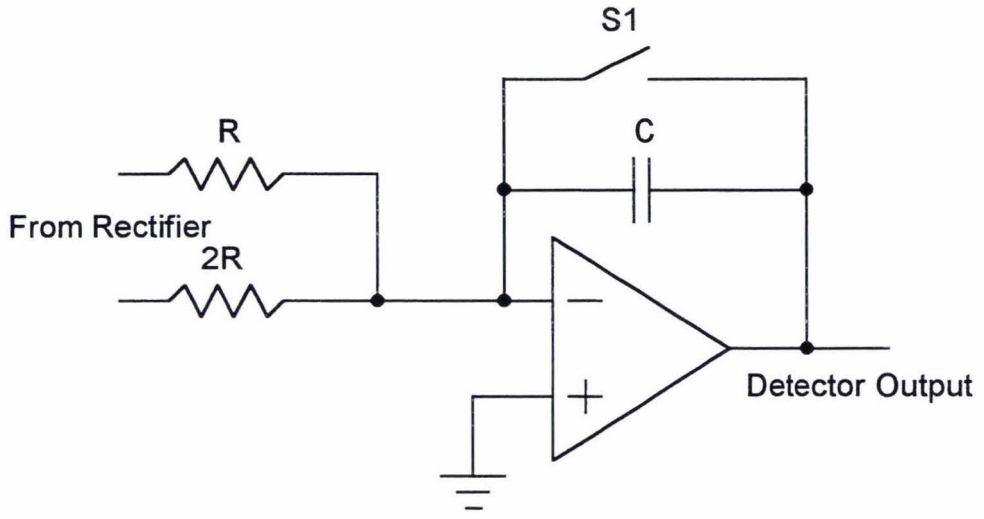


Figure 5.22 Integrator

The integrator has the disadvantage that if there is noise throughout the whole timeslot, and signal only after the internal delay of the filter, then the noise will have an increased influence on the filter. Figure 5.23 shows example output from the rectifier for filter noise and transmission. Figure 5.24 shows the integrator output for these input signals.

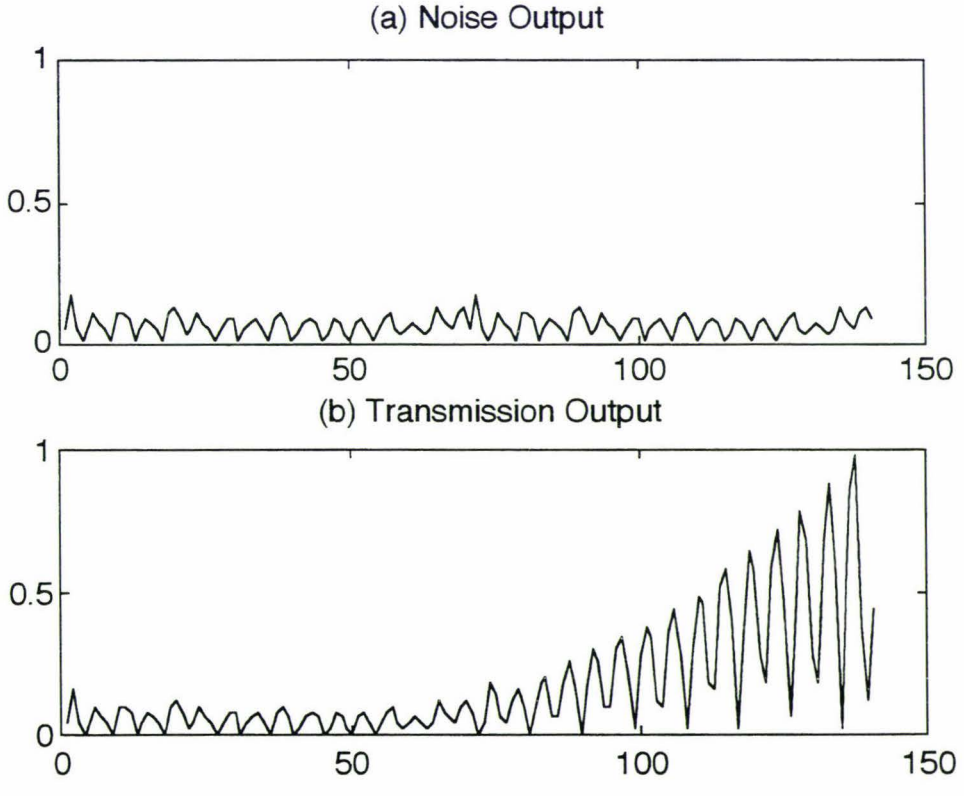


Figure 5.23 Rectifier Output For Noise (a) and Transmission (b).

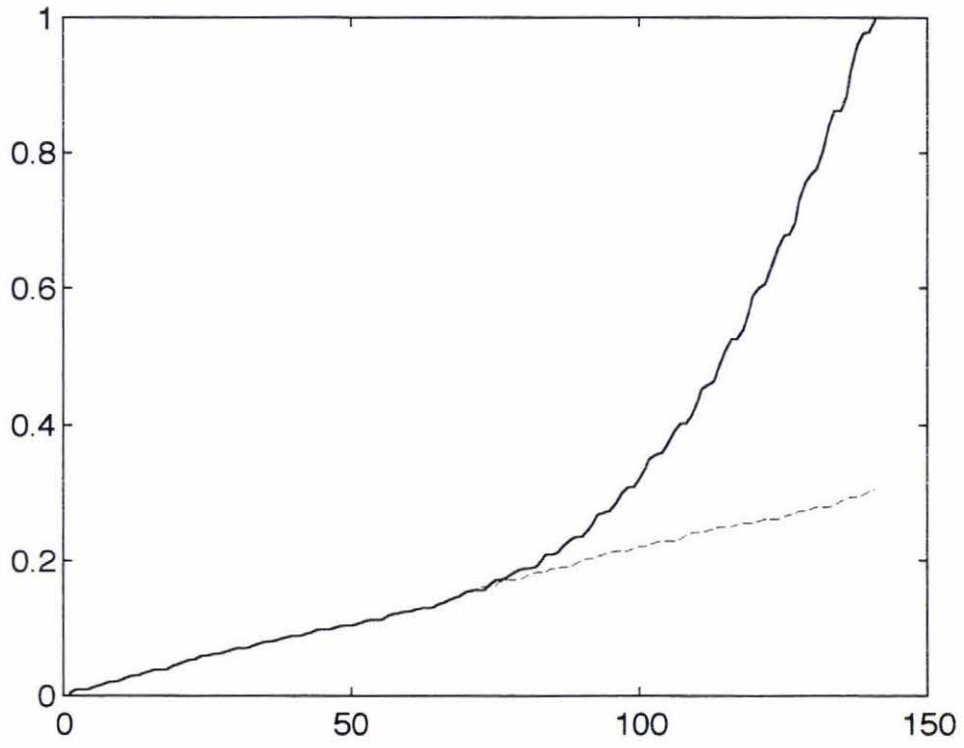


Figure 5.24 Integrator Output For Noise (Dashed) and Transmission (Solid).

Figure 5.24 shows that in the example the noise output when there is no transmission is approximately 30% of the output when there is a transmission present.

5.2.7.3 Peak Detector

The problem with the integrator circuit is that any noise pickup will have an increased influence on the microprocessor decision as to the presence of a transmission.

The knowledge that the transmission signal causes the output of the filter to continue to rise until the end of the timeslot can be used. Any output caused by the noise pickup will reach a peak value before the end of the timeslot, and should be lower in amplitude than the final transmission peak amplitude.

Figure 5.25 shows a sample peak detector circuit output. The solid line is the peak detector input and the dashed line the peak detector output. Note that the noise at the start of the timeslot has a far reduced influence on the peak final peak detector output, and any output for a timeslot in which there is no transmission, will be reduced in amplitude when compared to the integrator output.

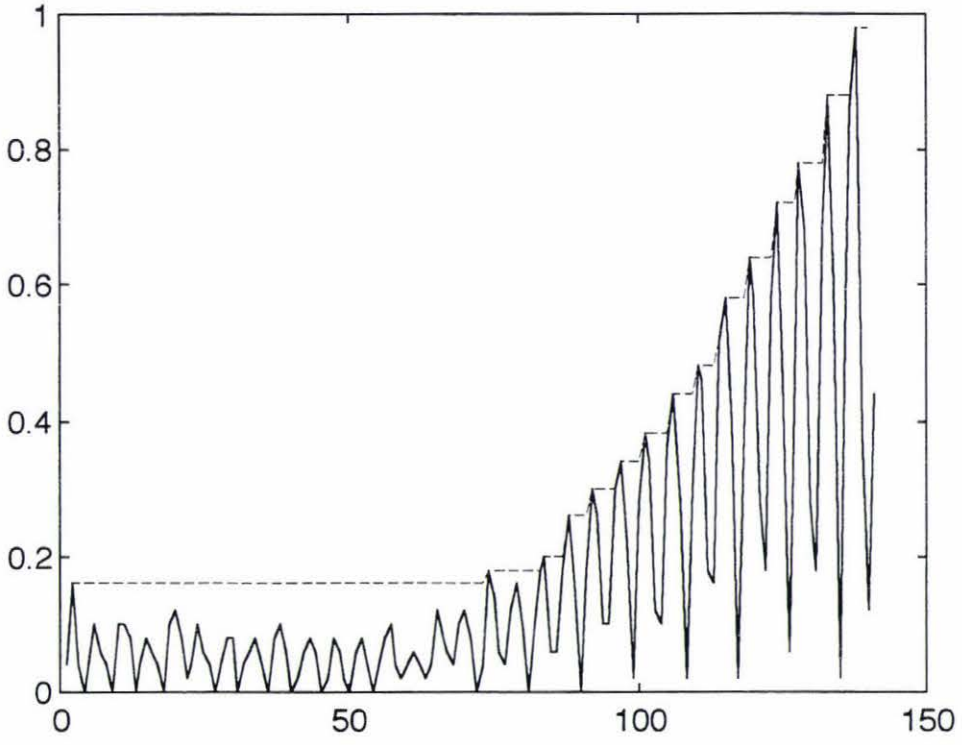


Figure 5.25 Peak Detector Output.

The peak detector circuit is shown in Figure 5.26.

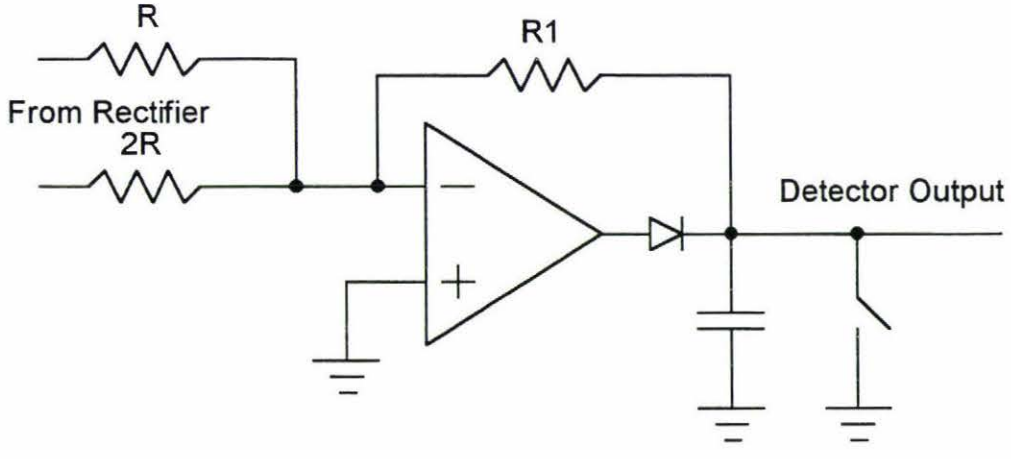


Figure 5.26 Peak Detector.

The peak detector still retains the advantage of not requiring the A/D to pick out the peaks of the individual 273Hz signals, rather the A/D only needs to read the output at the end of the timeslot. It is more difficult with the peak detector to recover envelope information, but with a continually rising signal this is still possible.

5.2.7.4 A/D Converter

Both the peak detector and the integrator have the possibility of suffering from an abnormal output of the filter. For example a large noise spike at any point in the filter

output would increase the output both the integrator and the peak detector. If the A/D was fast enough and the microprocessor could process the information fast enough then the microprocessor could look for the characteristic of the transmission in the timeslot.

A TL549 A/D used with a microprocessor running at 12MHz would be able to read an A/D output during one half of the filter clock cycle and allow the A/D to measure the input during the next half of the A/D filter clock cycle. With the filter clock running at 12.8kHz this allows 23 evenly spaced samples of each of the 546Hz signal peaks out of the rectifier. This should easily be enough to pull out a value very close to the signal peak. Figure 5.27 shows the worst case scenario for the A/D sampling. Each cross represents a sample from the A/D. The maximum sampled value is 99.8% of the sine wave amplitude.

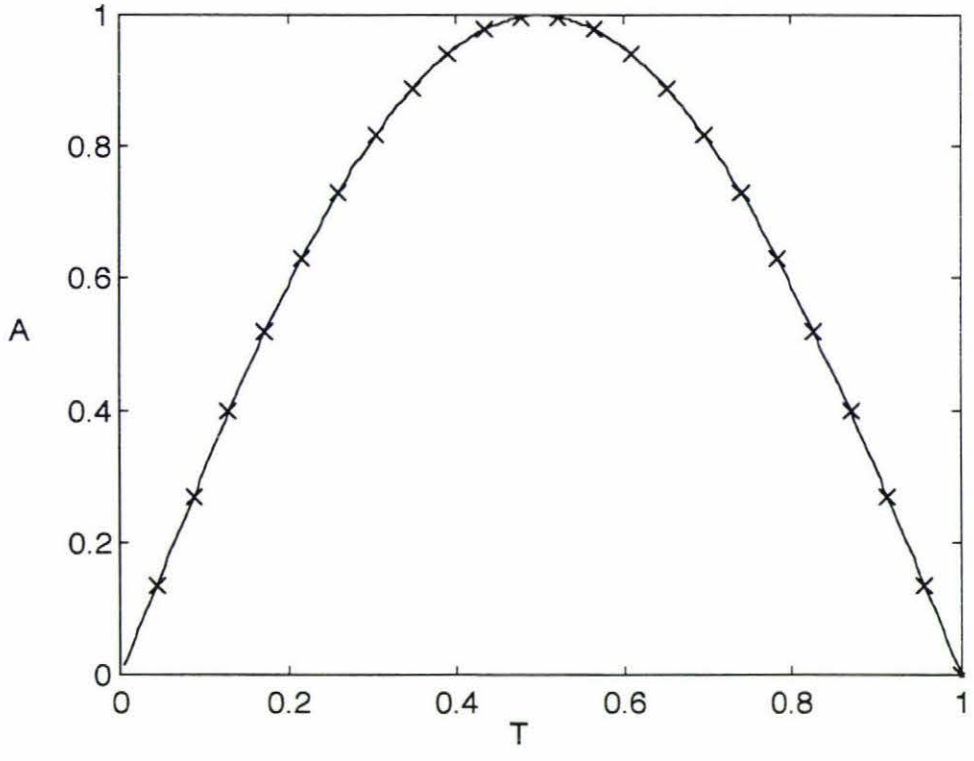


Figure 5.27 A/D Sample Rate

By using the microprocessor rather than hardware to decide if a pulse is present ensures that none of the information output of the filter is lost and in the future if necessary the algorithm may be updated to use more of the information available. Section 5.5.1.4 details the software algorithm that is currently used to detect the presence of a slave unit transmission.

5.2.8 Filter Noise.

In order to combat the noise output from the filter which is of constant amplitude, in timeslot zero the inputs are blocked and the filter is let run. The peak output from the filter is then subtracted from the future results. This is done in software as the A/D is

used to look at the envelope. Thus if no signal is present a result of zero should be obtained in a slave timeslot.

5.3 Filter AGC

The master unit will have to work in many varying fence conditions. The energiser impedance and the wire impedance are just two variables that would vary between systems and have an effect on the slave transmission voltages.

As it is impossible to predict under what situations the system will have to operate and at what level interference will be, it is necessary for the filter to be able to detect low amplitude slave responses in situations of low noise, and to reject large noise amplitudes in high amplitude slave transmission situations. This is accomplished with the use of a gain control circuit, where if the transmissions are causing the filter to be overdriven, then the input impedance can be changed at the filter input to lower the filter output and thus reduce the filter noise output.

This section outlines the requirements of an automatic gain control (AGC) circuit for the filter, and details the theory and circuit behind the final circuit used in the master unit. Protection for the Filter and AGC circuit is also discussed.

5.3.1 AGC Requirements

Table 5.1 shows the energiser impedances are rarely over 2kΩ. In order to ensure the filter output will not clip the AGC circuit is designed for a 4kΩ energiser to function without clipping. Figure 5.8 shows that for a 4kΩ energiser the expected slave transmission amplitude would be 4.9V peak to peak. This defines the maximum input to the AGC that should not allow clipping in the minimum gain state. The filter requirements (section 5.2.1) specify the maximum gain state. The AGC should make a signal of as little as 50mV peak to peak detectable.

5.3.2 AGC Circuit

AGC is implemented as a programmable voltage divider at the filter input. The simplified filter circuit is shown in Figure 5.28.

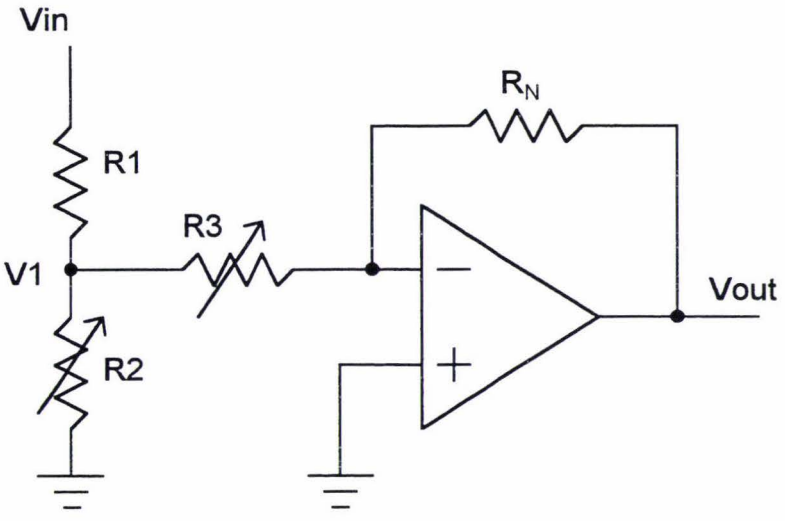


Figure 5.28 Signal Rectifier

Where R_N is the equivalent gain resistance of the total filter. The gain equations for the circuit are shown in equations 5.6 to 5.9.

$$V_{out} = \frac{R_N}{R3} V1 \tag{5.6}$$

$$\text{where } V1 = \frac{R2 \| R3}{R1 + R2 \| R3} V_{in} \tag{5.7}$$

$$V_{out} = \frac{R_N}{R3} \times \frac{R2 \| R3}{R1 + R2 \| R3} V_{in} \tag{5.8}$$

this simplifies to:

$$G = \frac{V_{out}}{V_{in}} = \frac{R_N R2}{R1R2 + R2R3 + R1R3} \tag{5.9}$$

Equation 5.9 shows that by increasing resistance R3 the gain of the filter is dropped as would be expected. The gain of the filter can also be changed by changing resistance R2. If R1>>R2 and R3>R2 then the dominant term in the denominator is R1R3. Thus if R2 is increased, the gain of the filter will increase.

Adjusting R2 is preferable to adjusting R3 as the circuit is then switching to ground and not switching a series resistance. This should reduce noise pickup at the front of the filter. R2 can be adjusted by the microprocessor.

Under normal filter operating conditions the voltage at V1 should not rise above 5V or below -5V. Note that the filter does not have to work during a fence pulse. This can be used to give values for R3 and R1 given that R1 is set to 100kΩ due to the availability of high voltage resistor values, and the minimum gain required for the filter. The maximum out of pass band input voltage for which the filter is expected to work is 100V peak to peak, thus ±50V with respect to ground.

$$\frac{R3}{R3+R1} = \frac{5}{50} = \frac{R3}{R3+100} \qquad (5.10)$$
$$\therefore R3 = 11k\Omega$$

The filter output will start to clip at an output of approximately 8V peak to peak. The AGC requirements specify a maximum input signal of 4.9V peak to peak signal causing the filter output to clip. This sets the minimum gain of the filter AGC at 1.63.

The maximum gain of the filter can be determined from the smallest signal that should cause the output to clip. The minimum detectable input voltage level is 50mV peak to peak. If this is to cause a half of full scale output then the filter output should clip at 100mV peak to peak. This corresponds to a gain of 80. Thus the filter may be able to measure lower (depending upon the noise level), yet it reaches the 50mVpp design specifications easily.

With a 8-way multiplexer there are 8 possible AGC levels to be evenly distributed throughout the gain range of 1.63 to 80. The gain ratio between adjacent stages (n) can be found as shown in equation 5.11.

$$80 = 1.63 \times n^7 \qquad (5.11)$$
$$n \approx 1.744$$

The maximum gain available for the first filter stage is given by equation 5.12. This gain of 3.9 corresponds to an overall filter gain of 78 which is adequate.

$$H1 = -\frac{R_N}{R1+R3} \qquad (5.12)$$
$$= -\frac{430}{100+11}$$
$$= -3.9$$

The resistance R2 for each of the stages can be found by using equation 5.13, where H1 is the gain of the first filter.

$$R2 = \frac{1100}{\frac{430}{H1} - 111} \qquad (5.13)$$

The gain of the filter is defined by equation 5.14, where Hn is the gain of the nth stage of the filter. It can be seen from this equation that K is the gain required from the combined filter stages if the overall gain of the filter is to be 1. K is fixed by the filter passband characteristics. Tables are used to set this value at 302 (Williams (1990)).

$$G = \frac{H1 \times H2 \times H3 \times H4}{K} \qquad (5.14)$$

Table 5.3 AGC Resistances.

AGC Level	Gain	K	H_{Total}	H2xH3x H4	H1	R_{AGC}
1	80	302	24160	6040	4	∞
2	45.87	302	13853	6040	2.294	14382
3	26.30	302	7943	6040	1.315	5093
4	15.08	302	4554	6040	0.754	2395
5	8.65	302	2611	6040	0.432	1245
6	4.96	302	1497	6040	0.248	678
7	2.84	302	858	6040	0.142	377
8	1.63	302	492	6040	0.082	213

Note that H1 of 4 is unattainable but a gain of 3.9 is achieved with no resistance R2. The low resistance of 213Ω is of concern due to the resistance of the multiplexer used to switch the resistances which adds a significant error when the AGC selects low values of R2. The multiplexer configuration is shown in Figure 5.29.

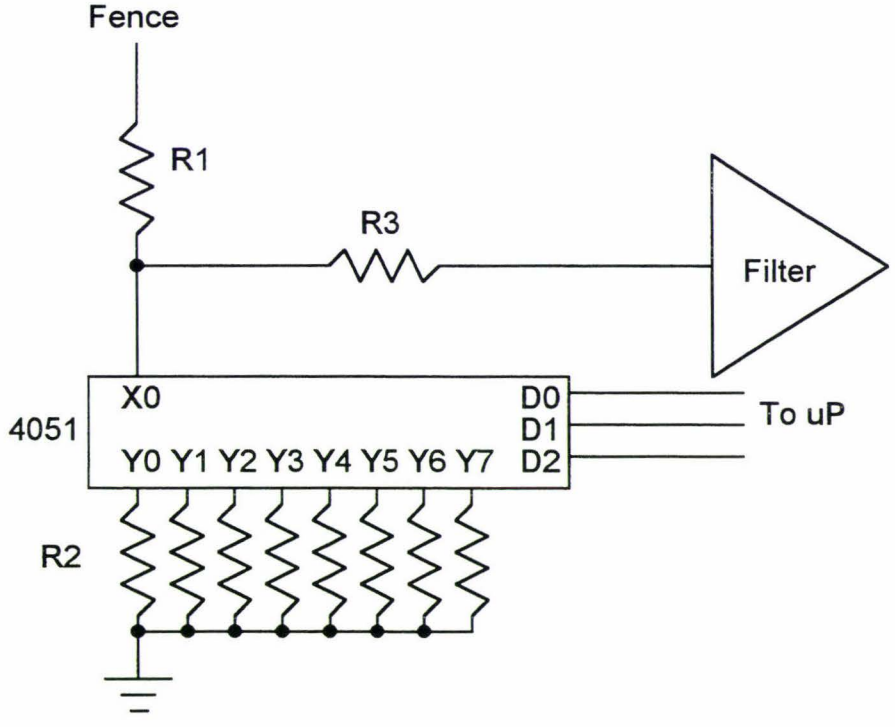


Figure 5.29 AGC Multiplexer

The typical resistance of the multiplexer switch (HC4051) when Vcc-Vee=9.0V is 60Ω. The AGC resistances should all have 60Ω subtracted to compensate for this resistance.

5.3.3 AGC Circuit Protection

High voltage resistors will be required at the filter input to withstand the energiser pulse continuously. Clipping is also required at the input, such that the pulse does no damage to the filter circuit.

The input to the multiplexer is not permitted to rise above the multiplexer supply voltage. Resistance R3 has already been sized, such that under normal operating conditions (when the filter is expected to run) the voltage will not rise above 5V. The fence pulse, however can be up to 10kV peak causing a voltage of 1kV at the multiplexer. By placing 5V zener diodes across the multiplexer input to ground provides protection without effecting the normal operation.

When the master unit is powered down an energiser pulse could cause damage to the multiplexer as the pin input is rising above the supply voltage. Schottky diodes are added to prevent the input from rising more than 0.3V above the power supply voltage.

The complete AGC circuit is shown in Figure 5.30.

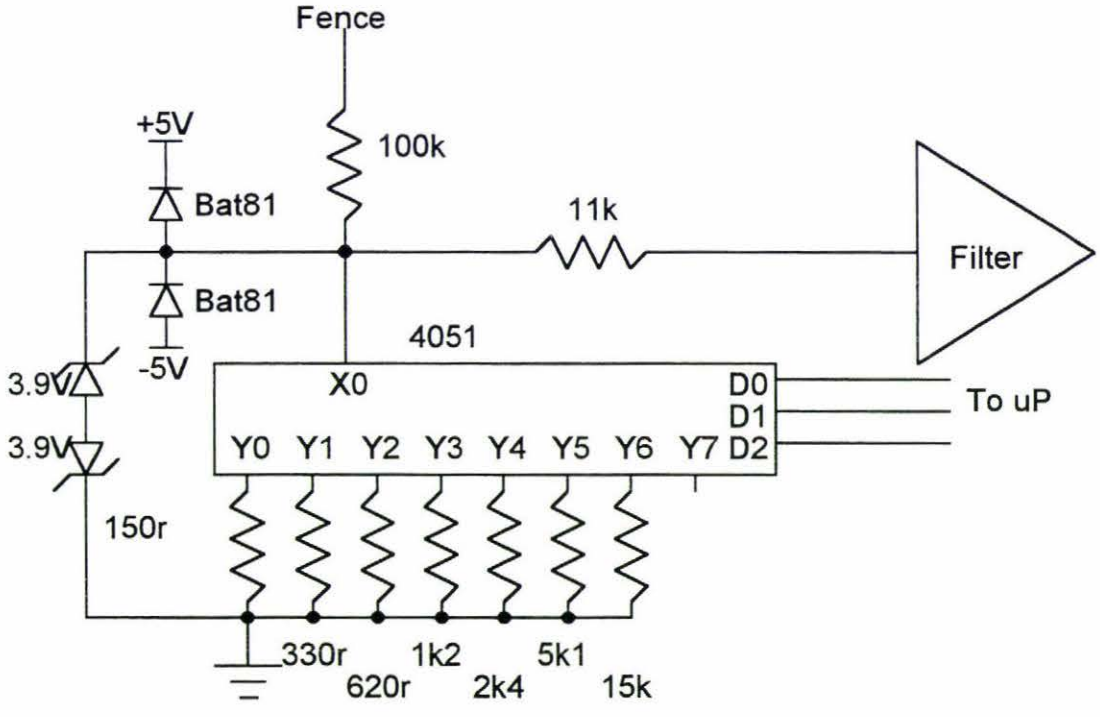


Figure 5.30 Complete AGC Circuit

5.4 Master Unit User Interface

The user interface consists of two major components. The first is the display. This is used to advise of the slave voltages and alarm status to the user. The slave voltage is the decoded voltage at the slave unit. The alarm status indicates if any of the decoded voltages at the slave unit are lower than their desired operating voltage. The display should reveal all the required information simply and quickly. The second component is the keypad. This is used for the user to set the alarm thresholds. The alarm thresholds are the desired operating voltages and one threshold is available for each slave unit. The keypad should be self explanatory and not necessarily require the user to read instructions to discover how to use it. Should the user have difficulty programming the unit he/she may neglect to use it to its full potential and then underestimate the system's value.

5.4.1 The Master Unit Display

The choice of display is a balance between cost and functionality. The larger the display the more information or the easier the same information can be displayed and the more expensive the display is. A small display is inexpensive but provides few options for display format.

The display chosen was a 4x20 character display. This allowed a few options for the format of the information. These are presented below and their merits discussed.

5.4.1.1 Information To Display

The master unit must display information for every slave that is registered (currently transmitting). For each of these slaves it must display the decoded voltage seen at the slave unit, and the alarm threshold for that slave unit if one has been set by the user. The display must allow a cursor when the user is setting a threshold for a slave unit, such that the user can identify which slave threshold is being set.

5.4.1.2 Grid Style Display

Figure 5.31 shows an example of this display format. The numbers in the box are the display output, while the text outside the box is written on the master unit case.

Slave Peak Voltage (kV)

Slave #				
1....4	5.7	6.3	4.2	6.4
5....8	8.4	6.2	---	7.2
9..12	---	---	2.1	5.3
13..16	3.4	---	5.1	8.1

Figure 5.31 Grid Style Display.

The '---' will indicate that the slave is not registered (no transmissions received). Slave unit threshold values will not be displayed until the user goes into an edit mode, whereupon all the slaves will display the threshold values instead of the voltage measurements.

This method displays all of the slave voltages at once, and thus the user can receive the information at a glance. The user will get familiar with the normal operating pattern of the display and be able to pick faults in the system quickly. This method requires writing on the case to line up with the display, and has the disadvantage that the slave voltages are not shown when editing the slave threshold values. The display is also under utilised if there are only a few slaves in the system.

5.4.1.3 Row Display

This method attempts to display both slave voltages and slave thresholds at the same time, by reducing the number of significant digits in the voltages and thresholds. Only two rows of data are required (and thus only a two line display). An example of this method is shown in Figure 5.32.

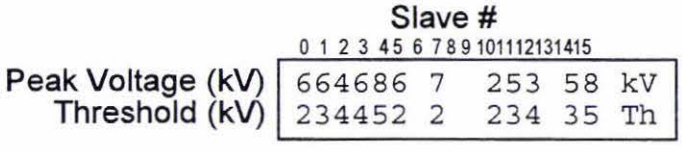


Figure 5.32 Row Display

The slave units are displayed with increasing ID's across the row. For example above, the peak voltage for slave ten is 2kV and the threshold is 2kV. This unit has the obvious advantage that all of the information is readily available on screen and that a smaller cheaper screen can be used. However the display is cluttered and difficult to read. It also has the disadvantage that the data has been reduced. Again the display will be under utilised if only a few slaves are present in the system.

5.4.1.4 Rolling Screen

This method splits the slave data over several screens. One row is dedicated to a slave, and thus there can be information from only four slaves on screen at any time. The display will continually scroll through the slaves if more than four slaves are registered in the system.

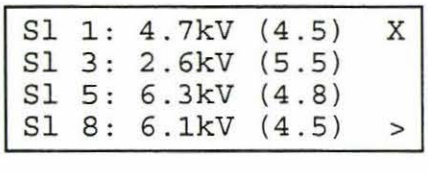


Figure 5.33 Rolling Screen

In Figure 5.33 the data for slaves 1,3,5 and 8 are shown. Slaves 2,4,6 and 7 are not registered. The values in brackets are the threshold values for the slave units. An 'X' in the top right corner indicates that there is at least one slave in the system (possibly on other pages) that is in alarm. The '>' indicates that there are more slaves in the system that are not shown on this page.

This method has the disadvantage that the user must now wait for the display to scroll or use the Page Down button to see data for all of the slaves.

This method was chosen for implementation in the master unit, as it allowed all data relevant to one slave to be displayed at once, and in the most likely usage scenario with only a few slaves in the system registered, the display will not be underutilised.

5.4.2 The Master Unit Keypad

The keypad is the link that allows the user to enter in slave unit thresholds, deregister slaves, and quickly view the slave voltages. This must be as simple as possible to use. Few buttons allow the user to find the button required quickly, however too few buttons with multiple tasks on each, will easily confuse the user. With an increase in the number of buttons, comes an increase in cost and complexity.

The buttons required are linked to the functions that the master unit must allow the user to perform, thus a description of the functions required is necessary.

5.4.2.1 Functions To Perform

The keypad's main task is to allow the user to set the thresholds (alarm setpoints) for each of the slave units. To do this the user will need to be able to select a slave unit. This will require some form of cursor keys. They will then have to advise the interface that they wish to change the threshold of the selected unit. They then must be able to set the threshold using the keypad.

Registration of slave units will occur automatically when a transmission is received from the slave unit. Deregistration must be selected by the user. This could be done as a special case of the slave threshold.

The user must also have some method of stopping the alarm from sounding once a slave voltage drops below its alarm threshold, without waiting for the voltage to return on the fence line. This is in effect an alarm acknowledgment.

5.4.2.2 Keypad Layout

The master unit keypad was kept as simple as possible. The final keypad is as shown in Figure 5.34.

Figure 5.34 Keypad Layout

The 'Acknowledge Alarm' button is used to stop the buzzer from sounding. All slaves in alarm status will remain in alarm, yet the alarm will stop from sounding until, the slave gives a non-alarm voltage and then falls below the threshold again. Should another slave go into alarm after the Acknowledge Alarm button has been pressed then the alarm will again sound.

The 'Set Threshold' button is used to go into set threshold mode. This should only be pressed when the cursor is on the slave to be adjusted. This will then change the function of the remaining two buttons into threshold adjustment rather than Page Down and Move Down buttons. When the user has finished setting the threshold this button is again pressed to return the master unit to normal and save the new threshold.

The 'Page Down / 1000's' button has a dual function. In normal mode (Page Down) this is used to show the cursor and to scroll the display to the next page. When moving past the last page of slaves, the display will return to the first page. In set threshold mode (after 'Set Threshold" has been pressed) the button is used to set the kV figure in the currently selected slave's threshold. Pushing this button will increment the threshold by 1kV. When the threshold rises above 10kV, the user will have the option of deregistering the slave with a '---' appearing for the threshold value. If the user continues to press the 1000's button the kV threshold will return to 0 and continue to rise as the button is pressed.

The 'Move Down / 100's' button is another dual function button. In normal mode (Move Down) the button is used to select the next slave on the page. If the user moves past the last slave, the cursor returns to the top of the current page. In set threshold mode (100's) the button is used to set the 100's value in the selected slave units threshold. Advancing the 100's value past 9 will return it to 0.

5.5 Microprocessors

Two microprocessors are required for the master unit. This is a result of the isolation requirement between the various parts of the master (NZS 3350.1:1994). Section 5.6.1 details the isolation requirements further.

The first microprocessor is used to control the filter (filter microprocessor), and to decide if a transmission is present. It also signals to the second microprocessor (the user interface microprocessor) the results of the transmissions after every pulse. An ATMEL 89C2051 microprocessor was used for this task.

The user interface microprocessor must decode the voltage at each slave, control the display and the keypad and alarms, and receive data from the filter microprocessor. An ATMEL 89C52 was used here.

This section outlines the microprocessor circuits and the software used.

5.5.1 Filter Microprocessor

The filter microprocessor must first detect when an energiser pulse has arrived. Then it controls the timeslots using the energiser pulse to synchronise with the slave units. The microprocessor must also control the filter clock. This means it must speed up the clock at the end of the timeslot to reset the filter (see section 5.2.4.2). The microprocessor is also responsible for detecting if there is a slave transmission in any timeslot. The microprocessor is then responsible for transmitting to the user interface microprocessor the slave units transmission data.

5.5.1.1 Energiser Pulse Detection

The filter output cannot be used to indicate if the energiser pulse has arrived as the pulse will not propagate through the filter. Measurement at the input to the filter is not possible either due to the clipping and AGC control. A separate connection to the fence is used. A dual comparator (LM319) is used to detect either positive or negative energiser pulses. The two open collector comparator outputs are wired ORed with a shunt capacitor to give a fast detect, slow return to normal characteristic. The circuit is shown in Figure 5.35.

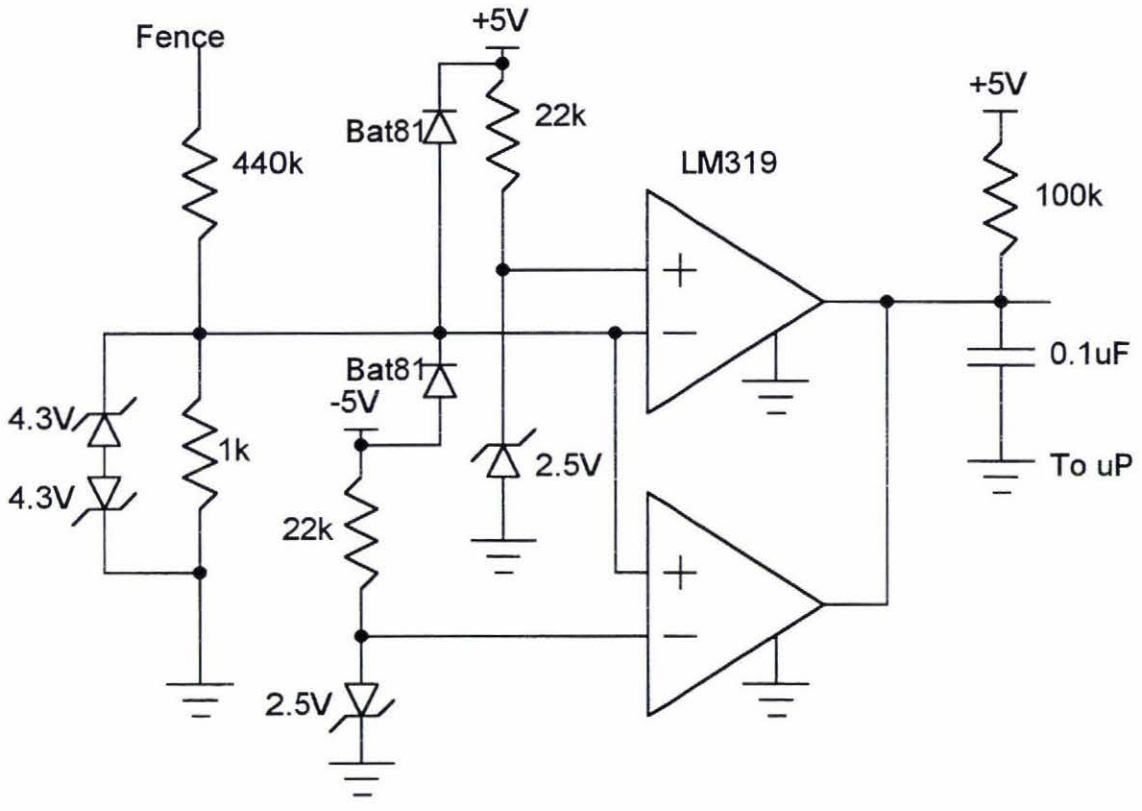


Figure 5.35 Energiser Pulse Detector.

The slow return to normal prevents multiple triggering that might occur with some energisers (especially RFI suppression energisers) have pulses that 'ring'.

The input resistor divider is sized such that voltages of ±1kV on the fence are detected as energiser pulses. Voltages greater than ±2kV would give a voltage greater than ±5V at the comparator input, thus zener diodes are added to the input ensure that the comparator is protected. Schottky diodes (Bat81) are also added, such that when the power supply is off, an energiser signal will not damage the IC.

5.5.1.2 Opto-Transmitter

The microprocessor must remain isolated from the User Interface microprocessor. The transfer of the slave transmission results to the User Interface microprocessor is accomplished optically. The transfer occurs in timeslot 0 at a speed of approximately 10kbps.

The opto-transmitter is driven directly from the microprocessor I/O pin. The circuit is shown in Figure 5.36.

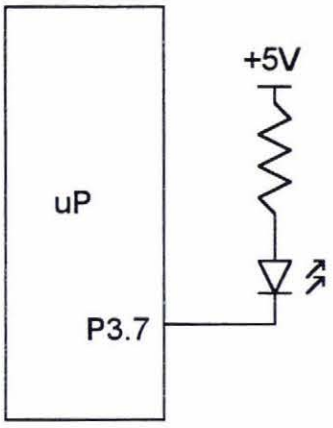


Figure 5.36 Opto-Transmitter.

5.5.1.3 Reset/Watchdog Timer

A reset IC is included in the filter microprocessor circuit to ensure that the microprocessor is put into reset when the power falls below the microprocessor operating voltage. A MAX1232 was selected for this task. A watchdog timer is also included in the IC such that should there be a code crash in the microprocessor, and the filter clock is no longer toggled, the reset IC will force a reset of the microprocessor. The circuit is shown in Figure 5.37.

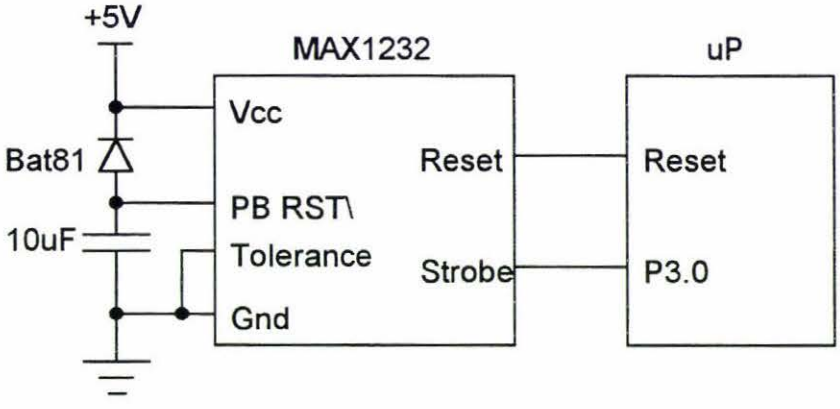


Figure 5.37 Filter Microprocessor Reset Circuit

5.5.1.4 Software

The structure of the filter microprocessor software is shown in Figure 5.38. The code for the filter microprocessor software is given in appendix 7.

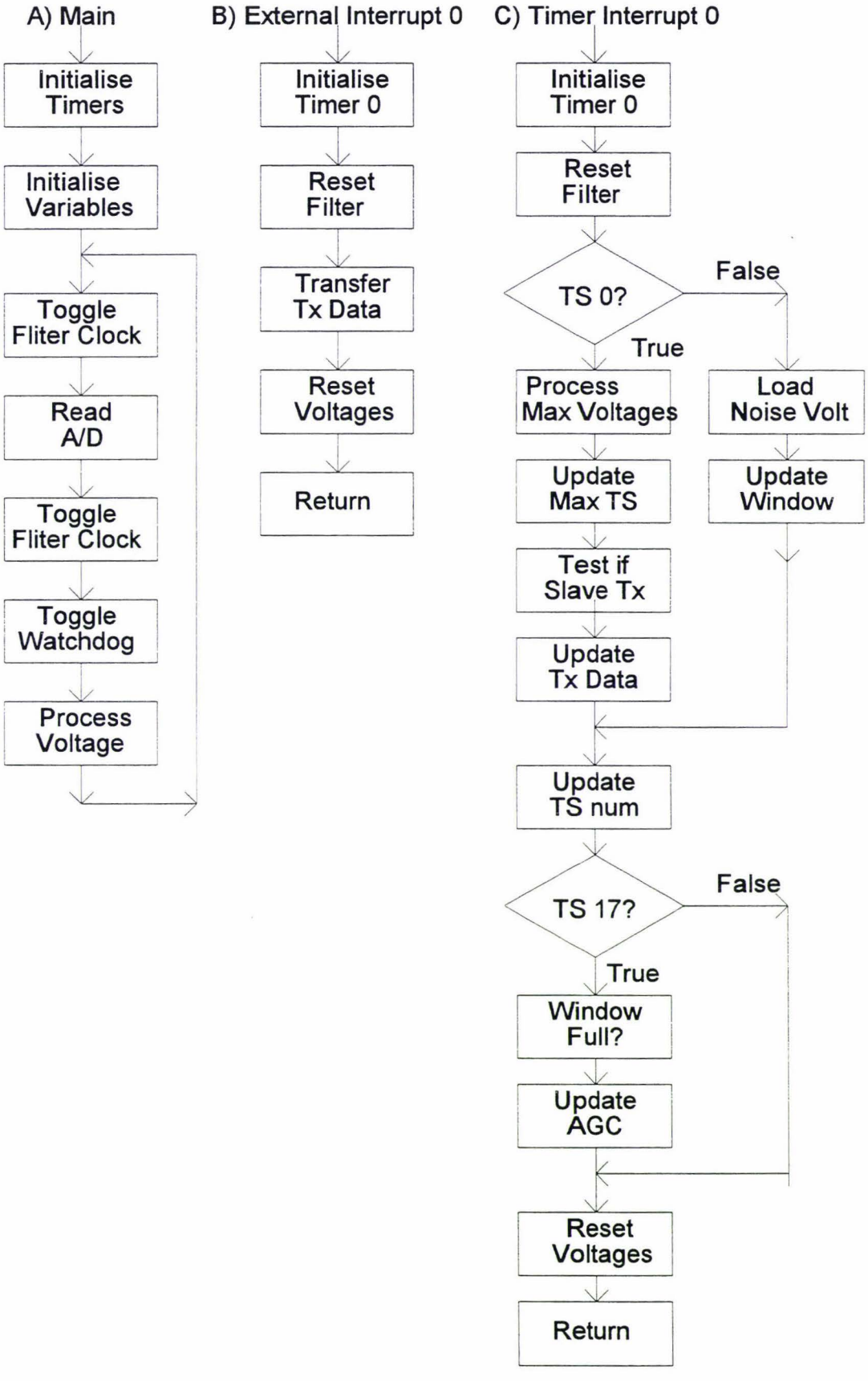


Figure 5.38 Filter Microprocessor Software.

A) Main Loop

The main loop of the program is responsible for keeping the filter clock toggling at 12.82kHz. During the first half of this loop the microprocessor reads a voltage value from the A/D. During the second half of this loop the software updates the maximum filter output for each timeslot used for the algorithm to determine if there is a slave transmission.

The algorithm used to detect a slave transmission looks for an increasing series of peaks with a minimum spacing (as shown in Figure 5.39). This helps the process to reject any noise peaks in the filter output.

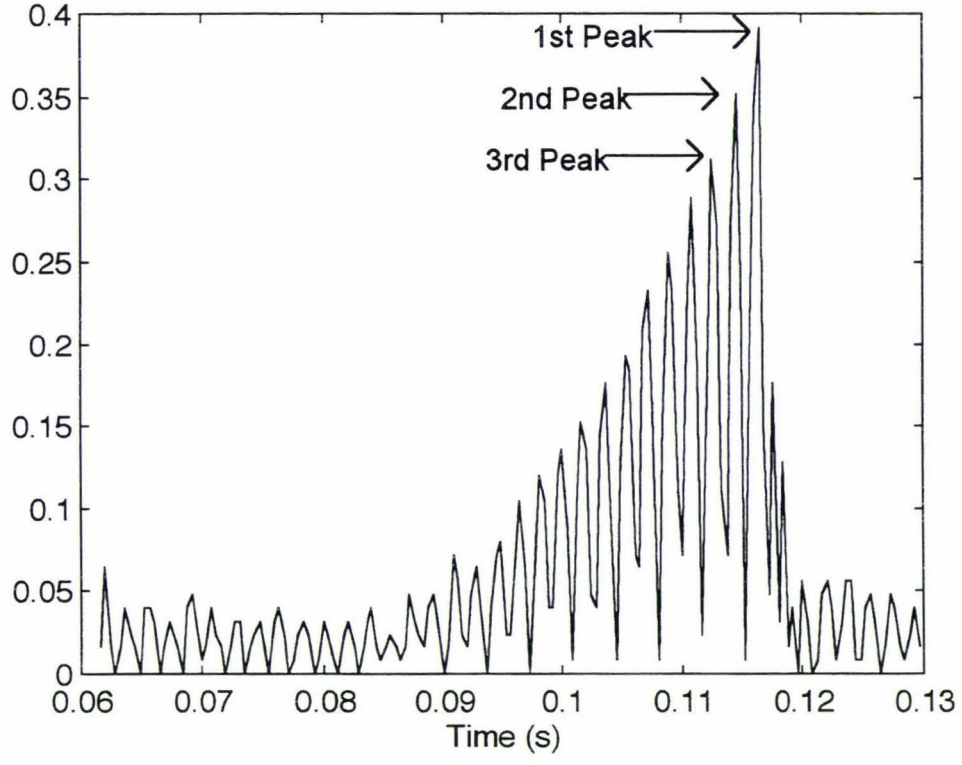


Figure 5.39 Filter Output Peaks.

As discussed in section 5.2.7.4, 23 measurements are taken for each of 546Hz filter output peaks. The software holds a count of how many measurements have been taken since a peak was last saved. If the last voltage reading is greater than the highest measured so far, then if the count is less than 15 the measurement is assumed to be in the same voltage peak and the highest reading is replaced with the last measurement. If the count is greater than 15 the peak is assumed to be a new peak. Thus the third highest is replaced with the second highest, the second highest replaced with the highest, and the highest replaced with the last reading. Whenever a reading is updated the count is reset. Figure 5.39 shows an example filter output, with the three highest peaks labelled.

For the software to determine that there was a slave transmission in the timeslot, two conditions must be met. The first is that the third highest peak must be significantly

greater than the highest noise peak. The second condition is that the third highest peak must be at least half the amplitude of the highest peak.

B) External Interrupt 0

The external interrupt indicates the presence of a energiser pulse. This is used to synchronise the timeslots between the slaves and the master unit. The interrupt procedure first loads timer 0 ready to trigger the start of the next timeslot. The filter is then reset by calling a subroutine dedicated to toggling the filter clock as fast as possible. The previous energiser pulse slave transmission data is then transferred to the user interface microprocessor. Note that the timeslot zero filter output is still required even thought there is no slave transmissions in this timeslot. The program must therefore transfer data to the user interface microprocessor and ensure the filter clock is toggled at the correct speed simultaneously. After the data transfer, the maximum voltage values are then reset ready for the next timeslot.

C) Timer 0 Interrupt

The timer 0 interrupt is called at the end of every timeslot. First the filter is reset by calling the reset filter routine. If this is the end of timeslot 0, the filter output is used to find the noise voltage. This is the maximum filter output for this AGC setting, with no slave transmissions present. This value is stored and will be subtracted from future readings to make sure that the reading is larger than the noise voltage. A time window is held for the AGC circuit. This window is long enough to ensure that there is at least one transmission from every slave unit before the maximum filter output reading is used to adjust the AGC. The current window length is updated in timeslot zero.

If the interrupt is not for timeslot 0 and the highest peak in the timeslot is greater than the highest reading in the current window the highest window reading is updated. Next the timeslot peak filter output voltage readings are tested to see if there is a transmission and the transfer data is updated.

The timeslot number is then increased to signify the start of a new timeslot. If the timeslot is now timeslot 17 (ie. after all slave transmissions have finished) a test is performed to see if the AGC window length has reached its maximum value. If so then the maximum reading in the window is used to determine if the AGC needs adjusting. If the maxreading is below the AGC lower threshold then the AGC is increased to boost the filter output. If the maxreading is above the AGC high threshold then the AGC is decreased to drop the filter output. The window maxreading and the current window length is then reset.

At the end of the timer 0 interrupt procedure the timeslot filter output voltage readings are reset ready for the next timeslot.

5.5.2 User Interface Microprocessor

The user interface microprocessor decodes the slave units voltage from the data transmissions from the filter microprocessor. It also displays the slave unit voltages and controls alarm annunciation depending upon the alarm thresholds set by the user.

5.5.2.1 Opto-Receiver

The Opto-Receiver circuit is used to tell the user interface microprocessor which slave units transmitted in the previous gap between energiser pulses. A start bit causes an external interrupt for the microprocessor, and indicates that the transfer follows. The speed of transmission is approximately 10kbps. The circuit for the opto-receiver is shown in Figure 5.40.

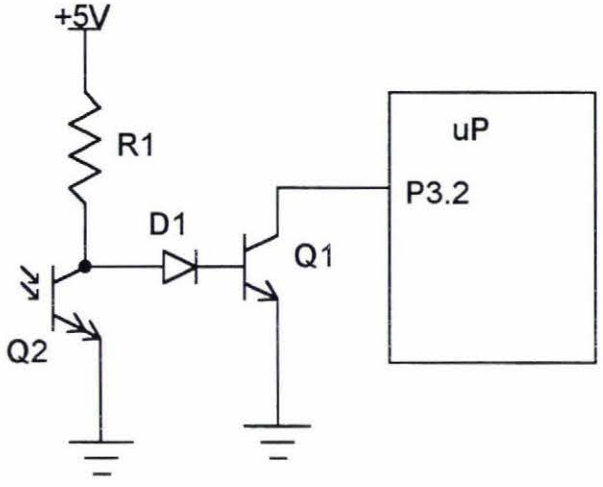


Figure 5.40 Optical Receiver Circuit.

In Figure 5.40 Q1 is required to ensure that the switching of P3.2 is from Vcc to ground and not Vcc to 0.7V as the minimum voltage across Q2 (a darlington photo-transistor) is 0.7V. A diode (D1) is added to ensure that the voltage across Q2 in the on state (0.7V) is not enough to turn Q1 on. The 0.7V difference at the collector of Q2 between on and off states ensures fast switching of Q2. R1 limits the current through the opto-transistor and the base of Q1.

5.5.2.2 Alarms

There are two audible devices that the microprocessor must control. The first is a buzzer. This will sound momentarily whenever a key is pressed and whenever the siren is on. The circuit is shown in Figure 5.41.

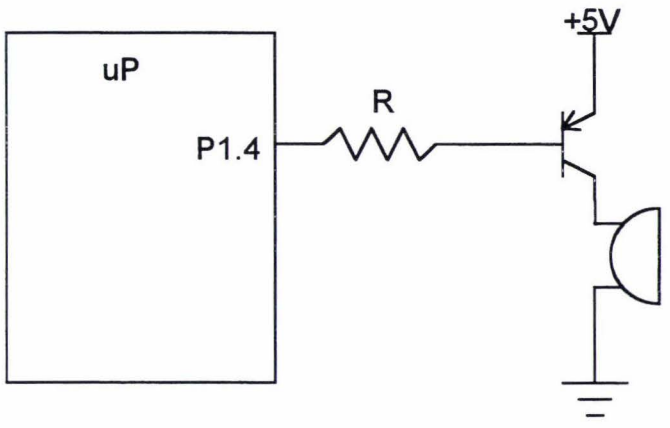


Figure 5.41 Buzzer Circuit.

The second audible output is a siren output. The siren is not included in the master unit, rather a 12V DC output is provided for the user to connect a siren to. The output is switched using transistors to provide a 300mA supply at 12V.

5.5.2.3 Reset/Watchdog Timer

Similar to the filter microprocessor a reset and watchdog timer is included in the circuit for the user interface microprocessor. The circuit is shown in Figure 5.42.

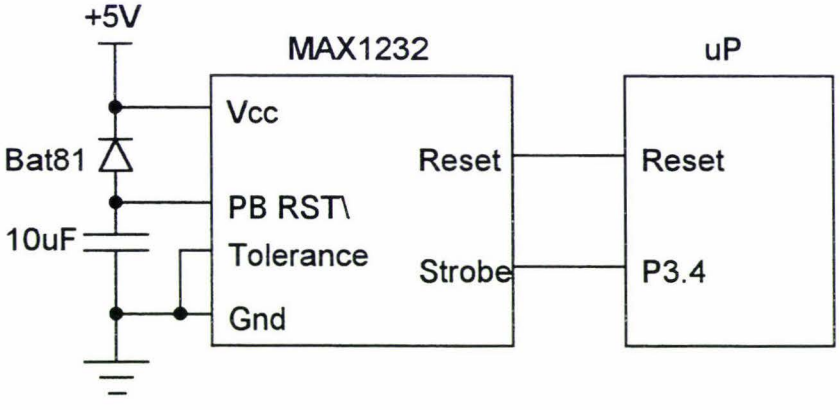


Figure 5.42 User Interface Microprocessor Reset Circuit

5.5.2.4 EEPROM

It is desirable that after recovery from a power failure the user should not have to re-program the master unit with which slaves are detected, or an alarm that was previously acknowledged require acknowledgment again due to the power failure. To achieve this a serial EEPROM IC has been included in the master unit. This is used to store: the desired measurement accuracy, the thresholds for each slave, if each slave has been detected and which alarms have been acknowledged. The circuit for the RAM is shown in Figure 5.43.

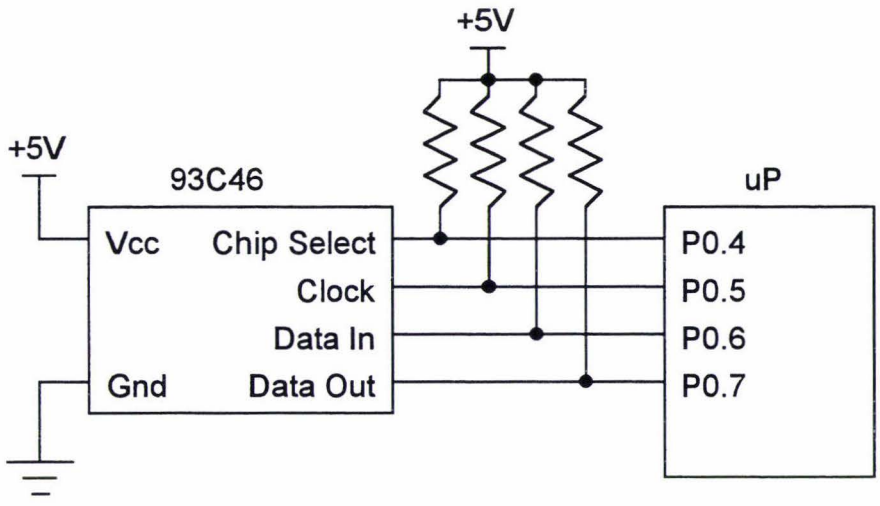
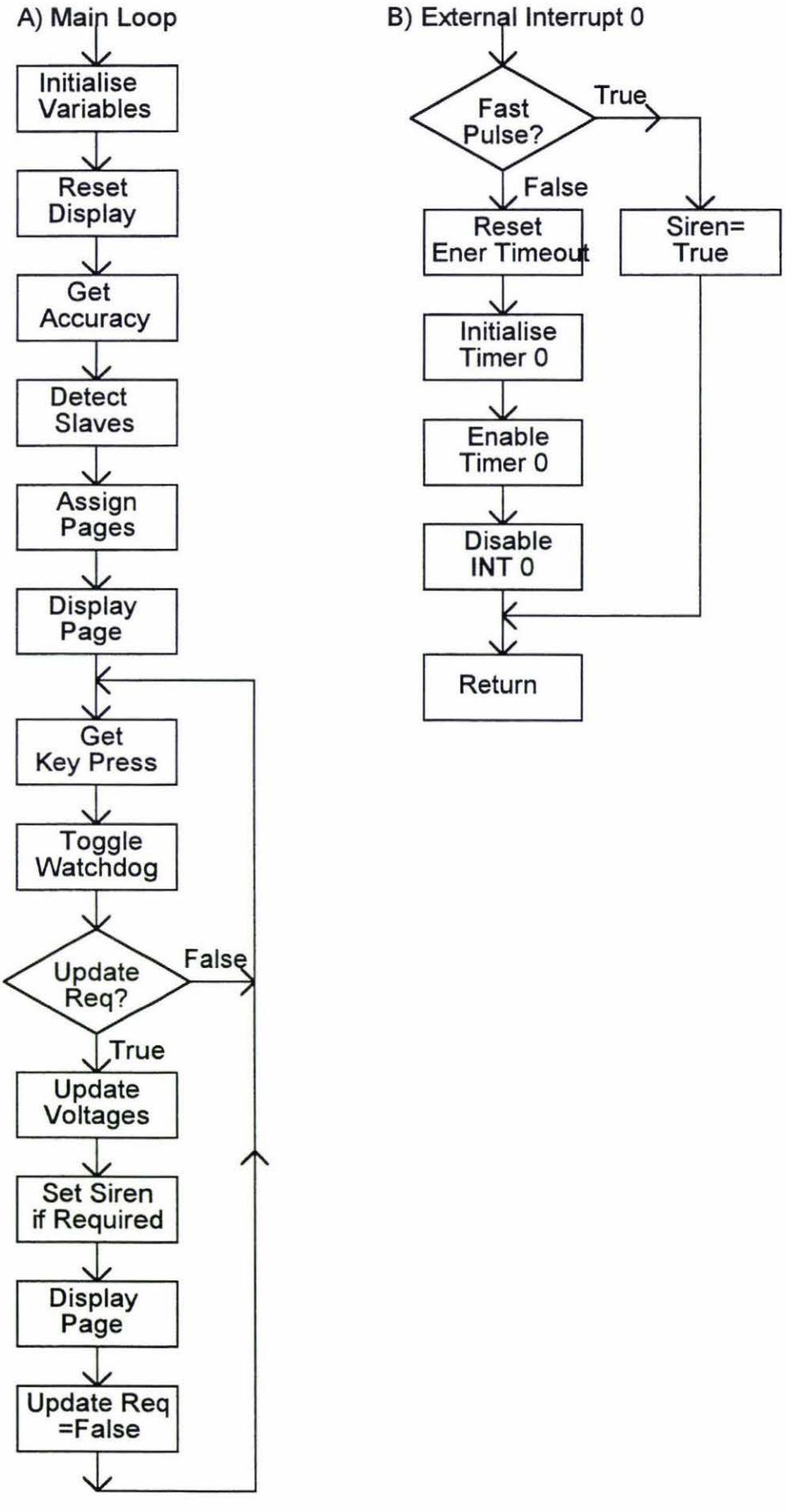


Figure 5.43 EEPROM Circuit.

Note that external pull-ups are required as there are no internal pull-ups for port 0, when not in memory addressing mode.

5.5.2.5 Software

The overall structure of the master unit user interface microprocessor software is shown in Figure 5.44. Appendix 7 shows the user interface microprocessor software itself.


A) Main Loop
Initialise Variables
Reset Display
Get Accuracy
Detect Slaves
Assign Pages
Display Page
Get Key Press
Toggle Watchdog
Update Req?
False
True
Update Voltages
Set Siren if Required
Display Page
Update Req =False
B) External Interrupt 0
Fast Pulse?
True
False
Reset Ener Timeout
Siren= True
Initialise Timer 0
Enable Timer 0
Disable INT 0
Return

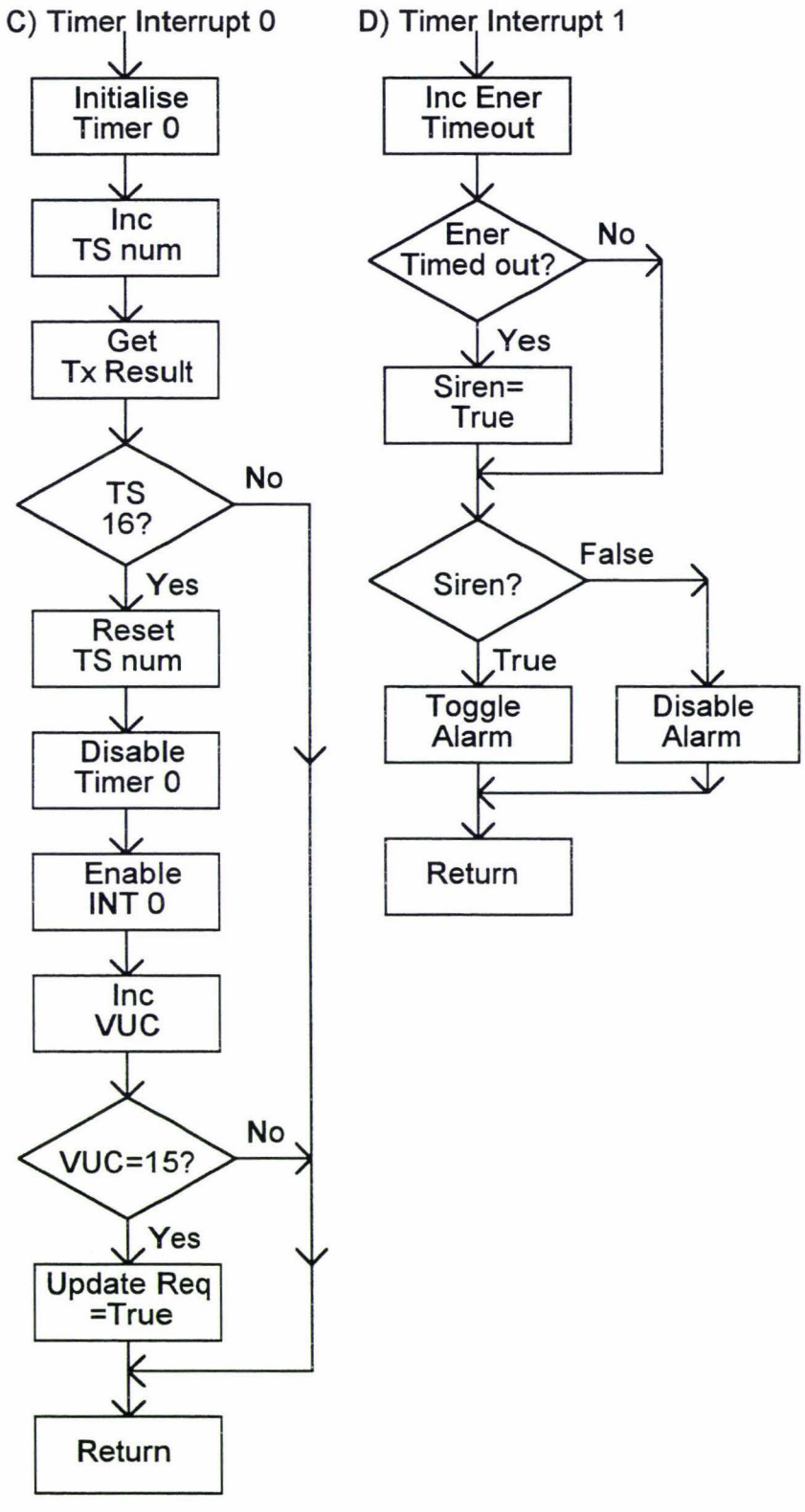


Figure 5.44 User Interface Microprocessor Software

A) Main Loop

After switching on, the master first asks for the accuracy setting desired by the user. Four options are available. The options and their accuracy are summarised in Table

5.4. The pulse history indicates how many pulses the master uses to determine the voltage reading. After setting the accuracy it is stored in the EEPROM.

Table 5.4 Accuracy Settings

Setting	Pulse History	Transmission Accuracy
Low	30	3.3%
Medium	60	1.7%
High	90	1.1%
Super	120	0.8%

If no accuracy is specified then the previous accuracy is loaded from the EEPROM. The user is then asked if he/she wishes to re-detect slave units on the fence. If so then the master will wait for 120 pulses to allow the slave unit to power up on the fence line, and to send a reply of medium accuracy. If the master detects a slave responding with a voltage greater than 1.5kV then this slave is 'detected', and the status stored in the EEPROM. If the user does not wish to re-detect the slave units, then the previous slave status is loaded from the EEPROM, and a time buffer is allowed for the slaves to power up and send a reply before any alarm is set.

The main program for the master unit consists of a loop which first calls the Getkey routine to see if a key is pressed. The Getkey routine checks if any of the four keys are pressed. If so the button timer is reset and the key pad timeout counter is turned on. Each key is then individually checked and the appropriate subroutine called to handle the keypress. A buzzer is then turned on for approximately 3ms to acknowledge the keypress to the user. The routine will then wait until the key is released plus a further period of time to ensure the routine is not re-entered because of key bounce.

A.i) GetKey Procedures

If the Acknowledge Alarm key is pressed then the ackalarm routine is called. This procedure stops the alarm and then uses a word (alarmacknowledge) to hold the status of which slaves are in alarm such that at the next update of the voltage values the alarm will not be turned on again.

If the Set Threshold key is pressed, the setthreshold procedure is called. A setmode flag is used to remember if the user is currently setting the threshold for a slave unit. If this flag is not set upon entry to the routine (the user is going into set mode), it is set and the blink is turned on. If moving out of setmode, then blink is turned off, and the new threshold saved to the non-volatile RAM. The following routines are then called: testalarm, assignpages and display page.

The Page Down/ 1000's key function is decided by the setmode flag. If not in set mode then the disnextpage procedure is called. This increases the current page number and then calls the displaypage procedure. If in set mode then the thousands procedure is called. This increases the threshold of the selected slave to 1kV and displays the new threshold. If the threshold rises above 10kV then the slave is deregistered (after leaving setmode).

The Move Down / 100's key's function is also decided by the setmode flag. If not in set mode then the down procedure is called. This simply increments the current selected line. If no slave is present in the line or line 5 is selected, then the current line is set to line one. This provides wrapping within a page. If in setmode the hundreds procedure is called. This increases the selected slaves threshold by 100V. If it tries to rise above nine then it is reset to zero.

A.ii) Voltage Processing Procedures

After the Getkey routine the main loop will check the update required flag to see if an update of the voltage values is required. This flag is needed as a procedure using the same register banks as the interrupt, may not be called from the interrupt procedure, thus a flag is set in the interrupt procedure and actioned in the main loop. If an update is required then the updatevoltages procedure is called. This calculates the slave voltage using the number of positive responses and the accuracy setting. The microprocessor has the number of responses in the last eight, fifteen second windows stored. It will use an even number of these fifteen second windows to calculate the result and update the display. For example on the 'Low' accuracy setting the number of responses is divided by 0.3 to obtain the fence voltage in kilovolts. It then resets all of the slave responses to zero and calls the assignpages, testalarm and display page routines.

The assignpages routine is responsible for inserting each of the registered slaves into any of the four pages available for the display. The slaves are ordered numerically into the pages if they are registered. Thus if slave 2 is not registered, then the first page will consist of slaves 1,3,4,5. The slaves are then tested for alarm conditions. The testalarm routine is complicated by the alarm acknowledge function. First each slave is checked to see if it is in alarm state (Threshold>Voltage) if so then the appropriate bit of an alarmstate word is set. For each slave if the slave is no longer in an alarm state then the appropriate bit of the alarmacknowledge word is cleared. If after setting all of the alarmstate bits alarmstate>alarmacknowledge then the siren flag is set.

The display page routine simply uses the data generated in the assign pages routine to cycle through each of the four display lines and write the slave unit details from the current selected page to that line on the LCD display.

B) External Interrupt 0

The external interrupt is used to signal the start of a transmission from the filter microprocessor. Thus it must initialise the timer 0 interrupt which is the bit rate timer for the transmission results. The external interrupt is disabled to stop it triggering during the data transmission. If the energiser is fast pulsing (pulsing at greater than 1 pulse per second), then the results are not displayed, as their accuracy can not be guaranteed. Fast pulsing is considered a fence fault and the alarm is also set.

C) Timer 0 Interrupt

The timer 0 interrupt first increases the timeslot number then reads the data transmission result. If it is reading the 16th bit then this is the end of the transmission. The timer is then disabled and the external interrupt enabled ready for the next energiser pulse transmission. The voltage update counter (VUC) is increased. This is the number of energiser pulses before the master unit will update the voltages. The larger this count the more accurate the result, yet the longer a result takes to obtain. A count of 100 was chosen for the update time giving a 1% accuracy in the results.

D) Timer 1 Interrupt

The timer 1 interrupt is used as a 65ms tick. This tick is used for controlling keypad timeouts and setting the timing flag to false ensuring the set threshold routine will be called in the main program loop. The duration since the last energiser pulse is also increased in this routine. If this is equal to its timeout value then the noenergiser flag is set and the siren flag is set. This will cause the main loop to run a routine which displays a message signalling that no energiser is detected. Care is made sure that the timer does not overflow, but also this alarm is only called once. This allows the acknowledge alarm procedure to turn the siren off. The no energiser message will still be displayed.

The timer 1 interrupt is also responsible for modulating the siren. Two siren speeds are available. The first turns the siren on with a 50% duty cycle. After 30 seconds the siren is turned on with a 5% duty cycle. This is helpful to be used as a reminder to the user of the alarm condition, and does not keep the siren going continuously when the user is not present.

5.6 Power Supply

This section outlines the requirements and details of the master unit power supply.

5.6.1 Power Supply Requirements

It is desirable for matters of cost, safety standards and consistency with current Speedrite products, that the master unit power itself from a mains plug pack. A 15VAC plug pack was chosen.

The safety requirements stipulated isolation of the keypad and display from the filter, such that if the earth should fail, there is no risk of electric shock from touching the master unit. Thus separate power supplies were required for each of the two microprocessors.

The filter microprocessor system requires both a +5V and a -5V power supply for the analogue circuitry.

The user interface microprocessor system has a 12VDC siren output while the rest of the circuitry requires a +5V supply.

The current consumption of the various components is listed below according to which supply they require:

Table 5.5 Filter Microprocessor System Power Consumption.

Filter Microprocessor System	Current (mA)	
	+5	-5
Microprocessor 89C2051	14	
Switched Capacitor Filter MF10	12	12
Switched Capacitor Filter MF10	12	12
Analogue Switch HC4316	0.1	
Analogue Multiplexer HC4051	0.001	
Dual Operational Amplifier NE5532	8	8
Reset IC MAX1232	0.001	
A/D TL549	2.5	
LED (Opto-link)	5	
Dual Comparator LM319	5	5
Filter Load	5	5
Total Current	63.602	42

Table 5.6 User Interface Microprocessor System Power Consumption.

User Interface Microprocessor System	Current (mA)	
	+12	+5
Microprocessor 89C52		25
Buzzer		6
Display LM044L		3.5
Opto-Receiver		5
EEPROM 93C46		0.001
Reset IC MAX1232		0.001
Siren	300	
Total Current	300	39.502

5.6.2 Power Supply Design

The isolation between the User Interface Microprocessor System and the Filter Microprocessor System of the master unit is achieved using a transformer. An optical link will provide the data transfer between the two microprocessors.

The design for the transformer is shown in equations 5.15 to 5.17.

$$V_{AC} = \frac{V_{DC} + V_{Drop} + V_{Diode} + V_{Ripple}}{\sqrt{2}} \qquad (5.15)$$

$$= \frac{5+3+0.7+2}{\sqrt{2}}$$

$$= 7.5V$$

$$I_{AC} = I_{DC} \times TU \times ff \qquad (5.16)$$

$$= 0.1056 \times 0.5 \times 1.3$$

$$= 0.0686A$$

$$VA = V_{AC} \times I_{AC} \qquad (5.17)$$

$$= 2 \times 7.5 \times 0.0686$$

$$= 1.03W$$

The windings should be a 1x15V primary and 2x7.5V secondary to allow for centre tapped rectifiers to give the dual rail supply. The capacitor values are found from equations 5.18 and 5.19.

$$C2 = \frac{I\tau}{\Delta V} \qquad (5.18)$$

$$= \frac{0.0636 \times .0.01}{2}$$

$$= 318\mu F$$

$$C3 = \frac{I\tau}{\Delta V} \qquad (5.19)$$
$$= \frac{0.042 \times .0.01}{2}$$
$$= 210\mu F$$

Thus a 470μF and 220μF capacitors were used. For the +5V and -5V supplies this gives a ripple of 1.3V and 1.9V respectively. The filter circuit +5V and -5V voltage regulators must be able to supply 63mA and 42mA respectively. The MC78L05AC and the MC79L05AC can both supply up to 100mA and thus are sufficient. The +5V regulator will have a voltage drop of 3.7V plus a 1.3V ripple.. The average power dissipation in the +5V regulator will be a maximum of 0.27W. The TO-92 package version of the MC78L05AC can easily dissipate this power up to 100°C ambient temperature. The power dissipation for the -5V regulator with a drop of 3.1V and ripple of 1.9V should have a maximum power dissipation of 0.17W. Thus the TO-92 package is again adequate.

The equations for the User Interface Microprocessor System power supply of the master unit are given in equations 5.20 to 5.21.

$$V_{Ripple} = V_{AC} \times \sqrt{2} - V_{Drop} - V_{Diode} - V_{DC} \qquad (5.20)$$
$$= 15 \times \sqrt{2} - 2.3 - 1.4 - 12$$
$$= 5.5V$$

$$C1 = \frac{I\tau}{\Delta V} \qquad (5.21)$$
$$= \frac{0.34 \times .0.01}{5.5}$$
$$= 680\mu F$$

A 1000μF capacitor was used as this is the nearest preferred value. In this case the ripple becomes 3.4V. The 12V regulator drop then increases to 5.5V. Under operating conditions the 12V regulator must dissipate an average of 7.2V at a current of 300mA. This is a maximum average power dissipation of 2.16W. A regulator in a large package such as a TO-220 with a heat-sink is thus required.

For the +5V display power supply the worst case voltage drop (when the siren is not operating) is shown in equation 5.22.

$$V_{Drop} = V_{AC} \times \sqrt{2} - V_{Ripple} - V_{Diode} - V_{DC} \qquad (5.22)$$
$$= 15 \times \sqrt{2} - 0.4 - 1.4 - 5$$
$$= 14.4V$$

At a current of 40mA the maximum average power dissipation in the regulator would be 0.6W. At this power dissipation the TO-92 package MC78L05 would only work to 50°C. A TO-39 package would allow the ambient temperature to rise to 75°C without a heatsink and well over 100°C with a heatsink.

All of the diodes would have to block at least 30V. Thus 1N4001 50V 1A diodes are sufficient.

The circuit diagram of the master unit power supplies is shown in Figure 5.45.

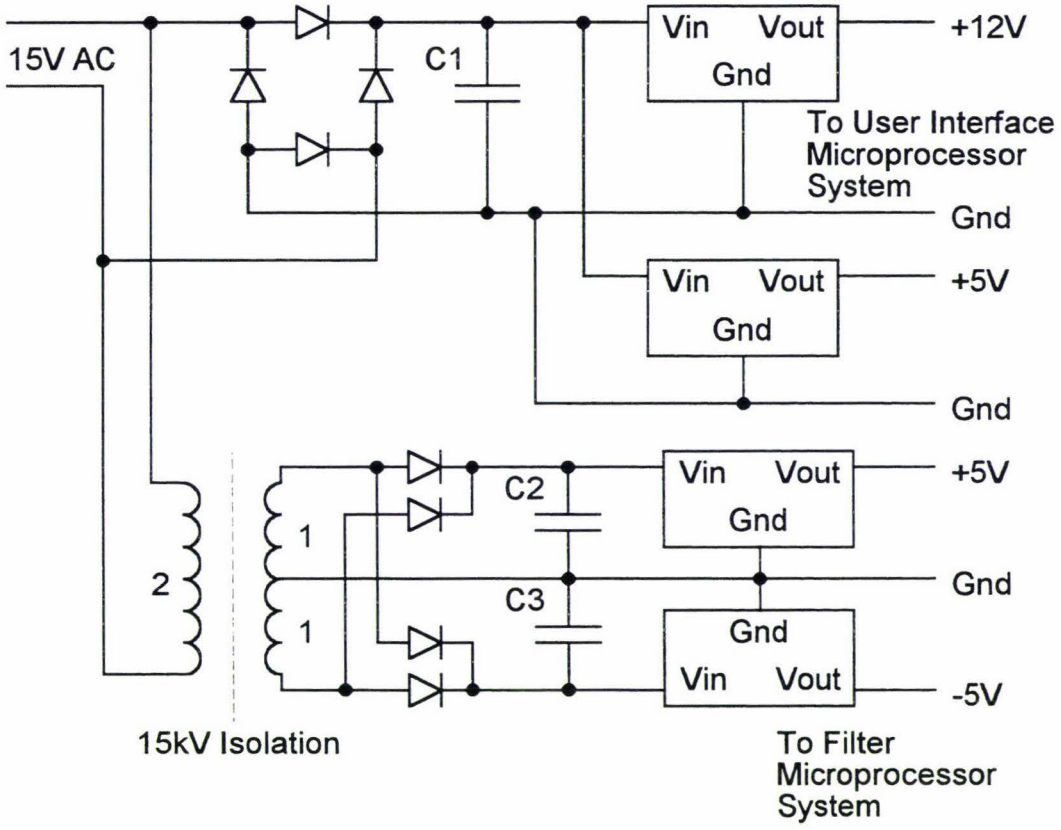


Figure 5.45 Master Unit Power Supply.

5.7 PCB Layout

The master unit components are separated onto three different PCBs. The first contains the voltage divider that is connected to the fence. The second contains the filter circuitry, and is connected to the third by an optical link and transformer connections. The third board has the user interface microprocessor, along with the power supply connections and plugs for the keypad and display. This section details the reasons for the arrangement of the components on these PCBs.

5.7.1 Fence PCB

This PCB prevents any fence voltages from getting onto the filter PCB. It contains two voltage dividers which allow only low voltages onto the filter PCB.

The first divider is used for the filter and AGC circuit. The second divider is that which is used in the energiser detection circuitry.

This board is single sided in order to keep costs to a minimum.

The prototype Fence PCB is shown in Figure 5.46.

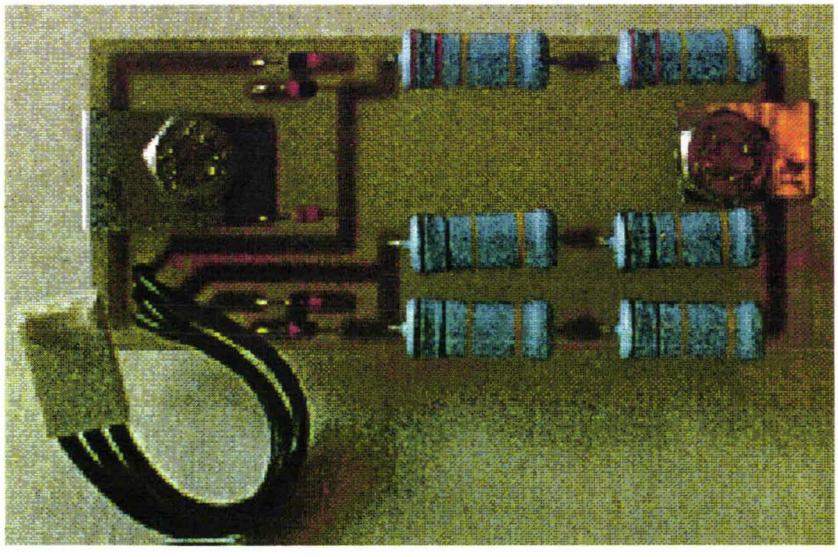


Figure 5.46 Prototype Fence PCB

5.7.2 Filter Microprocessor System PCB

This board contains the filter for detecting the slave unit transmissions and the energiser pulse detection circuitry. The optical link and power supply is the only connection with the User Interface PCB. The separation from the User Interface Microprocessor System allows the 15kV isolation requirement to be achieved easily.

A double sided board has been used to take maximum advantage of a large ground plane to reduce system noise.

Care has been taken with the board layout to ensure that the fast digital signals are kept as far away from the filer as possible. the filter clock (a fast changing digital signal) has been surrounded on both sides by a ground track, and above by a ground plane in order to reduce the coupling to other tracks as much as possible.

A large ground plane has been used to reduce the effect of interference as much as possible.

The prototype Filter Microprocessor System PCB is shown in Figure 5.47.

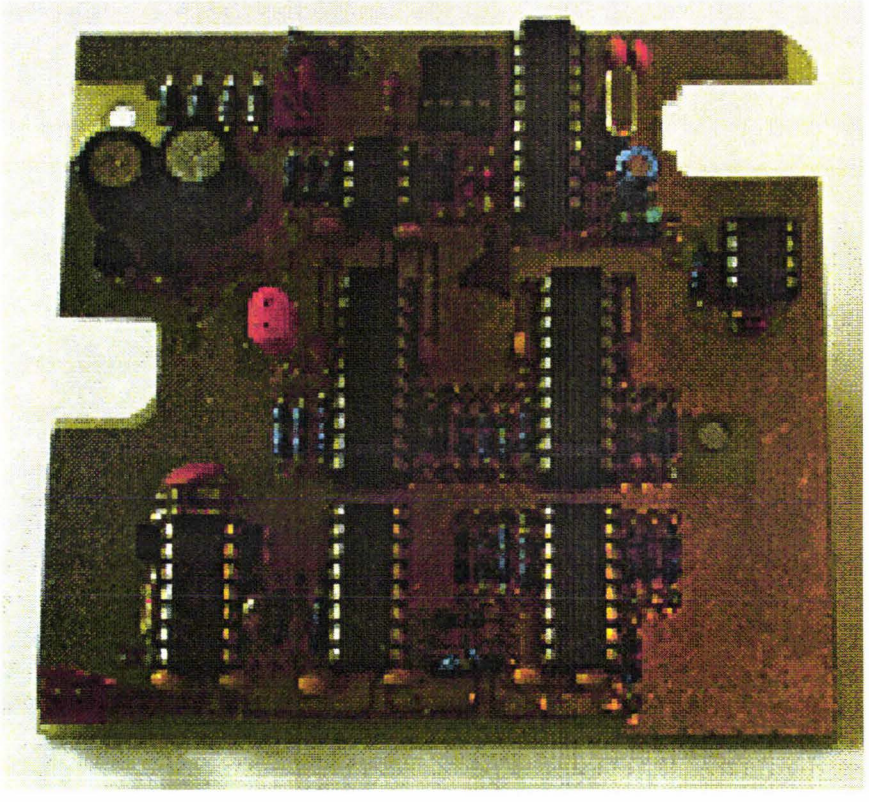


Figure 5.47 Prototype Filter Microprocessor System PCB

5.7.3 User Interface Microprocessor System PCB

This PCB has very few components as the microprocessor is only responsible for display, keypad and buzzer siren management. Thus connections for the display and keypad are provided at the top. the AC input and siren output are placed at the bottom of the board such that they appear underneath the unit to allow as much protection from the elements as possible.

A single sided board has been used in order to minimise the cost of the master unit.

the prototype User Interface Microprocessor System PCB is shown in Figure 5.48.

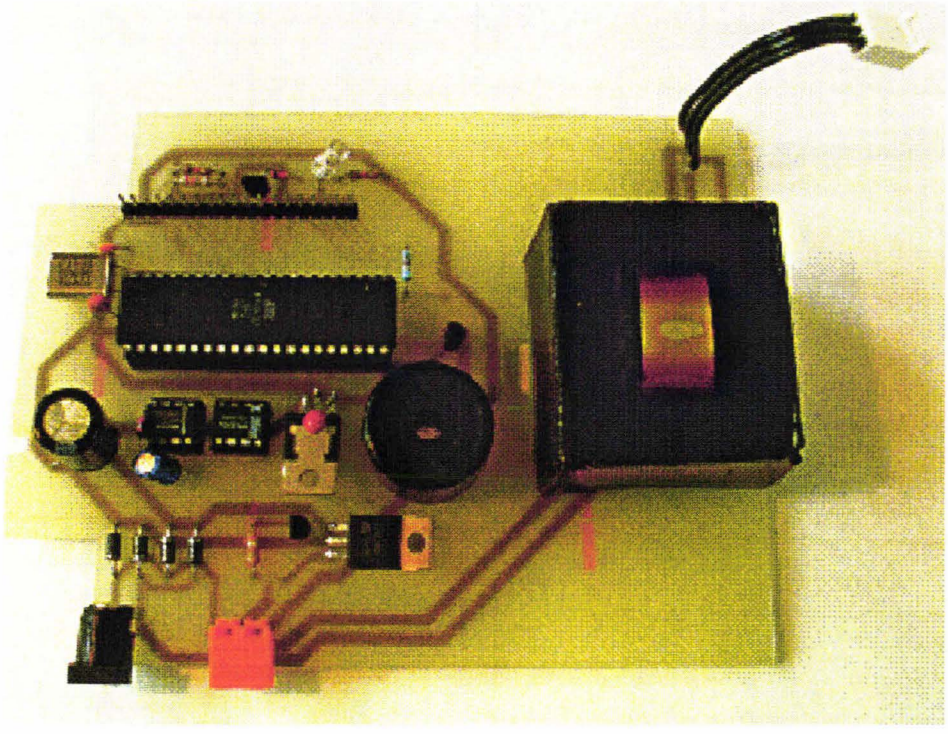


Figure 5.48 Prototype User Interface Microprocessor System PCB

5.8 Packaging

This section covers a brief look at the packaging of the master unit.

Similar to the slave unit, it is again most cost effective if the master unit can be placed in an existing Speedrite case. These are generally electric fence energiser cases, most with three output terminals. The master unit has only two fence terminals and very few cases were appropriate.

Figure 5.49 Master Unit Packaging

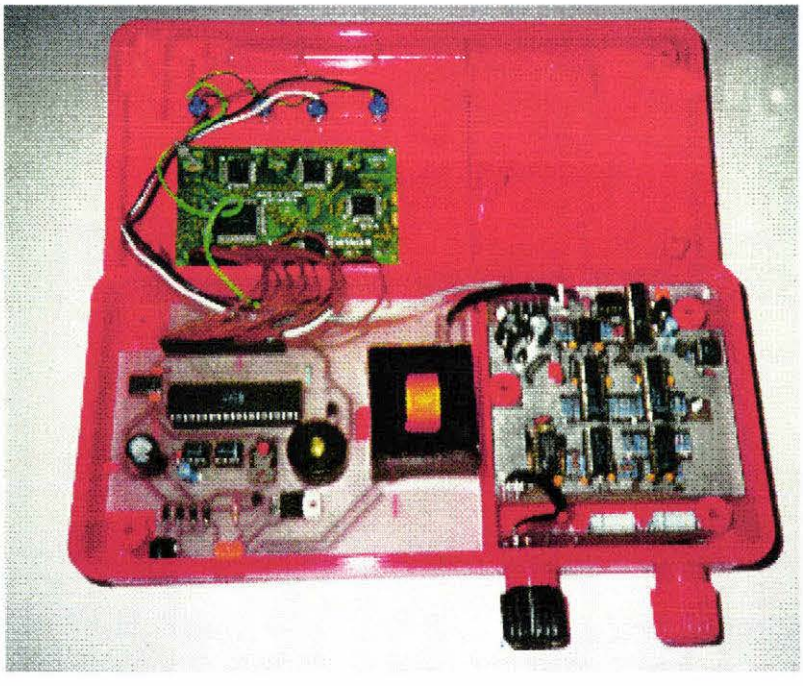


Figure 5.50 Master Unit Assembly

Figures 5.49 and 5.50 show the prototype master unit packaging and assembly.

The case chosen has two compartments with a separating wall forming a natural division for the filter PCB and the user interface PCB. The case is wall mountable, and all PCBs can easily be mounted inside. A hole must be cut in the top of the case for the display. Tools must also be made for inserts in the mould to create cavities to hold the alarm output plug and the mains input socket from the plugpack.

For the prototype the keypad is consturcted using push buttons. In the production version the keypad is made with membrane switches. This allows a cover to be placed over the display to stop water entering the unit. This cover can be extended across the top of the master unit. It also allows for graphics to be placed ontop of the unit and then protected by the membrane keypad cover.

Chapter 6
EFMS Testing

This chapter discusses the brief field testing of the EFMS.

The aim of this testing is to verify the operation of the EFMS and to check that there are no circumstances that have not already been considered that could effect the operation of the system. To ensure this, a 'real' electric fence system must be used. Section 6.1 presents the layout of the farm that was used to perform the tests.

To accomplish the testing eleven slave units and one master unit were constructed.

Several simple tests are presented each of which focuses on a potential operating problem for the EFMS. These tests are presented in section 6.2.

Note that the results presented in this chapter represent only the start of the field testing associated with the EFMS. It is expected that the EFMS will undergo much more testing of an extended duration before thought is given to commercial release of the product. The extended duration of these tests prevents them from being presented here.

6.1 Test Farm Layout

The farm used is located fifteen minutes drive from Massey University. The farm encompasses ten acres and extensive use is made of an electric fence to control stock. The fence is used to control cattle, sheep, goats and pigs.

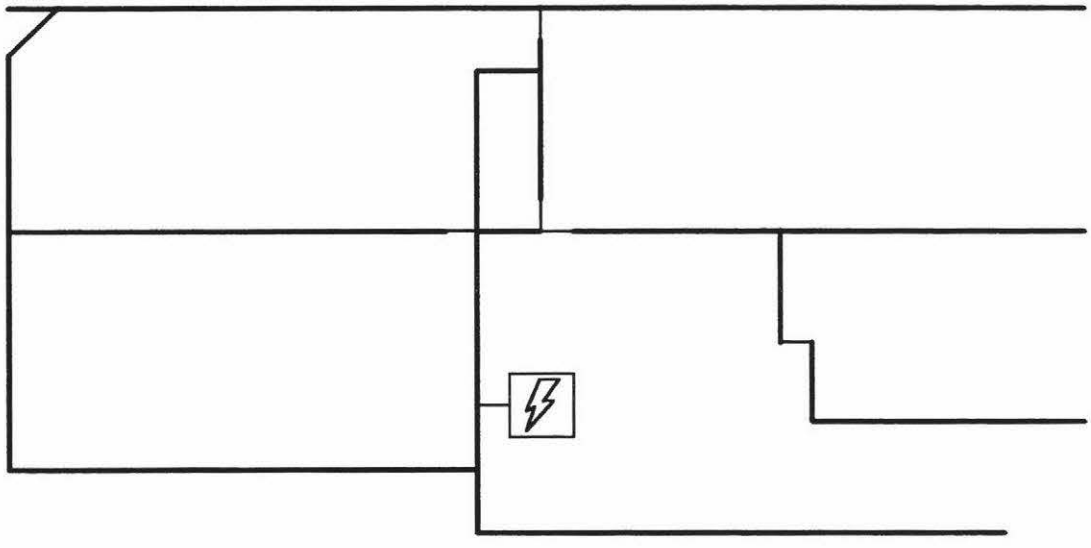


Figure 6.1 Test Farm Layout

The energiser used was a Speedrite SM2400. This is a 2.4J RFI suppressed energiser. The fence is in working order, yet is significantly loaded.

6.2 EFMS Tests

6.2.1 Test 1: Slave Power-Up

This test involves connecting all of the slave to various points on the fence and monitoring which slaves power-up. It checks that all will power-up when they are each at different peak voltages etc.

Power-up is verified by use of a scope on the transmission pin of the slave unit microprocessor, to test that the microprocessor receives power and the code starts correctly, such that the microprocessor will transmit.

6.2.2 Test 2: Slave Accuracy

This test aims to test the accuracy of a slave unit over the required measurement range of the slave unit. The slave and master unit are connected to an energiser which can have its peak output voltage divided down to achieve lower output voltages.

The system is setup with one slave unit and the master set on the highest accuracy setting. The energiser is then adjusted to a required peak output voltage. An oscilloscope is used to measure the peak fence voltage. The slave voltage is read from the master unit output.

6.2.3 Test 3: Slave Accuracy Variation

This tests aims to test the variation in accuracy of the slave units measurement circuit and that the transmission system functions correctly. For this test all slaves are placed at the same fence location and the master is put onto the maximum accuracy setting.

The results of the test can be found from the display of the master unit. Any variation in the reading at the master can be attributed to inaccuracies in the measurement circuit components.

6.2.4 Test 4: System Performance

This test checks the function of the EFMS under normal operating conditions. The slave units are connected to various parts of the fence. The readings of the master unit were then verified with a scope or electric fence voltmeter.

The positioning of the slave units is shown in Figure 6.2.

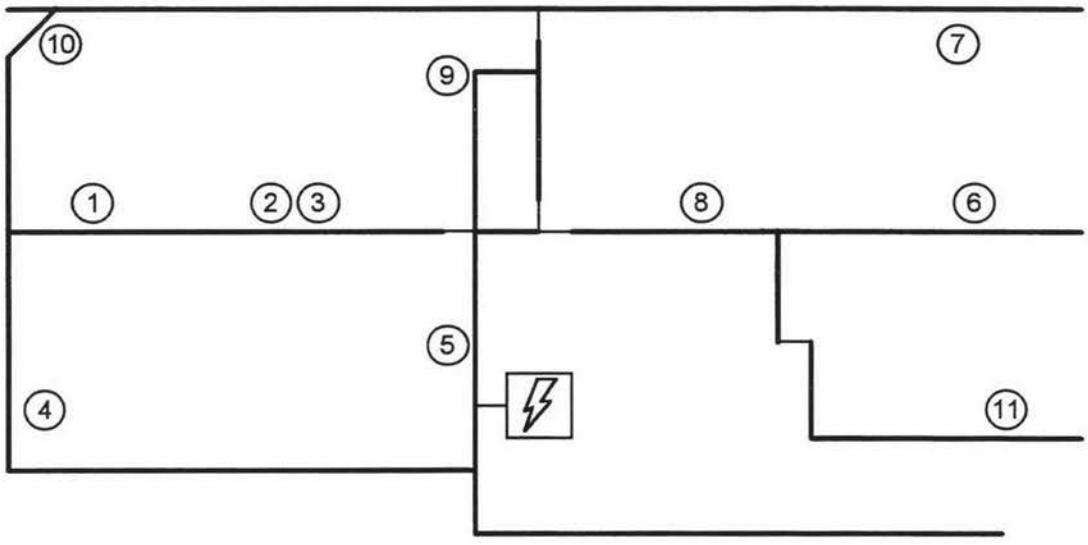


Figure 6.2 Test 4 Slave Unit Positioning

6.2.5 Test 5: System Performance With Induced Noise

Critical to the function of the EFMS is the noise on the fence line at 273Hz. In this test an attempt is made to induce noise (through the mains) to the filter in the master unit. The noise was simulated by using an electric drill from the same mains supply as the energiser and the master unit. The slave units were positioned as shown in Figure 6.2. The influence of the noise can be found by measuring the A/D input in the master unit.

6.3 Test Results

6.3.1 Test 1 Results

All slaves powered up and ran at five volts from the fence pulse

6.3.2 Test 2 Results

The voltage output from the slave unit for each peak voltage is shown in Table 6.1.

Table 6.1 Test 2 Slave Accuracy Results

Predicted Voltage	Measured Voltage
8.00	8.0
6.94	6.9
6.21	6.3
5.52	5.4
4.83	4.7
4.14	4.2
3.45	3.5
2.76	2.8
2.07	2.0
1.38	---

6.3.3 Test 3 Results

The voltage and variation for each slave is shown in Table 6.2.

Table 6.2 Test 3 Slave Variation Accuracy Results

Slave #	Voltage	Peak Fence Voltage
1	6.06	6.12
2	6.06	6.12
3	6.03	6.12
4	6.06	6.12
5	6.13	6.12
6	6.06	6.12
7	6.06	6.12
8	6.07	6.12
9	6.18	6.12
10	6.14	6.12
11	6.11	6.12

6.3.4 Test 4 Results

The voltage measured at each slave connection to the fence and the slave measurement is shown in Table 6.3.

Table 6.3 Test 4 Slave Performance

Slave #	Voltage
1	6.6
2	6.6
3	6.5
4	6.5
5	6.6
6	6.7
7	6.4
8	6.6
9	6.7
10	6.8
11	6.6

6.3.5 Test 5 Results

The voltage response from the slave units is shown in Table 6.4.

Table 6.4 Test 5 Slave Performance Under Noise

Slave #	Voltage
1	6.6
2	6.6
3	6.5
4	6.5
5	6.6
6	6.7
7	6.4
8	6.6
9	6.7
10	6.8
11	6.6

Note that the results shown in Table 6.4 are the same as those in Table 6.3 indicating that the connection of the electric drill made no difference to the operation of the EFMS.

An example of the noise at the A/D input caused by the electric drill is shown in Figure 6.3.

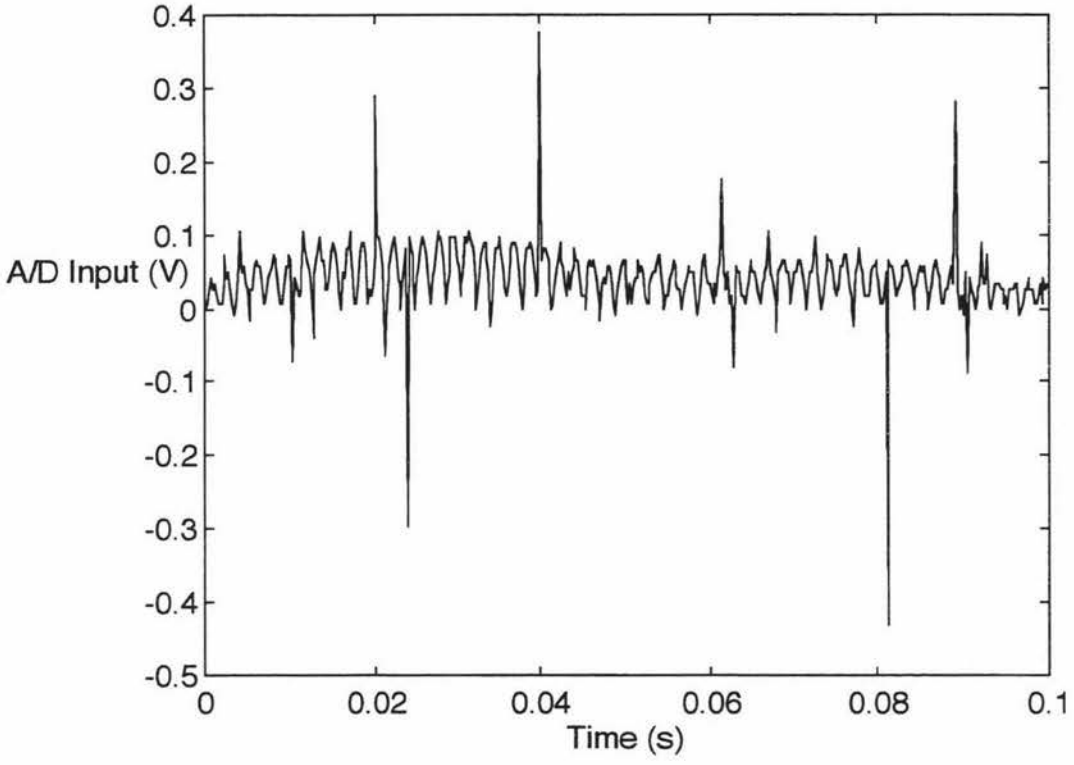


Figure 6.3 Induced Noise At A/D Input.

6.4 Test Conclusions

These tests are only the start of field testing of the EFMS.

The tests show that the slave units are all capable of powering up from the fence pulse. The tests also show that the slave units achieve a measurement accuracy of a few percent over the range of measurement. The slave units are able to reply on the fence wire to the master unit.

Test 5 shows that noise on the mains supply does not have a significant effect on the EFMS output.

Chapter 7
Conclusion

7.1 The Contribution Of This Work

7.1.1 Energiser And Fence Modelling

Models have been written to simulate the electric fence energiser and the electric fence itself.

7.1.1.1 Fence Model

This lumped parameter model represents an electric fence. The model is represented by Laplace equations, and a Matlab program is used to solve the model. It allows an external input to view the output at the end of an electric fence.

The model has been used in the energiser model to show the energy available to a slave unit at the end of a fence line, and alone to show the effects of the fence line on a transmission from a slave unit to the master unit.

7.1.1.2 Energiser Model

This model extends the electric fence model by including the energiser output circuitry to account for the interaction between the energiser and the electric fence.

Two parts to the model were developed. The first places a resistive load on the energiser. The second places an electric fence on the energiser.

This model accounts for the variation in the magnetising inductance of the energiser output transformer, by changing the inductance in accordance with the current flowing through the inductor.

After selection of parameters for the transformer gap and core losses, the simulations showed good agreement with measured results.

The model was used to show the energy available to a slave unit at the end of the fence. The model highlights that the construction of the transformer is a critical factor in determining the output of the energiser. The model can be used in the future as a design tool for the energiser by Speedrite International.

7.1.2 The EFMS System

A working Electric Fence Monitoring System has been developed which consists of a single master unit and up to sixteen slave units.

7.1.2.1 Transmission Algorithm

An algorithm for the communication between the slave unit and the master unit has been developed. This algorithm allows the slave unit to transmit the peak fence voltage to the master unit. The energy that is required for this transmission is shown to be related to the energy available to the slave unit for transmission.

This accuracy of this transmission is shown to depend upon the time given to transmit the result. For an increase in transmission accuracy the receiver only need aquire the result over a longer time period.

7.1.2.2 The Slave Unit

The slave unit has been designed and tested with good results. The unit is placed on the fence line and measures the peak voltage on the fence and transmits this voltage to the master unit. The slave unit is completely powered from the electric fence pulse. This removes the need for an external power source or batteries which need replacement periodically. The power supply design has been optimised to reduce losses.

Several options for a measurement circuit for the slave unit have been developed. The final circuit chosen is one that is cost effective, accurate and power efficient.

A transmission circuit is presented for the slave unit which performs an impedance transformation to lower the source impedance of the slave transmission circuit, and converts the square wave microprocessor output to a sine wave for energy efficiency in transmission. A model of this transmission circuit has been developed which was used to evaluate the response of the circuit to input waveforms, the performance of

the circuit, the response to different terminating impedances and the response from circuits with component errors.

A microprocessor powerup circuit has been designed that provides a large noise immunity while also providing instantaneous switching hysteresis.

Software has been written for the slave unit to allow the unit to perform the tasks required in the running of the slave unit.

7.1.2.3 The Master Unit

The master unit has been designed and tested with good results. The master unit detects slave unit transmissions on the fence line, calculates the peak fence voltage at each slave unit, and displays this for the user. The master unit also interfaces with the user allowing alarm points, for a siren should the peak fence voltage fall too low, to be set.

A non-linear chebyschev filter has been developed which has a very narrow bandwidth. By varying the filter clock speed the master unit is able to reset the filter. This removes artifacts from previous timeslots from the filter, allowing narrower guard bands between timeslots. For the same noise rejection specifications this doubles the number of slaves possible for the EFMS system. A detection algorithm has been designed and implemented to allow the master unit to use the filter output to decide if a slave transmission is present.

The master unit is protected from any damage from the energiser pulse, but still detects the energiser pulse in order to synchronise with the slave units. The display and keypad in the unit is isolated from the fence with an optical link and an isolating transformer to meet legislative safety requirements.

Two programs have been written for the master unit to allow the unit to perform the tasks required in the running of the user interface and the filter circuits.

7.2 Extensions To The Work

7.2.1 EFMS Cost Reduction

The EFMS is a commercial product, and cost is an important consideration. The slave output circuit accounts for a large proportion of the cost. Work on the transformer and inductors manufacturability could give some cost savings.

In the master unit the display and filter are the highest cost parts. The filter could be simplified by reducing the operation specifications or selecting alternative parts. The display cost could be reduced by seeking a functionally equivalent display from an alternative supplier.

7.2.2 EFMS Extension

At present the EFMS measures the peak voltage on the fence. The system could be extended to allow more functionality. Examples are

i) Remote Fence Turn off.
ii) Water Trough / Dam / River Heights
iii) Open Gate Detection
iv) Wind Speed
v) Rainfall

7.2.3 Model Improvements

The fence model is a lumped parameter model. This is adequate for most electric fence distances, but for very large fences the wavelength of the pulse becomes comparable to the length of the fence, and the model is not accurate. An extension to the model could be made such that transmission line equations were used instead of a lumped parameter model, allowing the use of very large electric fences possible.

Under very low load conditions the model output varys from the measured output of energisers. This suggests that the model could be improved possibly through a more fundamental study of ferrous electromagnetic system.

Appendix 1
The Fence Model

The fence model is used to model a typical electric fence of various lengths using a nominal Π equivalent circuit. The model is presented in chapter 3. The code is written for use in Matlab. The code to create and interpret the model is gievn below.

Code for the function 'fenmodel.m' used to interpret the fence model.

```
function [a,b,c,d,u,T,y]=fenmodel(lengthfence,loadres,ground,u,T);
%function
[a,b,c,d,u,T,y]=fenmodel(lengthfence,loadres,ground,inputfunction);
%
% generates electric fence parameters using inputfunction string
%
% converts fence model to ss
%
% plots ss model for input function
%
%example usage:
%   fenmodel(1000,4500,'linear',pl628,T)')

  %generate fence parameters
  fence=genfen(lengthfence,loadres,ground);
  disp('      Rl          Rg        Rw          C          L')
  disp(fence)

  %convert fence model to ss
  [a,b,c,d]=tf2ss([fence(1)],[fence(4)*fence(1)*fence(5),(fence(4)*f
ence(1)*fence(3)+fence(5)+fence(4)*fence(2)*fence(1)),(fence(3)+fenc
e(1)+fence(2))]);

  %find and evaluate desired input function
%  [u,T]=eval(inputfunction);

  %simulate input in model
  [y,x]=lsim(a,b,c,d,u,T);

  %give energy estimations:
  %E=V*V/(R*s)
  E=0;
  for i=1:length(T)-1,
    E=E+(y(i)*y(i)*(T(i+1)-T(i)))/(loadres);
  end  %for
  disp('Energy   (Joules)');
  disp(E);

  %plot input and output of model
  figure;
  plot(T,u,'y-',T,y,'r-');
```

```
  title(sprintf('Electric Fence Response %i
km',lengthfence/1000),'FontUnderline','on');
  YLabel('Volts (V)');
  XLabel('Time (s)');
  yax=get(gca,'ylim');
  xax=get(gca,'xlim');
  H=text(xax(2)*0.7,0.9*yax(2),'Input');
  set(H,'color',[1,1,0]);
  H=text(xax(2)*0.7,0.85*yax(2),'Output');
  set(H,'color',[1,0,0]);
  text(xax(2)*0.7,0.8*yax(2),sprintf('Rl = %i r',fence(1)));
  text(xax(2)*0.7,0.75*yax(2),sprintf('Rg = %i r',fence(2)));
  text(xax(2)*0.7,0.7*yax(2),sprintf('Rw = %i r',fence(3)));
  text(xax(2)*0.7,0.65*yax(2),sprintf('C = %1.2e F',fence(4)));
  text(xax(2)*0.7,0.6*yax(2),sprintf('L = %1.2e H',fence(5)));
  text(xax(2)*0.7,0.5*yax(2),sprintf('E = %1.4f J',E));
end
```

Code for the Matlab program 'Genfen.m' used to create the electric fence model:

```
function [fence]=genfen(lengthfence,loadres,ground);
% generates Rl, Rg, Rw, L, C from a given length of fence
% assumes L = 4uH/m
%   C = 8pF/m
%   Rw = 44r/km (2.5mm HT with skin effect)
%   Rg = 100r/km if 'linear', or 1k if 'fixed', else val ground
%   Rl = loadres
[fence]=
[loadres,100*lengthfence/1000,44*lengthfence/1000,0.000000000008*len
gthfence,0.000004*lengthfence];
if isstr(ground)
  if ground=='linear'
    fence(2)=100*lengthfence/1000;
  elseif ground=='fixed'
    fence(2)=1000;
  end %if
else
  fence(2)=ground;
end
```

Appendix 2
The Energiser Model

The energiser model is used to model a typical electric fence energiser and fence. For the fence the nominal Π circuit is used. Two versions of the model have been created. the first is just the energiser with a resistive load, the second includes the fence model as a load. The model is presented in chapter 3. The code is written for use in Matlab. The code to create and interpret the model is gievn below.

Code for the Matlab program 'Go.m' used by the energiser/energiser-fence model.

```
function [tott,toty,totl]=go(fun,t0,tf,input);
%function [t,y,l]=go(fun,t0,tf,input);

time=t0:0.000001:tf;

%set screen display divider:
divider=10;
count=0;

%display number of screen steps
num_steps=(length(time)-1)/divider

%get initial y values for first iteration
y(1,:)=input;

for i=1:length(time)-1
   count=count+1;
   [t,y]=ode45(fun,time(i),time(i+1),y(size(y,1),:),1e-7,0);
   tott=[tott; t(length(t))];
   toty=[toty; y(size(y,1),:)];

   [m n]=size(y);
   L2=induct3(y(m,5)+y(m,6)*7.7);
   totl=[totl; L2];
   %display time every 10th iteration:
   if count==divider
      disp(time(i+1))
      count=0;
      save temp tott toty totl;
   end;
% disp([time(i+1) induct3(y(m,5)-y(m,6)*8)])
end;

%plot results
figure
plot(tott,toty(:,length(input)));
title(['Output at time ',num2str(time(i+1))]);
end;
```

Code for the Matlab program 'Simfen5.m' used to model the energiser with a resistive load. The function is a series of first order ordinary differential equations called from the Matlab function 'ode.m'.

```
function xdot=simfen5(t,x)

Is=2*1.9e-9;        %2 diodes in parallel
Vt=2*42.8e-3;       %2 diodes in each arm
Ce=40e-6;
Ls=25e-6;
Cs=7e-6;
Rt1=0.085;
Lt1=1e-6;
Lt2=1e-5;
N=8.667;
Rm=4;
Lm=induct3(x(5)+N*x(6));
Rt2=7.5;
Rl=1000;

xdot(1)=(-x(2)+Is*(exp(-x(1)/Vt)-1))/Ce;
xdot(2)=(x(1)-x(3))/Ls;
xdot(3)=(x(2)-x(4))/Cs;
xdot(4)=(x(3)-x(4)*(Rt1+Rm)+x(5)*Rm)/Lt1;
xdot(5)=(Rm*(x(4)-x(5))/Lm)-(N*N*Rm*(x(4)-x(5))-
N*x(6)*(Rt2+Rl))/Lt2;
xdot(6)=(N*Rm*(x(4)-x(5))-x(6)*(Rt2+Rl))/Lt2;

end;
```

Code for the Matlab program 'Simfen4.m' used to model the energiser and fence.

```
function xdot=simfen4(t,x)

Is=2*1.9e-9;        %2 diodes in parallel
Vt=2*42.8e-3;       %2 diodes in each arm
Ce=40e-6;
Ls=25e-6;
Cs=7e-6;
Rt1=0.02;
Lt1=160e-9;
Lt2=1000e-6;
N=7.7;
Rm=4;
%30/(x(5)+7.7*x(6)+10);
Lm=induct3(x(5)+N*x(6))/3;

%Lm=0.000254;
Rt2=7.5;
C1=20e-9;
Rw=440;
Rg=100;
Lw=40e-3;
Rl=5e2;
C2=40e-9;
```

```
xdot(1)=(-x(2)+Is*exp((-x(1)/Vt)-1))/Ce;
xdot(2)=(x(1)-x(3))/Ls;
xdot(3)=(x(2)-x(4))/Cs;
xdot(4)=(x(3)-x(4)*(Rt1+Rm)+x(5)*Rm)/Lt1;
xdot(5)=((Lt2-N*N*Lm)/Lt2)*((x(4)-
x(5))/Lm)*Rm+(N/Lt2)*(x(6)*Rt2+x(7));
xdot(6)=(1/(Lt2-N*N*Lm))*(N*Lm*xdot(5)-x(6)*Rt2-x(7));
xdot(7)=(x(6)-x(8))/C1;
xdot(8)=(x(7)-x(9)-x(8)*Rw-x(8)*Rg)/Lw;
xdot(9)=(x(8)-x(9)/Rl)/C2;

end;
```

Code for the Matlab program 'Induct3.m' used by 'Simfen.m' to determine the inductance of the energiser output transformer.

```
function L=induct3(Inew);
%inductance of 5800 transformer
%area=0.0254*0.018;      %x-sect area of 5800 lamination

Magforce=[0 20 30 40 50 60 70 80 90 100 200 400 600 1000 2000 4000
10000 100000];
mu=[0 .002 .0033 .005 .0064 .008 .009 .0092 .0092 .009 .0062 .0035
.0024 .0016 .0009 .0006 .0004 2.2e-7];

H=45*Inew/0.1524; %H for 5800 transformer

if (H>100000)|(H<-100000)
  L=0.000001337;
else
  if H>0
  L=(0.92583*0.92)/((0.1524/interp1(Magforce,mu,H,'linear'))+1404.25
3); %10 thou gap
  else
    L=(0.92583*0.92)/((0.1524/interp1(Magforce,mu,-
H,'linear'))+1404.253); %10 thou gap
  end
end;
if L<0.000001337
  L=0.000001337;
end;

end;
```

Appendix 3
Measurement Circuit Model

Matlab code for the model to view slave unit transmission rates compared to peak fence voltage.

```
function numpulses=txpulsim(c,d,n)

%c=charging constant
%n=number of simulation steps
%pulse rate=numpulses/n

for Vc1=1:100
      numpulses(Vc1)=0;
      Vc2(Vc1)=2;
      for i=1:n,
            %charge C2
            Vc2(Vc1)=(Vc1-Vc2(Vc1))*c+Vc2(Vc1);

            %discharge and pulse
            if Vc2(Vc1)>2.5
                  Vc2(Vc1)=Vc2(Vc1)-0.5;
                  numpulses(Vc1)=numpulses(Vc1)+1;
            end; %if
      end %for
end;  %for

figure;
plot(numpulses/n);
title(sprintf('Tx Reply Pulse Sim %f %f %i',c,d,n));
xlabel('Voltage');
ylabel('Pulse Rate');
grid on;
figure;
plot(Vc2);
end %funct
```

Appendix 4
Transmission Circuit Model

The model for the transmission circuit is presented in Chapter 4. It is used to determine the amplitude of the slave unit transmissions. The model includes two functions, the first interprets the model, and the second is the differential equations that define the model.

Code for the Matlab program 'Runsimtx.m' used to interpret the transmission circuit model.

```
function [T,Y]=runsimtx(t0,tf,La1,Ra1,La2,Ra2,Ca1,Ca2,Ra3);
%function [T,Y]=runsimtx(t0,tf,La1,Ra1,La2,Ra2,Ca1,Ca2,Ra3);
global L1 R1 L2 R2 C1 C2 R3;
L1=La1;
R1=Ra1;
L2=La2;
R2=Ra2;
C1=Ca1;
C2=Ca2;
R3=Ra3;

[T,Y]=ode45('simtx',t0,tf,[0,0,0,0]);
T=T/0.001831;
figure;
subplot(4,1,1)
plot(T,Y(:,1));
title('Slave Transmission Simulation','FontUnderline','on');
ylabel('i1');
grid on;
subplot(4,1,2)
plot(T,Y(:,2),'r');
ylabel('Vc1');
grid on;
subplot(4,1,3)
plot(T,Y(:,3),'g');
ylabel('Vc2');
grid on;
subplot(4,1,4)
plot(T,Y(:,4),'c');
ylabel('i2');
xlabel(sprintf('L1:%0.3fH %2.0fr  L2:%0.3fH %1.1fr  C1:%1.1e
C2:%1.1e  R:%1.2fr',L1,R1,L2,R2,C1,C2,R3));
grid on;

end %funct
```

Code for the Matlab program 'Simtx.m' which defines the transmission circuit model.

```
function xdot=simtx(t,x)

global L1 R1 L2 R2 C1 C2 R3;


if ( (floor(t/0.001831))/2 - floor( (floor(t/0.001831))/2 ) ) ~= 0,
      V1=0.6;
else
      V1=3;
end %if

if t<0
      V1=0;
end
if t>0.058
      V1=0;
end

xdot(1)=( V1-x(1)*R1-x(2)-x(3) )/L1;
xdot(2)=(x(1))/C1;
xdot(3)=(x(1)-x(4))/C2;
xdot(4)=(x(3)-R2*x(4)-R3*x(4))/L2;

end;  %function
```

Appendix 5
Slave Unit Software

```
;SLAVE.ASM
;Code for slave unit for EFMS system.
;ver 1.04
;written by Paul Adamson
;last edited 23/9/96
;
;uP goes into idle mode when not in use
;
;T0 controls tx start 16 bit
;T1 measures pulse height (length) 8 bit
;INT1 signals pulse arrival

;P1.0-P1.1 comparator inputs unused
;P1.2 unused
;P1.3 turn on ID switch (output)
;P1.4-P1.7 ID data (input)

;P3.0 test: trigger
;P3.1-P3.2 unused
;P3.3 Pulse test active low!
;P3.4-P3.5 Burst Fire 3.4 fires first (output)
;P3.6 comparator out unused (input)
;P3.7 test: show output from A/D


PTEST       EQU P3.3
IDON        EQU P1.3 ;(Active low)

IDLE_BIT    EQU 01H  ;(Idle mode bit)
NUM_PULSES EQU 0DH  ;(Number of pulses to send in 1 timeslot
(13/15.16))
PUL_WIDTH   EQU 98H  ;(Timeslot Duration in machine cycles (151))
THRESHOLD   EQU 0FFH ;(Tx thresh (sum of peak voltages)) DON'T
CHANGE!
CODETIME    EQU 28H  ;(Number macs to offset tx timer for code)
SUM         EQU 32H  ;(address for peak summation)
PEAK        EQU 33H  ;(address to hold ADC result)
ADTIME      EQU 34H  ;(address to hold offset time to subtract from
timeslot 0)

;---------- RESET VECTOR -----------
  ORG   00
  AJMP  MAIN
;-------------- INTERRUPT VECTOR, IE0    ---------------
  ORG   03H       ;IE0
  RETI
;---------- INTERRUPT VECTOR Timer0 --------
  ORG   0BH     ;TF0
  AJMP  Timer0INT
  RETI
```

```
; -------------- INTERRUPT VECTOR, IE1     ----------------
  ORG   13H      ;IE1
  ;do every thing here!
  SETB  TR1      ;Turn timer 1 on
  AJMP  EXT1
  RETI
; -------------- INTERRUPT VECTOR, Timer1 ----------------
  ORG   1BH       ;TF1
  RETI
; -------------- INTERRUPT VECTOR, UART   ----------------
  ORG   23H       ;TI OR RI
  RETI



; -----IE1 body -----------
EXT1:
 ; -- Pulse has arrived (AND TIMER STARTED) -------
WAIT:
  JNB   P3.3,WAIT       ;Wait for pulse to go low
  CLR   TR1        ;Stop timer 1
    ; --get A/D measurement
  MOV   A,TL1      ;Get time P3.3 was low
  MOV   TL1,#00H  ;Reset timer1 ready for next pulse
;  MOV       TMOD,#01H   ;Reload timer configuration
  MOV   ADTIME,A  ;Store time to get measurement

  ADD   A,#2       ;Add jmp cont for decoding
  MOVC  A,@A+PC    ;get A/D result
  AJMP  CONT       ;Skip past lookup table
    ; --Lookup table
  DB    21H       ;0 in Acc

  DB    21H       ;1 in Acc
  DB    22H       ;2 in Acc ...
  DB    23H
  DB    24H
  DB    24H
  DB    25H
  DB    26H
  DB    26H
  DB    27H
  DB    28H

  DB    29H ;11
  DB    2AH
  DB    2AH
  DB    2BH
  DB    2CH
  DB    2DH
  DB    2EH
  DB    2FH
  DB    30H
  DB    31H

  DB    32H ;21
  DB    33H
  DB    34H
```

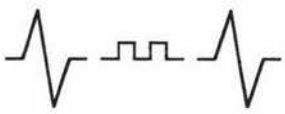
```
DB     35H
DB     36H
DB     37H
DB     38H
DB     39H
DB     3AH
DB     3CH

DB     3DH ;31
DB     3EH
DB     3FH
DB     40H
DB     42H
DB     43H
DB     44H
DB     46H
DB     47H
DB     49H

DB     4AH ;41
DB     4BH
DB     4DH
DB     4FH
DB     50H
DB     52H
DB     53H
DB     55H
DB     57H
DB     58H

DB     5AH ;51
DB     5CH
DB     5EH
DB     60H
DB     62H
DB     64H
DB     66H
DB     68H
DB     6AH
DB     6CH

DB     6EH ;61
DB     70H
DB     72H
DB     75H
DB     77H
DB     79H
DB     7CH
DB     7EH
DB     81H
DB     83H

DB     86H ;71
DB     89H
DB     8CH
DB     8EH
DB     91H
DB     94H
```

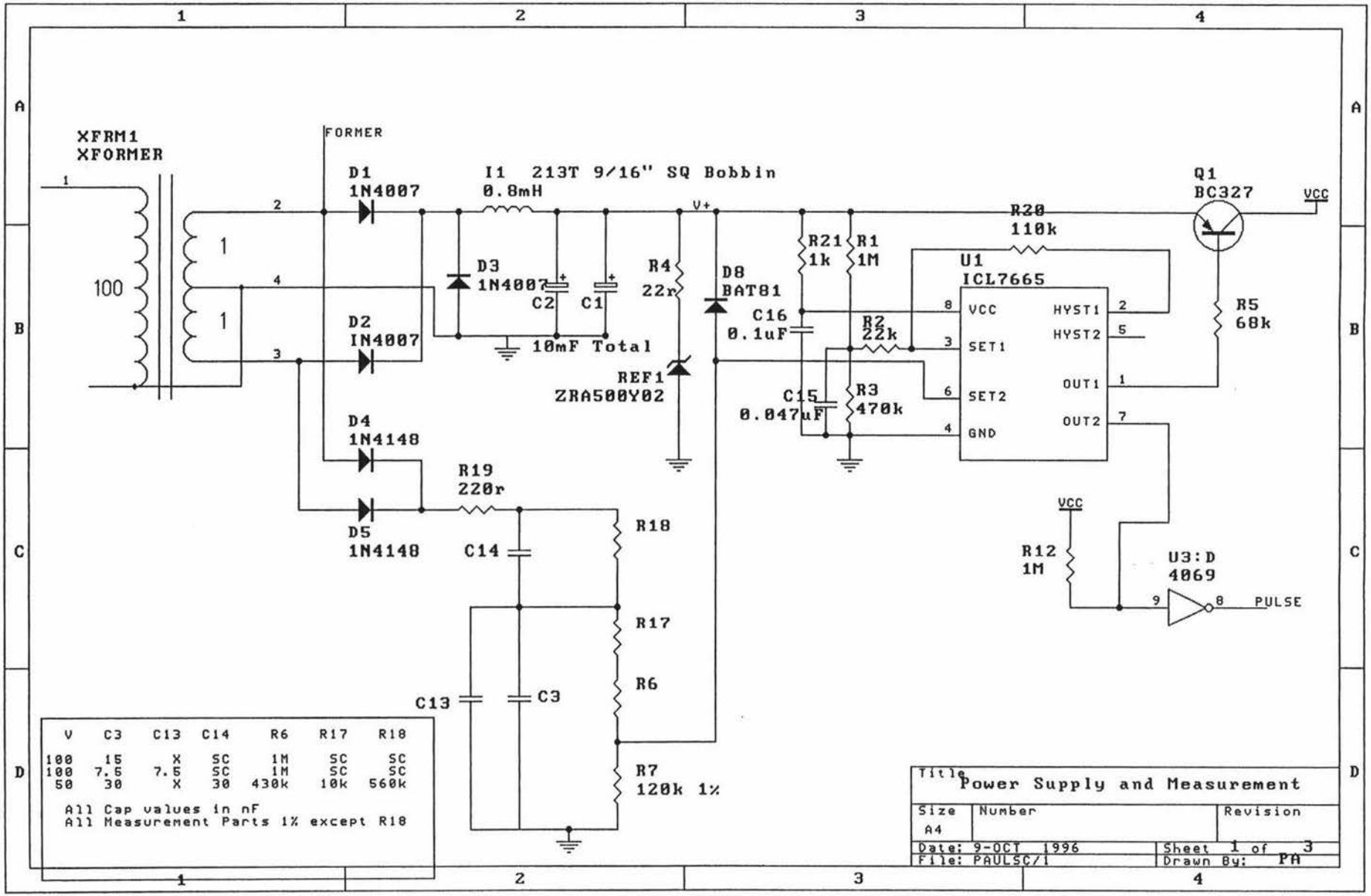
```
  DB    97H
  DB    9AH
  DB    9DH
  DB    0A0H

  DB    0A3H        ;81
  DB    0A7H
  DB    0AAH
  DB    0ADH
  DB    0B1H
  DB    0B4H
  DB    0B8H
  DB    0BCH
  DB    0C0H
  DB    0C3H

  DB    0C7H        ;91
  DB    0CBH
  DB    0CFH
  DB    0D4H
  DB    0D8H
  DB    0DCH
  DB    0E0H
  DB    0E5H
  DB    0EAH
  DB    0EEH

  DB    0F3H        ;101
  DB    0F8H
  DB    0FDH
  DB    0FFH
  DB    0FFH
  DB    0FFH
  DB    0FFH
  DB    0FFH
  DB    0FFH
  DB    0FFH

CONT:
  MOV   PEAK,A  ;Store A/D result

    ; --Test if tx req'd
  MOV   A,SUM
  ADD   A,PEAK          ;Add measurement to integrator
  MOV   SUM,A           ;Store sum result
  JNC   EXIT            ;IF TX NOT REQ'D THEN skip tx
; --TX REQ'D !!!!
  SUBB  A,#THRESHOLD    ;Subtract threshold (includes carry bit)
  MOV   SUM,A           ;Update value of integrator

  ;add const to A/D time taken for code (T0 offset)
  MOV   A,ADTIME        ;Load time A/D conversion took
  ADD   A,#CODETIME     ;Add time for code
  MOV   ADTIME,A        ;Store time used in timeslot 0

; -- get ID --
  CLR   IDON    ;Turn on ID switch
  MOV   A,P1    ;GET ID
```

```
  SETB  IDON    ;Turn off ID switch
  ANL   A,#0F0H ;Remove BOTTOM 4 bits (non-ID part of P1)
  SWAP  A
  INC   A       ;Set ID0 as ID1,..,ID15 as 16

; -- load ID to T0 --
  ;Multiply ID b0y timeslot duration to get wait duration (mac
cycles)
  MOV   B,#PUL_WIDTH   ;Timeslot duration (mac cycles)
  MUL   AB      ;wait duration from start of pulse to tx time
  ;subtract time already passed from tx wait time
  CLR   C       ;Clear carry bit for upcoming subtraction
  SUBB  A,ADTIME  ;Sub time already passed
  ;load timer etc...
  CPL   A       ;Complement for count up
  MOV   TL0,A   ;Load timer 0
  MOV   A,B     ;Get ready to complement top bits
  SUBB  A,#00H  ;subtract carry bit (borrow) from adtime sub if
required.
  CPL   A       ;Complement for count up
  MOV   TH0,A   ;Load top half of timer

;---------------------------------------
;---test routine -------
;  CLR   P3.0    ;TRIGGER FOR SCOPE
;  SETB  P3.0
;;--output a/d num to p3.7
;  RLC   A
;  MOV   P3.7,C  ;msb
;  RLC   A
;  MOV   P3.7,C
;  RLC   A
;  MOV   P3.7,C
;  RLC   A
;  MOV   P3.7,C
;  RLC   A
;  MOV   P3.7,C
;  RLC   A
;  MOV   P3.7,C
;  RLC   A
;  MOV   P3.7,C
;  RLC   A
;  MOV   P3.7,C
;  RLC   A
;  MOV   P3.7,C  ;LSB
;  RLC   A       ;Restore acculmulator
;  CLR   P3.0
;  SETB  P3.0
;  SETB  P3.7    ;Restore P3.7 (if req'd)
;---end of test routine
;-----------------------------

  SETB  EX1     ;Enable external int 1 (not disabled, but just in
case)
  SETB  EA      ;Enable interrupts
;  CLR      TF0   ;Remove previous timer 0 overflow
  SETB  TR0     ;Turn timer0 on
  SETB  ET0     ;Enable t0 int
EXIT:
;  -- Get to here after tx has been setup
```

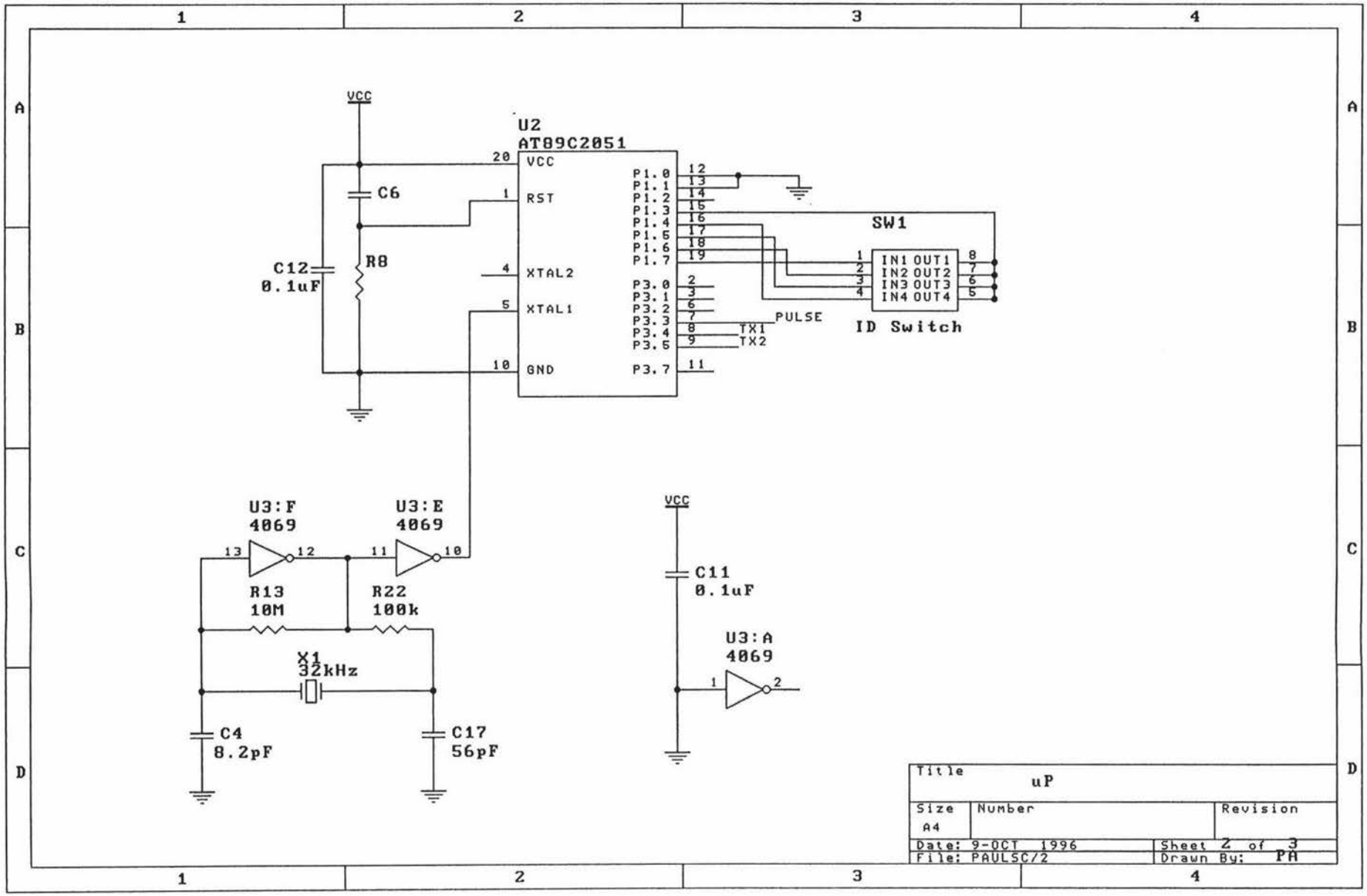
```
;  -- Go back to main to wait for another pulse or tx
   RETI



;----Timer 0 Body -------
 ;This is the part that controls the communication
 ;It currently works similar to a push pull driver, pulsing the
 ;filter at a rate of 273 pulses/sec.
 ;Actual looks similar to this:
 ;    ____       ____       ____
 ;___|  |_    _|  |_    _|  |_    __     ;top half    (+)
 ;       |__|       |__|       |__|       ;bottom half (-)
 ;
 ; 00++++0----0++++0----0++++0----00
 :
Timer0INT:
   CLR   ET0      ;Disable timer0 interrupt
   CLR TR0    ;Stop timer 0
   MOV   R1,#NUM_PULSES
   CLR   P3.4     ;1st transistion (before toggling)
   NOP            ;NOP to slow down to 273 Hz
   NOP
LOOP:
   NOP
   SETB  P3.4
   CLR   P3.5
   NOP
   NOP
   NOP
   SETB  P3.5
   CLR   P3.4
   DJNZ  R1,LOOP
   NOP
   SETB  P3.4
   RETI


MAIN:
;  -- Initialise
   MOV   P3,#0FFH         ;Set all P3 for input
   MOV   P1,#0FFH         ;Set all P1 as input
   MOV   SUM,#00H         ;Reset sum
;  -- Initialise TIMERs ------
   MOV   TMOD,#01H        ;Set for 16 bit timer
   CLR   TR1              ;Turn timer1 off
   MOV   TL1,#00H         ;Reset timer1 value
;  -- Init external int ------
   SETB  IT1      ;Set external int for falling edge trigger
   SETB  EX1      ;Enable external int 1
   SETB  EA

RUN:
;Wait for external interrupt and power down to idle mode
   ORL   PCON,#IDLE_BIT
   AJMP  RUN

;---- END OF SLAVE.ASM -----
```

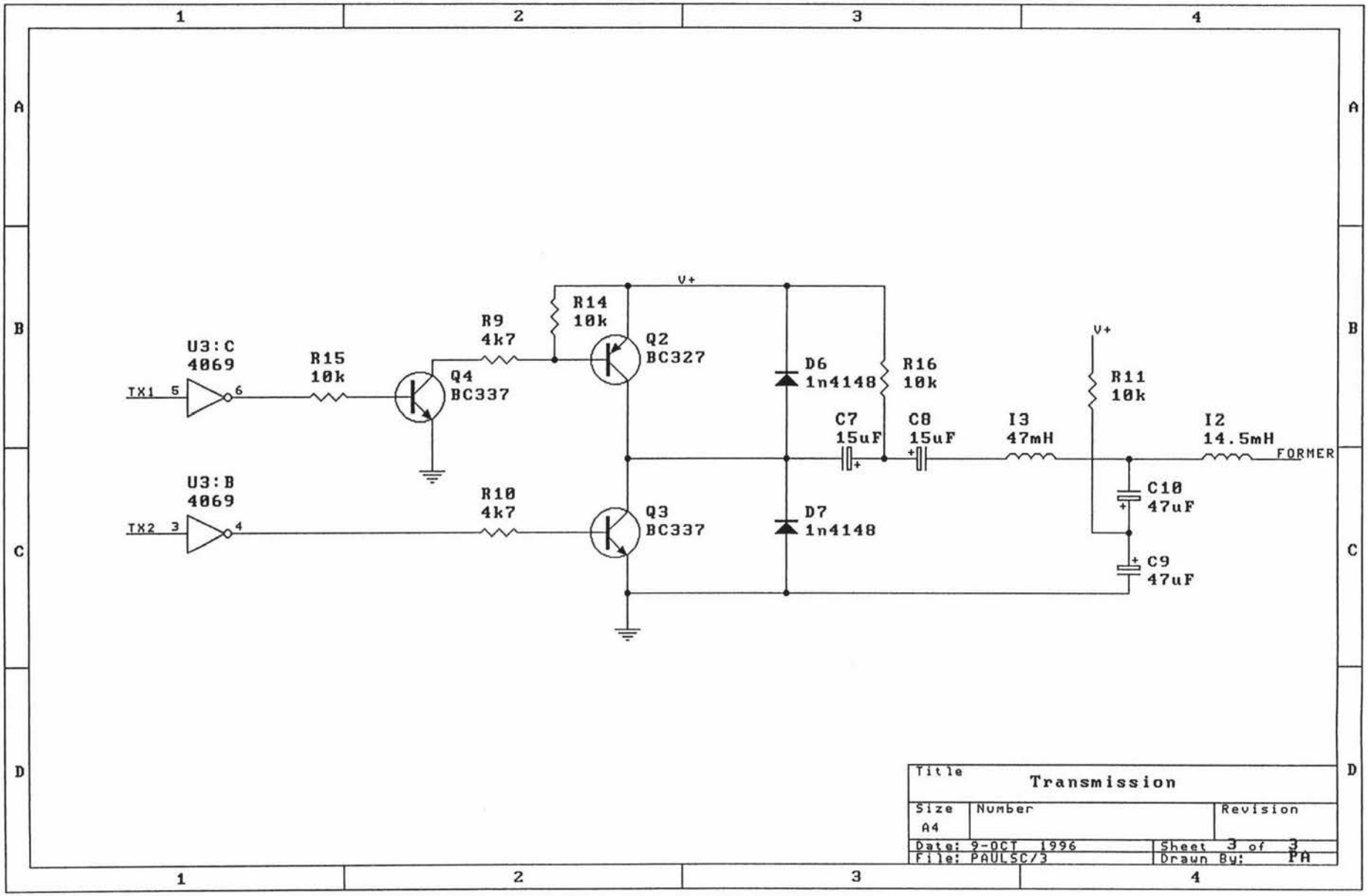
Appendix 6
Slave Unit Schematics



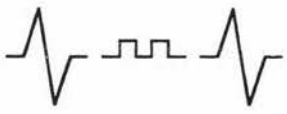


V	C3	C13	C14	R6	R17	R18
100	15	X	SC	1M	SC	SC
100	7.5	7.5	SC	1M	SC	SC
50	30	X	30	430k	10k	560k


VCC
U2
AT89C2051
20 VCC
1 RST
4 XTAL2
5 XTAL1
10 GND
P1.0 12
P1.1 13
P1.2 14
P1.3 15
P1.4 16
P1.5 17
P1.6 18
P1.7 19
P3.0 2
P3.1 3
P3.2 6
P3.3 7
P3.4 8
P3.5 9
P3.7 11
PULSE
TX1
TX2
C6
C12
0.1uF
R8
SW1
IN1 OUT1
IN2 OUT2
IN3 OUT3
IN4 OUT4
ID Switch
U3:F
4069
U3:E
4069
R13
10M
R22
100k
X1
32kHz
C4
8.2pF
C17
56pF
C11
0.1uF
U3:A
4069
Title
uP
Size
A4
Number
Revision
Date: 9-OCT 1996
Sheet 2 of 3
File: PAULSC/2
Drawn By: PH


1
2
3
4
A
B
C
D
U3:C
4069
TX1
5
6
R15
10k
R9
4k7
Q4
BC337
R14
10k
Q2
BC327
V+
D6
1n4148
R16
10k
C7
15uF
C8
15uF
I3
47mH
V+
R11
10k
I2
14.5mH
FORMER
C10
47uF
C9
47uF
U3:B
4069
TX2
3
4
R10
4k7
Q3
BC337
D7
1n4148
Title
Transmission
Size
A4
Number
Revision
Date: 9-OCT 1996
Sheet 3 of 3
File: PAULSC/3
Drawn By: PA

Appendix 7
Master Unit Software



Code used for Filter microprocessor:

```
;FILTER.ASM
;Code for filter side of master unit
;ver 1.00
;written by Paul Adamson
;Last edited: 2  September 96

;12MHz clk
;Standard filter clock = 12.82kHz (256.4Hz) (toggle every 39 mac)
;Fast filter clock = 0.5MHz
;uP TX to UI uP at half standard filter clock rate

;Timeslot duration 55.5ms
;10ms used to reset filter

;T0 CONTROLS TIMESLOTS duration
;IE0 STARTS TIMESLOTS
;R3 HOLDS TIMESLOT NUMBER
;R4 transmission counter
;R5 delay counter
;R6 AGC out port 1.4-1.6
;R7 maxreading window counter (num readings taken)

 ;outputs
ADCCLK      EQU P3.3
ADCDAT      EQU P3.5
ADCCS       EQU P3.4
FLTCLK      EQU P1.0    ;no int pullups for clock
EI          EQU P3.1
TX          EQU P3.7
WDOG        EQU P3.0
 ;inputs
ENER        EQU P3.2
 ;memory
TXLO        EQU 30H     ;transmission bits low
TXHI        EQU 31H     ;transmission bits high
TXHIBT      EQU 00H     ;set if in last eigth time slots
THROW       EQU 01H     ;HOLDS IF DATA SHOULD BE THROWN AWAY
THROW2      EQU 02H
LVOLT       EQU 32H     ;holds last A/D measurement
NVOLT       EQU 33H     ;holds max reading in timeslot0
COUNT       EQU 34H     ;Holds number measures since point grabbed
MAXRDG      EQU 36H     ;holds max window measurement voltage
MAXTS1        EQU 37H       ;holds lowest max voltage in current
timeslot
MAXTS2      EQU 38H
MAXTS3      EQU 39H     ;holds highest max voltage in timeslot
TCOUNT      EQU 3AH     ;TESTING COUNTER
MYVAR       EQU 3BH     ;TESTING VARIABLE
```

```
TEMP        EQU 3CH
AGCSET      EQU 3DH      ;AGC Status
 ;data
THRES       EQU 10H      ;ACCOUNTS FOR VARIATION IN NOISE VOLTAGE
AGCHI       EQU 50H      ;If nvolt>=thres dec AGC (1.5V)
AGCLO       EQU 20H      ;If nvolt<thres inc AGC  (0.6V)
WINLEN      EQU 05H    ;64H      ;AGC window length (100)
TSHI        EQU 26H      ;18th sec @ 12MHz
TSLO        EQU 0FCH
FCLKDR      EQU 64H      ;10ms @12MHz

;Reset Vector
  ORG  00
  AJMP MAIN
;Interrupt vector IE0
  ORG  03H
   ;Energiser pulse arrived
  AJMP PULSE
  RETI
;interrupt vector timer0
  ORG  0BH
  AJMP TSLOT
  RETI
;interrupt vector IE1
  ORG  13H
  RETI
;interrupt vector timer1
  ORG  1BH
  RETI
;interrupt vector UART
  ORG  23H
  RETI


;---------Fast Clock Subroutine----------
FCLOCK:
  MOV  R4,#FCLKDR
   ;fast clock duration=0.1*FASTCLKDUR ms
FCLK:
  SETB  FLTCLK ;1
  CLR   FLTCLK
  SETB  FLTCLK
  CLR   FLTCLK
  SETB  FLTCLK
  CLR   FLTCLK
  SETB  FLTCLK
  CLR   FLTCLK
  SETB  FLTCLK
  CLR   FLTCLK
  SETB  FLTCLK
  CLR   FLTCLK
  SETB  FLTCLK
  CLR   FLTCLK
  SETB  FLTCLK
  CLR   FLTCLK
  SETB  FLTCLK
  CLR   FLTCLK
  SETB  FLTCLK ;10
```

```
CLR   FLTCLK
SETB  FLTCLK
CLR   FLTCLK
SETB  FLTCLK
CLR   FLTCLK
SETB  FLTCLK
CLR   FLTCLK
SETB  FLTCLK
CLR   FLTCLK
SETB  FLTCLK
CLR   FLTCLK
SETB  FLTCLK
CLR   FLTCLK
SETB  FLTCLK
CLR   FLTCLK
SETB  FLTCLK
CLR   FLTCLK
SETB  FLTCLK
CLR   FLTCLK
SETB  FLTCLK
CLR   FLTCLK
SETB  FLTCLK ;20
CLR   FLTCLK
SETB  FLTCLK
CLR   FLTCLK
SETB  FLTCLK
CLR   FLTCLK
SETB  FLTCLK
CLR   FLTCLK
SETB  FLTCLK
CLR   FLTCLK
SETB  FLTCLK
CLR   FLTCLK
SETB  FLTCLK
CLR   FLTCLK
SETB  FLTCLK
CLR   FLTCLK
SETB  FLTCLK
CLR   FLTCLK
SETB  FLTCLK
CLR   FLTCLK
SETB  FLTCLK ;30
CLR   FLTCLK
SETB  FLTCLK
CLR   FLTCLK
SETB  FLTCLK
CLR   FLTCLK
SETB  FLTCLK
CLR   FLTCLK
SETB  FLTCLK
CLR   FLTCLK
SETB  FLTCLK
CLR   FLTCLK
SETB  FLTCLK
CLR   FLTCLK
SETB  FLTCLK
CLR   FLTCLK
SETB  FLTCLK
```

```
  CLR   FLTCLK
  SETB  FLTCLK ;40
  CLR   FLTCLK
  SETB  FLTCLK
  CLR   FLTCLK
  SETB  FLTCLK
  CLR   FLTCLK
  SETB  FLTCLK
  CLR   FLTCLK
  SETB  FLTCLK
  CLR   FLTCLK
  SETB  FLTCLK
  CLR   FLTCLK
  SETB  FLTCLK
  CLR   FLTCLK
  SETB  FLTCLK
  CLR   FLTCLK
  SETB  FLTCLK ;48
  CLR   FLTCLK
  DJNZ  R4,FAST
  AJMP  EFAST
FAST:
  AJMP  FCLK
EFAST:
  RET


;----------EXT 0 Body-----------------
PULSE:
  PUSH  ACC
  PUSH  PSW
  PUSH  05H             ;PUSH R5
  SETB  THROW
  SETB  THROW2

   ;Initialise T0 interrupt
  MOV   TL0,#TSLO       ;Load timer with timeslot duration
  MOV   TH0,#TSHI
  SETB  TR0             ;Turn timer0 on
  MOV   R3,#00H         ;Initialise timeslot number
  CLR   TXHIBT          ;move to low tx bits
  SETB  ET0             ;Enable timer0 interrupt
    ;empty filter
;  SETB  EI              ;earth inputs
  ACALL FCLOCK          ;Clear filter
;  CLR   EI
   ;tx last second results
  CLR   TX              ;Prep for falling edge
  NOP
  SETB  TX              ;Start bit (length of normal bit)
  MOV   R4,#08H         ;Load counter with number of high bits
  MOV   R5,#0CH
SB1:
  DJNZ  R5,SB1
  SETB  FLTCLK
  NOP
  MOV   R5,#12H
SB2:
```

```
  DJNZ  R5,SB2
  CLR   FLTCLK
  NOP
  MOV   R5,#12H
D1:
  DJNZ  R5,D1
  SETB  FLTCLK
  NOP
  MOV   R5,#12H
D2:
  DJNZ  R5,D2
  CLR   FLTCLK
  MOV   A,TXLO
  NOP
TXNLO:
  RRC   A               ;Send Msb first
  CPL   C
  MOV   TX,C            ;move bit to port
  MOV   R5,#10H         ;wait until toggle filter
D3:
  DJNZ  R5,D3
  SETB  FLTCLK
  MOV   R5,#12H
D4:
  DJNZ  R5,D4
  NOP
  CLR   FLTCLK
  MOV   R5,#12H
D5:
  DJNZ  R5,D5
  NOP
  SETB  FLTCLK
  MOV   R5,#12H
D6:
  DJNZ  R5,D6
  NOP
  CLR   FLTCLK
  DJNZ  R4,TXNLO  ;jump to next bit tx
  ;tx low bits
  MOV   R4,#08H         ;load counter with nuber of low bits
  MOV   A,TXHI          ;load acc with lo bits
TXNHI:
  RRC   A               ;Send Lsb next
  CPL   C
  MOV   TX,C            ;move bit to port
  NOP
  MOV   R5,#0DH
D7:
  DJNZ  R5,D7
  NOP
  SETB  FLTCLK
  MOV   R5,#12H
D8:
  DJNZ  R5,D8
  NOP
  CLR   FLTCLK
  MOV   R5,#12H
D9:
```

```
  DJNZ  R5,D9
  NOP
  SETB  FLTCLK
  MOV   R5,#12H
D10:
  DJNZ  R5,D10
  NOP
  CLR   FLTCLK
  NOP
  NOP
  DJNZ  R4,TXNHI
  SETB  TX                ;end bit for tx
  MOV   MAXTS1,#00H       ;Reset timeslot reading
  MOV   MAXTS2,#00H       ;Reset timeslot reading
  MOV   MAXTS3,#00H       ;Reset timeslot reading
;will be slight jitter on filter clock now as exit interrupt.

  POP   05H               ;POP R5
  POP   PSW
  POP   ACC
  RETI



;-------Timer 0 Body-------------
TSLOT:
  PUSH  ACC
  PUSH  PSW
  PUSH  05H
  SETB  THROW
  SETB  THROW2

    ;Update timer0
  MOV   TL0,#TSLO
  MOV   TH0,#TSHI
    ;empty filter
  SETB  EI                ;earth inputs
  ACALL FCLOCK            ;Clear filter
  CLR   EI
   ;check if timeslot zero
  CJNE  R3,#00H,NZEROT
  SJMP  NEXTTS            ;if zeroth timeslot exit

NZEROT:
   ;check if noise measurement timeslot (17)
  CJNE  R3,#11H,NNT
   ;timeslot17
   ;load noise threshold
  MOV   A,MAXTS3
  ADD   A,#THRES          ;add measurement threshold to noise voltage
  JC    NOVER             ;CARRY SET IF MAXTS3+THRES>255
  MOV   NVOLT,A
  SJMP  CONT:
NOVER:
  MOV   NVOLT,#0FFH
CONT:
  INC   R7      ;update window length
  AJMP  NEXTTS
```

```
NNT:
   ;load tx with result ready for tx to display uP
   ;for +ve maxts1>noise and maxts1>0.5*maxts3
  CLR   C
  MOV   A,MAXTS1          ;MAXTS1 must be > noise
  SUBB  A,NVOLT           ;Carry set if no pulse
  CPL   C                 ;carry set if pulse
  JNC   DTSTST
   ;test if maxts1>0.5maxts3
  MOV   A,MAXTS3
  CLR   C
  RRC   A                 ;divide ts3 by two
  CLR   C
  SUBB  A,MAXTS1          ;carry set if maxts1>0.5*maxts3
DTSTST:
   ;Want carry set by here if pulse in timeslot
  JB    TXHIBT,HIGHTX     ;if in last eight timeslots
  MOV   A,TXLO
  RRC   A
  MOV   TXLO,A
  AJMP  NEXTTS
HIGHTX:
  MOV   A,TXHI
  RRC   A
  MOV   TXHI,A
   ;test if end of timeslot 16 (ie AGC change req'd)
  CJNE  R3,#10,NEXTTS
   ;test if window=win length
  CJNE  R7,#WINLEN,NEXTTS
  MOV   R7,#00H           ;reset window
   ;***SET AGC CONTROL****
  CLR   C
  MOV   A,#AGCLO          ;test if noise readings not high enough
  SUBB  A,NVOLT           ;carry set if no adjust req
  JC    TESTHI
   ;inc AGC (carry not set)
  MOV   A,AGCSET          ;load current AGC status
  CJNE  A,#7H,DOINC:
  SJMP  NEXTTS
DOINC:
  INC   A                 ;inc AGC setting
  MOV   AGCSET,A          ;save new AGC setting
  SWAP  A                 ;move to AGC pins
  ANL   P1,#8FH           ;set AGC pins to low
  ORL   P1,A              ;move new AGC status to P1
  SJMP  NEXTTS
TESTHI:
  CLR   C
  MOV   A,NVOLT           ;test if noise readings too high
  SUBB  A,#AGCHI          ;carry set if no adjust req
  JC    NEXTTS
   ;dec agc
  MOV   A,AGCSET          ;load agc status
  JZ    NEXTTS            ;if AGC=0 then exit as no drop is possible
  DEC   A                 ;decrease AGC
  MOV   AGCSET,A          ;save new AGC setting
  SWAP  A
```

```
  ANL   P1,#8FH          ;set AGC pins to low
  ORL   P1,A             ;move new AGC status to P1

NEXTTS:
   ;Start of next timeslot
  INC   R3               ;inc timeslot number
   ;move to hi bits if req'd
  CJNE  R3,#09H,TS18
  SETB  TXHIBT
TS18:
  CJNE  R3,#12H,EXITT0
   ;timeslot18
  CLR   TR0              ;turn timer0 off
  CLR   TXHIBT           ;reset to low bits
  MOV   R3,#00H          ;reset timeslot number
EXITT0:
   ;GENERAL TIMESLOT STUFF
  MOV   MAXTS1,#00H      ;Reset timeslot voltage reading
  MOV   MAXTS2,#00H      ;Reset timeslot voltage reading
  MOV   MAXTS3,#00H      ;Reset timeslot voltage reading
  MOV   COUNT,#00H
  POP   05H
  POP   PSW
  POP   ACC
  RETI


;----------MAIN BODY---------------
MAIN:
   ;set timer0 and timer 1 for 16 bit timers
  MOV   TMOD,#11H
  SETB  TR0      ;start timer0
  CLR   ET0      ;disable timer0 interrupt
  SETB  ENER
   ;init it0
  SETB  IT0      ;falling edge
  SETB  EX0      ;enable external interrupt0
  SETB  EA       ;enable interrupts
   ;Initialise ADC
  SETB  ADCCS
  CLR   ADCCLK
  MOV   P1,#0BFH          ;init P1 (and AGC)
  MOV   AGCSET,#03H      ;init AGC status
  MOV   TXLO,#00H
  MOV   TXHI,#00H
  MOV   R7,#00H          ;Reset max window for AGC
  MOV   MAXRDG,#00H      ;Reset max window reading
  MOV   MAXTS1,#00H      ;Reset max timeslot reading
  MOV   MAXTS2,#00H      ;Reset max timeslot reading
  MOV   MAXTS3,#00H      ;Reset max timeslot reading
  MOV   COUNT,#00H
  MOV   NVOLT,#20H       ;Reset noise voltage with initial value
  CLR   EI
  SETB  TX

RLOOP:
   ;set filter clock
  SETB  FLTCLK
```

```
   ;get adc data bit:

  CLR   ADCCS
  SETB  WDOG
  MOV   C,ADCDAT        ;Bit 7
  RLC   A
  SETB  ADCCLK
  CLR   ADCCLK
  MOV   C,ADCDAT        ;bit 6
  RLC   A
  SETB  ADCCLK
  CLR   ADCCLK
  MOV   C,ADCDAT        ;Bit 5
  RLC   A
  SETB  ADCCLK
  CLR   ADCCLK
  MOV   C,ADCDAT        ;Bit 4
  RLC   A
  SETB  ADCCLK
  CLR   ADCCLK
  MOV   C,ADCDAT        ;Bit 3
  RLC   A
  SETB  ADCCLK
  CLR   ADCCLK
  MOV   C,ADCDAT        ;Bit 2
  RLC   A
  SETB  ADCCLK
  CLR   ADCCLK
  MOV   C,ADCDAT        ;Bit 1
  RLC   A
  SETB  ADCCLK
  CLR   ADCCLK
  MOV   C,ADCDAT        ;Bit 0
  RLC   A
  SETB  ADCCLK
  CLR   ADCCLK
  SETB  ADCCS
  MOV   LVOLT,A
  INC   COUNT
  CLR   WDOG
   ;toggle filter clock
  CLR   FLTCLK

;test code added here
   ;LOAD MAXTS3 WITH PEAK IN TIMESL
;  NOP
;  NOP
;  MOV   A,MAXTS3
;  CLR   C
;  SUBB  A,LVOLT          ;Carry set if lvolt>myvar
;  JNC   MACS2
;  MOV   MAXTS3,LVOLT
;  MOV   R5,#0EH
;  AJMP  D11
;MACS2:
;  NOP
;  NOP
;  MOV   R5,#0EH
```

```
;  AJMP  D11
;DUMP:
;  NOP
;  MOV   R5,#11H
;  SJMP  D11
;TEST CODE ENDS

  CLR   EA                ;DON'T LET TIMER 0 INTERRUPT IN MIDDLE OF
LOAD
  NOP
  JBC   THROW,DUMP
    ;test if last>max3
  JBC   THROW2,DUMP2
  CLR   C
  MOV   A,LVOLT
  SUBB  A,MAXTS3          ;Carry will be set if max3>lastVOLT
  JC    MACS12            ;Must jump away to get same code dur.
   ;new maximum value for timeslot
   ;test if count>15      (N.B carry already clear)
  MOV   A,COUNT
  SUBB  A,#0FH            ;Carry set if count<15
  JC    MACS4
   ;Update max values
  MOV   MAXTS1,MAXTS2
  MOV   MAXTS2,MAXTS3
CLOW:
  MOV   MAXTS3,LVOLT
  MOV   COUNT,#00H        ;Reset count
GOTVAL:
  SETB  EA                ;REENABLE INTERRUPT TIMER0
  MOV   R5,#05H

D11:
  DJNZ  R5,D11            ;delay for clock
  AJMP  RLOOP


  ;Delay to even up timings.
DUMP:
  NOP
  NOP
DUMP2:
  MOV   R5,#0DH
  SJMP  D11

MACS4:
  NOP
  NOP
  AJMP  CLOW

MACS12:
  NOP
  NOP
  NOP
  NOP
  NOP
  NOP
  NOP
```

```
  NOP
  NOP
  NOP
  AJMP  GOTVAL
```

Code used for User Interface Microprocessor.

```
Program Master;
{
use one counter for timing of tx
use another timer for 65ms tick
use one interrupt for tx input start bit
}

var lcd_special_g:boolean imported;

procedure imported lcd_driver;

{$L Master3 EFMSlcd}

const
    num_u_d_chars = 8;
    u_d_c_len = 66;
    u_d_chars : array[1..u_d_c_len] of byte =
                ($90,

                 %00000,
                 %00000,
                 %00000,
                 %00000,
                 %00000,
                 %11111,
                 %11111,
                 %00000,

                 %00000,
                 %00000,
                 %00000,
                 %00000,
                 %11111,
                 %11111,
                 %11111,
                 %00000,

                 %00000,
                 %00000,
                 %00000,
                 %11111,
                 %11111,
                 %11111,
                 %11111,
                 %00000,

                 %00000,
                 %00000,
                 %11111,
```

```
                %11111,
                %11111,
                %11111,
                %11111,
                %00000,

                %00000,
                %11111,
                %11111,
                %11111,
                %11111,
                %11111,
                %11111,
                %00000,

                %00100,
                %01110,
                %10001,
                %10001,  {Alarm Bell}
                %10001,
                %11111,
                %00010,
                %00000,

                %00000,
                %10001,
                %01010,  {Down Arrows}
                %00100,
                %00000,
                %10001,
                %01010,
                %00100,

                %00000,
                %00000,
                %01111,  {g moved down one row}
                %10001,
                %10001,
                %01111,
                %00001,
                %01110,

                $A0);

    timeslot_dur=98;     {256-timeslot_duration 6.33kbps @ 12MHz (1/2
filter clock speed)}
    curthou=14;          {location of thousands character in line}
    curhun=16;           {location of hundreds character in line}
    reg_thresh=15;       {minimum voltage for slave registration}
    winlen=15;           {check code before changing}
    keytimeout=153;      {10 sec @12MHz 16 bit (max 256) for keypad}
    enertimeout=153;     {10 sec @12MHz 16 bit (max 256)}
    fastpulsetime=15;    {1 sec @12MHz 16 bit}
    slowsiren=458;       {30 sec @12MHz 16 bit (max 65535)}
    num_slaves=16;       {maximum number of slave units in system}
    half_num_slaves=8;
    num_pages=4;         {number pages to hold slaves}
```

```
     beep_time=200;        {duration of buzzer for key_press * 16us
(max 256)}
    dereg_time=10;       {number times longer than normal beep}
     debounce_time=250;   {wait duration for key debounce * 32us (max
256)}
    {eeprom commands, NMC93C06 from National Semiconductor}
    write_comm=$40;
    read_comm=$80;
    write_enable_comm=$30;
    write_disable_comm=0;
    accuracy=0;          {address of accuracy field in eeprom}
    alarmloc=17;         {address of alaramack field in eeprom}
      enerack=18;             {address of fastpack and noenerack in
eeprom}

type
    slavetype = record
       voltage:byte;
       threshold:byte;
    end;      {record}

    pagetype = record
       line:array[1..4] of byte;
    end;

    slavestype= array[1..num_slaves] of slavetype;
    pagestype = array[1..num_pages] of pagetype;

var
   ctr,i:byte;              {general counter}
   slaves:slavestype idata; {array of slave data}
   pages:pagestype idata; {array of slave allocations to pages}
   responses:array[1..8] of array[1..half_num_slaves] of byte idata;
   win:byte;             {holds number of responses}
   currpage:byte;        {current page number (1-num_pages)}
   currline:byte;        {current line cursor is on (1-4)}
   cursor:byte;          {current character number in line (1-20)}
   siren:boolean;        {flag indicating if siren should be pulsing}
   setmode:boolean;      {flag indicating if user in setmode}
   noslaves,noenergiser:boolean; {flags for full screen displays}
   noenerack:boolean;    {set when noenergiser acknowledged}
   alarm:word;           {holds 16 flags of slaves in alarm}
   alarmack:word;        {holds 16 flags of alarms acknowledged}
   txcount:byte;         {holds tx number upto from filter}
   sirentime:word;       {holds time siren has been going}
   buttontime:byte;      {holds time since button press}
   enertime:byte;        {holds time since last tx}
   timing:boolean;       {keypad waiting for timeout}
   siren_switch:boolean at P3.4;   {turn on siren switch}
   fastpulse:boolean;    {set when energiser pulsing <1 sec}
   fastpack:boolean;     {set when fastpulsing acknowledged}
   tx:boolean at P3.2;   {recieve from filter tx I/O pin}
   bp:boolean at P1.4;   {buzzer I/O pin}
   test1:boolean at P1.6;
   test2:boolean at P1.7;
   ackalarmkey:boolean at P1.0; {keypad}
   setthreskey:boolean at P1.1;
   thousandskey:boolean at P1.2;
```

```
   hundredskey:boolean at P1.3;
   TEST:BOOLEAN AT P0.0;
   watchdog:boolean at P3.4;
   cs_eeprom:boolean at P0.4;
   sk_eeprom:boolean at P0.5;
   di_eeprom:boolean at P0.6;
   do_eeprom:boolean at P0.7;
   data_eeprom:word;     {data field for eeprom}
     updatereq:boolean;     {set if interrupt requires an updateof
voltages}
   dispagereq:boolean;
   acc:byte;             {holds accuracy setting}
   timeout:boolean;      {set when timeout on start up}
   startmode:byte;       {holds if in a startup mode}
   temp2,temp3:byte;       {test vars}
   temp1,temp4:word;

{-------------------------------------------------------------}

procedure keypress;
begin
  {beep}
  bp:=false;
  for ctr:=1 to beep_time do;
  for ctr:=1 to beep_time do;
  bp:=true;
  {wait until no key is pressed}
  repeat until (P1 and $0F)=$0F;
  {wait for debounce (four cycles of counter)}
  for ctr:=1 to debounce_time do;
  for ctr:=1 to debounce_time do;
  for ctr:=1 to debounce_time do;
  for ctr:=1 to debounce_time do;
end;

{-------------------------------------------------------------}

procedure eeprom(instruction:byte;address:byte);
var
    i:byte;
    command_address:byte;
begin
    command_address:=instruction + address;
    sk_eeprom:=false;
    cs_eeprom:=true;
    di_eeprom:=false;
    sk_eeprom:=true;
    sk_eeprom:=false;
    di_eeprom:=true;     {move start bit to eeprom}
    sk_eeprom:=true;
    for i:=1 to 8 do     {move command to eeprom}
      begin
      sk_eeprom:=false;
      command_address:=command_address shl 1;
      di_eeprom:=psw.7;
      sk_eeprom:=true;
      end;
```

```
    if instruction=write_comm then      {move out data}
      begin
      for i:=1 to 16 do
        begin
        sk_eeprom:=false;
        data_eeprom:=data_eeprom shl 1;
        di_eeprom:=psw.7;
        sk_eeprom:=true;
        end;
      sk_eeprom:=false;
      cs_eeprom:=false;
      sk_eeprom:=true;
      cs_eeprom:=true;
      do_eeprom:=true;
      repeat until do_eeprom;
      cs_eeprom:=false;
      cs_eeprom:=true;
      end;

    if instruction=read_comm then
      begin
      do_eeprom:=true;
      data_eeprom:=0;
      for i:=1 to 16 do {read data in}
        begin
        sk_eeprom:=false;
        sk_eeprom:=true;
        data_eeprom:=data_eeprom shl 1;
        if do_eeprom then inc(data_eeprom);
        end;
      end;

    cs_eeprom:=false;
end;

{-------------------------------------------------------------}

procedure movecursor(line,character:byte);
{moves the cursor to a specified location on screen}
begin
   case line of
     1:write(display,chr($A0+character-1));
     2:write(display,chr($D0+character-1));
     3:write(display,chr($B4+character-1));
     4:write(display,chr($E4+character-1));
   end
end;

{-------------------------------------------------------------}

procedure setblink(on_off:boolean);
{makes cursor visible and character blink}
begin
   {send blink to screen}
            if    on_off    then    write(display,chr($8D))    else
write(display,chr($8A));
end;
```

```
{-------------------------------------------------------------}

procedure setcursor(on_off:boolean);
{makes cursor visible}
begin
   {send blink to screen}
            if    on_off    then    write(display,chr($8C))    else
write(display,chr($8A));
end;

{-------------------------------------------------------------}

procedure getaccuracy;

begin
  {setup timeout timer}
  buttontime:=0;
  timeout:=false;
  {write display format}
  write(display,chr($80),'Select EFMS Accuracy');
  write(display,chr($B5),'Move Down = Change');
  write(display,chr($E5),'Set Thres = Select');
  write(display,chr($D7),'LOW');
  acc:=3;
  setcursor(true);
  movecursor(2,8);       {move cursor to Low}
  repeat
    watchdog:=false;
    watchdog:=true;
    if hundredskey=false then
      begin
      buttontime:=0;
      acc:=acc+3;
      if acc=15 then acc:=3;
      case acc of
         3:write(display,'LOW   ');
         6:write(display,'MEDIUM');
         9:write(display,'HIGH  ');
         12:write(display,'SUPER ');
      end; {case}
      movecursor(2,8);
      keypress;
      end;
  {add timout to following clause}
   until  (not(setthreskey)  or  timeout);          {wait  until  set
threshold key pressed or timeout}
  if setthreskey then
    begin
    {read old data}
    eeprom(read_comm,accuracy);
    acc:=lo(data_eeprom);
    end
  else
    begin
    keypress;
    {write new data}
    eeprom(write_enable_comm,0);
    data_eeprom:=acc;
```

```
    eeprom(write_comm,accuracy);
    eeprom(write_disable_comm,0);
  end;
  setcursor(false);
  timeout:=true;
end;

{-------------------------------------------------------}

procedure getautoset;

begin
  buttontime:=0;
  timeout:=false;
  write(display,chr($80),'Press Set Threshold');
  write(display,chr($D3),'To Re-register');
  write(display,chr($B8),'Slave Units');
  write(display,chr($E5),'(Ack Alarm=Cancel)');
  repeat
    watchdog:=false;
    watchdog:=true;
  until (timeout or not(setthreskey)or not(ackalarmkey));
  if setthreskey then
    begin
      startmode:=((acc*2) div 3)+12;     {turn off alarm until full
reading obtained + 1 min}
     {2*15 sec slots added to account for power up time for slave
units}
    {read in thresholds for slaves}
    for ctr:=1 to num_slaves do
      begin
      eeprom(read_comm,ctr);
      slaves[ctr].threshold:=lo(data_eeprom);
      end;
    updatereq:=true;     {update screen after load}
    end
  else
    begin
    startmode:=8;        {register slaves at first update}
    acc:=6;      {will be reset when required by update voltages}
     write(display,chr($80),chr($D3),'Searching For',chr($B8),'Slave
Units');
    write(display,chr($E8),'Pulses: ',(winlen*startmode-win):3);
    end;
  keypress;
  timeout:=true;
end;

{-------------------------------------------------------}

Procedure disnoenergiser;
{displays message no energiser detected}
begin
                                   write(display,chr($80),chr($D4),'No
Energiser',chr($BA),'Detected');
end;

{-------------------------------------------------------}
```

```
Procedure disnoslave;
{displays message no slave detected}
begin
                        write(display,chr($80),chr($D3),'No        Slave
Units',chr($B9),'Registered');
end;

{-------------------------------------------------------------}

Procedure disfastpulse;
{displays message energiser pulse rate too high}
begin
    write(display,chr($80),chr($D2),'Energiser  Pulse',chr($B7),'Rate
Too High');
end;

{-------------------------------------------------------------}

procedure DisplayPage;
{Refreshes the screen with slave data}
var
  linecount:byte;
  sln,slnv,slnt,slr:byte;
begin
   dispagereq:=false;
   if noslaves or fastpulse then
     begin
     if noenergiser then
                                    write(display,chr($80),chr($D5),'No
Energiser',chr($BB),'Detected')
       else if fastpulse then disfastpulse else disnoslave;
     end
   else
      begin
        write(display,chr($80));     {clears screen and sends cursor
home}
      for linecount:=1 to 4 do
         begin
         movecursor(linecount,1); {move cursor start of next line}
         if pages[currpage].line[linecount]<>0 then
            begin
               sln:=pages[currpage].line[linecount];    {gets current
slave number}
            slnv:=slaves[sln].voltage;
            slnt:=slaves[sln].threshold;
               write(display,'SL',sln:2,':',(slnv div 10):2,'.',slnv
mod 10,'kV (',slnt div 10,'.',slnt mod 10,')');
            if slnt>slnv then write(display,'A');
            end; {if}
         end; {for}
             if  (currpage<>1)or(pages[currpage+1].line[1]<>0)  then
write(display,chr($F7),#6);
      if (alarm>0) then write(display,chr($B3),#5);
      write(display,chr($C7));
      case acc of
      3:write(display,'L');
      6:write(display,'M');
```

```
        9:write(display,'H');
        12:write(display,'S');
        end; {case}
        {update pulse display}
        movecursor(2,20);
        case win of
          0,14:write(display,' ');
          1,13:write(display,'_');
          2,12:write(display,#0);
          3,11:write(display,#1);
          4,10:write(display,#2);
          5,9:write(display,#3);
          6,8:write(display,#4);
          7:write(display,chr($7F),chr($FF));
        end; {case}
        movecursor(currline,cursor);
      end; {else}
    eeprom(write_enable_comm,0);
    data_eeprom:=(ord(noenerack)*2)+ord(fastpack);
    eeprom(write_comm,enerack);
    eeprom(write_disable_comm,0);

end; {display page}

{-------------------------------------------------------------}

Procedure disnextpage;
begin
    inc(currpage);
    setcursor(true);
    {check new page is not empty}
         if   (currpage>num_pages)or(pages[currpage].line[1]=0)   then
currpage:=1;
    {check slave on current line}
    if (pages[currpage].line[currline]=0) then currline:=1;
    displaypage;
end; {proc}

{-------------------------------------------------------------}

Procedure AssignPages;
{assigns each registered slave to a page}
{and clears pages not used}
var
    slavecount,pagecount,linecount:byte;
begin
    pagecount:=0;
    {clear pages of all slaves}
    for pagecount:=1 to num_pages do
       for linecount:=1 to 4 do
          pages[pagecount].line[linecount]:=0;
    linecount:=0;
    pagecount:=1;
    for slavecount:=1 to num_slaves do
       begin
       if (slaves[slavecount].threshold<=99) then
          {slave is registered}
          begin
```

```
            inc(linecount);
            Pages[pagecount].line[linecount]:=slavecount;
            if linecount=4 then
               begin
               inc(pagecount);
               linecount:=0;
               end; {if}
         end; {if}
      end;
   if pages[1].line[1]=0 then noslaves:=true;
   if pages[currpage].line[currline]=0 then
      begin
      currpage:=1;
      currline:=1;
      end;
end;

{-------------------------------------------------------------}

procedure down;
begin
   setcursor(true);
   inc(currline);
         if   (currline>4)or(pages[currpage].line[currline]=0)   then
currline:=1;
   movecursor(currline,cursor);
end;

{-------------------------------------------------------------}

procedure hundreds;
{increse threshold by 100V}
var
   sln:byte;
begin
   {move cursor if req'd}
   cursor:=curhun;
   movecursor(currline,cursor);
   sln:=pages[currpage].line[currline];
   if (slaves[sln].threshold mod 10)<>9 then
      inc(slaves[sln].threshold)
   else
      slaves[sln].threshold:=slaves[sln].threshold-9;
   write(display,slaves[sln].threshold mod 10);
   movecursor(currline,cursor);
end;

{-------------------------------------------------------------}

procedure thousands;
{increase threshold by 1000V or deregister slave}
var
   sln:byte;    {holds current selected slave number}
begin
   cursor:=curthou;
   movecursor(currline,cursor);
   sln:=pages[currpage].line[currline];
   if slaves[sln].threshold<100 then
```

```
      begin
       slaves[sln].threshold:=slaves[sln].threshold+10;
       if (slaves[sln].threshold>99)and(slaves[sln].voltage>0) then
         slaves[sln].threshold:=slaves[sln].threshold-100;
      end
    else
       slaves[sln].threshold:=slaves[sln].threshold-100;
    {test to see if deregistered}
    if slaves[sln].threshold>=100 then
      write(display,'-.-')
    else
       begin
       write(display,slaves[sln].threshold div 10,'.');
       write(display,slaves[sln].threshold mod 10);
       end;
    movecursor(currline,cursor);
end;

{-------------------------------------------------------------}

procedure testalarm;
var
    count:byte;
    slt:byte;
    mask:word;
begin
    mask:=1;
    alarm:=0;
    for count:=1 to num_slaves do
       begin
       {set alarm field if req'd}
       slt:=slaves[count].threshold;
       if (slaves[count].voltage<slt)and(slt<100) then
         begin
         alarm:=(alarm or mask);
         end;
       {update alarmack if slave no longer in alarm}
       if (alarmack and mask)>(alarm and mask) then
         begin
         alarmack:=alarmack and (mask xor 65535);
         end;
       mask:=mask shl 1;
       end;
    {set siren if req'd}
     if alarm>alarmack then siren:=true else if (noenergiser=false)
and (fastpulse=false) then siren:=false;
    {write result to eeprom}
    eeprom(write_enable_comm,0);
    data_eeprom:=alarmack;
    eeprom(write_comm,alarmloc);
    eeprom(write_disable_comm,0);
end;

{-------------------------------------------------------------}

procedure setthreshold;
var
    sln:byte;
```

```
   j:byte;       {beep counter}
begin
   if setmode then
      begin
      {already in set threshold mode}
      setmode:=false;
      {give special beep for slave deregister}
      sln:=pages[currpage].line[currline];
      if slaves[sln].threshold>99 then
        begin
        bp:=false;
        for j:=1 to dereg_time do
         begin
          for ctr:=1 to beep_time do;
          for ctr:=1 to beep_time do;
         end;
        bp:=true;
        end;
      setcursor(true);
      {write results to eeprom}
      eeprom(write_enable_comm,0);
      data_eeprom:=slaves[sln].threshold;
      eeprom(write_comm,sln);
      eeprom(write_disable_comm,0);
      assignpages;
      if startmode=0 then testalarm;
      displaypage;
      end
   else
      begin
      {moving into setmode}
      setmode:=true;
      setblink(true);
   end; {if}
end;

{-------------------------------------------------------------}

procedure updatevoltages;
{also tests for alarm and removes ack if req'd}
var
   mask,sln,slv:byte;
   slr:word;
begin
   noslaves:=true;
   updatereq:=false;
   for sln:=1 to num_slaves do
      begin
        if (sln mod 2)=0 then mask:=$F0  {even slave num}
        else mask:=$0F;  {odd slave num}
        case acc of
              3:slr:=word(responses[7,(sln+1) div 2] and mask) +
word(responses[8,(sln+1) div 2] and mask);   {30 sec}
              6:slr:=word(responses[5,(sln+1) div 2] and mask) +
word(responses[6,(sln+1) div 2] and mask)
                  + word(responses[7,(sln+1) div 2] and mask) +
word(responses[8,(sln+1) div 2] and mask);   {60 sec}
```

```
                9:slr:=word(responses[3,(sln+1) div 2] and mask) +
word(responses[4,(sln+1) div 2] and mask)
                    + word(responses[5,(sln+1) div 2] and mask) +
word(responses[6,(sln+1) div 2] and mask)
                    + word(responses[7,(sln+1) div 2] and mask) +
word(responses[8,(sln+1) div 2] and mask);   {90 sec}
              12:slr:=word(responses[1,(sln+1) div 2] and mask) +
word(responses[2,(sln+1) div 2] and mask)
                    + word(responses[3,(sln+1) div 2] and mask) +
word(responses[4,(sln+1) div 2] and mask)
                    + word(responses[5,(sln+1) div 2] and mask) +
word(responses[6,(sln+1) div 2] and mask)
                    + word(responses[7,(sln+1) div 2] and mask) +
word(responses[8,(sln+1) div 2] and mask);   {120 sec}
        end; {case}

        if mask=$F0 then slr:=slr shr 4;
        slr:=slr*10;
        if (slr mod acc)>=((acc+1) div 2) then inc(slr);
        slr:=slr div acc;
        slaves[sln].voltage:=lo(slr);
        if slaves[sln].threshold<100 then noslaves:=false;

      end; {for}

   {update responses}
   for sln:=1 to half_num_slaves do
     begin
       responses[1,sln]:=responses[2,sln];
       responses[2,sln]:=responses[3,sln];
       responses[3,sln]:=responses[4,sln];
       responses[4,sln]:=responses[5,sln];
       responses[5,sln]:=responses[6,sln];
       responses[6,sln]:=responses[7,sln];
       responses[7,sln]:=responses[8,sln];
       responses[8,sln]:=0;
     end;

   win:=0;
   if (startmode>0) and (startmode<=8) then
     begin
     dec(startmode);
     if startmode=0 then
       begin
       {write results to eeprom}
       eeprom(write_enable_comm,0);
       for sln:=1 to num_slaves do
         begin
         {register slave if giving voltage}
                        if   (slaves[sln].voltage>=reg_thresh)   then
slaves[sln].threshold:=0
         else slaves[sln].threshold:=100;
         data_eeprom:=slaves[sln].threshold;
         eeprom(write_comm,sln);
         end; {for}
       eeprom(write_disable_comm,0);
       {read in desired accuracy}
       eeprom(read_comm,accuracy);
```

```
        acc:=lo(data_eeprom);
        end; {if startmode=0}
      end; {if startmode>0}
    if (startmode=0)or (startmode>8) then
      begin
      assignpages;
      if startmode>8 then dec(startmode) else testalarm;
      if startmode=8 then startmode:=0;
      displaypage;
    end; {if startmode=0}

    {beep}
    bp:=false;
    for ctr:=1 to beep_time do;
    for ctr:=1 to beep_time do;
    bp:=true;
end;

{-------------------------------------------------------------}

procedure ackalarm;
var
    mask:word;
    count:byte;
begin
    mask:=1;
    alarmack:=0;          {reset alarm acknowledgements}
    if (noenergiser=true) then noenerack:=true;
    if (fastpulse=true) then fastpack:=true;
    siren:=false; {stop siren}
    sirentime:=0;         {reset siren timer}
    for count:=1 to num_slaves do
       begin
       if (alarm and mask)>0 then
         begin
         alarmack:=alarmack or mask;
         end;
       mask:=mask shl 1;
       end;
    eeprom(write_enable_comm,0);
    data_eeprom:=alarmack;
    eeprom(write_comm,alarmloc);
    data_eeprom:=(ord(noenerack) *2)+ord(fastpack);
    eeprom(write_comm,enerack);
    eeprom(write_disable_comm,0);

end;

{-------------------------------------------------------------}

procedure interrupt int0;
{sets up timer0 for tx recieving slave tx bits}
begin
    {toggle port pin for debugging}
    test1:=false;
    if noenergiser=true then
       begin
       if (startmode<=8) and (startmode>0) then
```

```
                              write(display,chr($80),chr($D3),'Searching
For',chr($B8),'Slave Units',chr($E8),'Pulses: ')
        else dispagereq:=true;
       if (alarm<=alarmack) and (fastpulse=false) then siren:=false;
       end;
   noenergiser:=false;
   noenerack:=false;
   txcount:=0;
   if enertime<fastpulsetime then {energiser fast pulsing}
     begin
       if not(fastpack) then siren:=true; {set siren if alarm not
acknowledged}
     fastpulse:=true;
     dispagereq:=true;
     end
   else
     begin
     if fastpulse then
       begin
       fastpulse:=false;
       fastpack:=false;
              if (alarm<=alarmack) and (noenergiser=false) then
siren:=false;
       dispagereq:=true;
       end;
     TL0:=timeslot_dur;    {load timer0 with duration of startbit}
     TR0:=true;   {start timer0}
     ET0:=true;   {enable timer0 int}
     EX0:=false;  {disble int0}
     test1:=true;
     IE0:=false;  {clear interrupt flag}
   end;
   enertime:=0; {reset energiser timer}

end;

{-------------------------------------------------------------}

procedure interrupt timer0;
{called for each slave after being setup by int0}
begin
   {toggle port pin for debugging}
   test2:=false;
   inc(txcount); {holds number of slave timeslot}
   {get tx from port}

   if tx=true then
     begin
       if (txcount mod 2)=0 then  {even slave, high nibble}
        begin
         responses[8,(txcount + 1) div 2]:=responses[8,(txcount + 1)
div 2]+16;
        end
       else  {odd slave}
         responses[8,(txcount + 1) div 2]:=responses[8,(txcount + 1)
div 2]+1;
     end; {if tx}
   if txcount=num_slaves then
```

```
      begin
      {reset timer0 and turn off}
      txcount:=0;        {reset tx counter}
      inc(win);          {update window}
      if win=winlen then updatereq:=true;
                         if    (startmode>=1)and(startmode<=8)    then
write(display,chr($F0),(winlen*startmode-win):3)      else
        begin
        write(display,chr($E3));
        case win of
          0,14:write(display,' ');
          1,13:write(display,'_');
          2,12:write(display,#0);
          3,11:write(display,#1);
          4,10:write(display,#2);
          5,9:write(display,#3);
          6,8:write(display,#4);
          7:write(display,chr($7F),chr($FF));
        end; {case}
      end;  {if autoset}
      {return cursor to location}
      case currline of
        1:write(display,chr($A0+cursor-1));
        2:write(display,chr($D0+cursor-1));
        3:write(display,chr($B4+cursor-1));
        4:write(display,chr($E4+cursor-1));
      end;
      IE0:=false;        {clear any external interrupts which are the
result of tx}
      EX0:=true;         {turn external interrupt 0 on}
      TR0:=false;        {stop timer0}
      ET0:=false;        {turn timer0 interrupt off}
      end; {if}
   {toggle pin for debugging}
   test2:=true;
   TF0:=false;  {clear timer overflow flag}
end;

{-------------------------------------------------------------}

procedure interrupt timer1;
{second counter for keypad timeout and minute counter for alarm}
{also counter for energiser timeout}
begin
   {handle keypad timeout for setthreshold}
   if (timing or not(timeout)) then inc(buttontime);
   if buttontime=keytimeout then
      begin
      if not(timeout) then timeout:=true
      else
        begin
        buttontime:=0;
        timing:=false;
        write(display,chr($8A))
      end;
      end;

   {handle energiser timeout}
```

```
   if timeout then inc(enertime);          {inc energiser time if not
in startup menu}
   if enertime=enertimeout then
      begin
      noenergiser:=true;
                                  write(display,chr($80),chr($D4),'No
Energiser',chr($BA),'Detected');
       if not(noenerack) then siren:=true;          {set alarm if no
energiser not acknowledged}
      end;
   {stop enertime from overflowing}
   if enertime=(enertimeout+1) then dec(enertime);

   {handle siren}
   if siren then
     begin
      inc(sirentime);
      if sirentime<slowsiren then
        begin
         {toggle buzzer/siren}
         if (sirentime mod 10)=1 then
          begin
           siren_switch:=true;
           bp:=false;
          end;
         if (sirentime mod 10)=6 then
          begin
           siren_switch:=false;
           bp:=true
          end
        end
      else
        begin
         if (sirentime mod 100)=1 then
          begin
           siren_switch:=true;
           bp:=false;
          end;
         if (sirentime mod 100)=6 then
          begin
           siren_switch:=false;
           bp:=true;
          end;
         {stop sirentime from overflowing}
         if sirentime=64577 then sirentime:=slowsiren;
      end;
     end  {if siren then}

   else
     begin
      {reset timercount}
        siren_switch:=false;
        bp:=true
   end;  {if siren else}
   TF1:=false;  {clear interrupt flag}
end; {proc}

{-------------------------------------------------------------}
```

```
procedure getkey;
begin
   {check if keypressed}
   if (P1 and $0F)<$0F then
     begin
      buttontime:=0;        {reset timercounter}
      timing:=true;         {turn keypad timer on}
      if not(P1.0) then ackalarm;
      if not(noenergiser or noslaves) then
        begin
        if not(P1.1) then setthreshold;
        if not(P1.2) then
           if setmode then thousands else disnextpage;
        if not(P1.3) then
           if setmode then hundreds else down;
        end;
      keypress;
     end;
end;

{-------------------------------------------------------------}

begin
   watchdog:=false;
   watchdog:=true;
   bp:=false;            {turn buzzer on}
   TEST:=TRUE;
   reset(display);
   for ctr:=1 to u_d_c_len do
        write(display,chr(u_d_chars[ctr]));
   lcd_special_g:=true;
   currpage:=1;
   currline:=1;
   cursor:=curthou;
   alarm:=0;
   fastpulse:=false;
   setmode:=false;
   updatereq:=false;     {no update of voltages on reset}
   dispagereq:=false;
   noslaves:=true;       {no slaves registered}
   noenergiser:=false;
   enertime:=0;
   txcount:=0;
   sirentime:=0;         {reset alarmtimer}
   buttontime:=0;        {reset keypad timeout}
   siren:=false;
   win:=0;
   eeprom(write_enable_comm,0); {initialise eeprom}
   eeprom(write_disable_comm,0);
   {read in alaramack}
   eeprom(read_comm,alarmloc);
   alarmack:=data_eeprom;
   eeprom(read_comm,enerack);
   data_eeprom:=data_eeprom shr 1;
   fastpack:=psw.7;
   data_eeprom:=data_eeprom shr 1;
   noenerack:=psw.7;
```

```
   bp:=true;              {turn buzzer off}

   {timer 0 setup}
    TMOD:=$12;              {timer 0 set for 8 bit autoreload, timer1
set for 16 bit}
   TR0:=false;            {turn timer0 off}
   ET0:=false;            {disable timer0 interrupt}
   TH0:=timeslot_dur;     {timer reload set}
   TL0:=timeslot_dur;     {reset timer0 for initial time}

   {timer1 setup}
   TR1:=true;             {turn timer1 on}
   ET1:=true;             {enable timer1 interrupt}
   EX0:=false;            {disable falling edge interrupt}
   EA:=true;              {enable interrupts}

   for ctr:=1 to num_slaves do
     begin
     slaves[ctr].voltage:=0;
     end;

   for ctr:=1 to half_num_slaves do
     for i:=1 to 8 do
       responses[i,ctr]:=0;

   getaccuracy;
   getautoset;

   enertime:=fastpulsetime+1;
   {external interrupt 0}
   IT0:=true;             {specify falling edge interrupt}
   IE0:=false;            {remove previouse interrupts}
   EX0:=true;             {enable external interrupt}


   {main loop}
repeat
  getkey;
  watchdog:=true;
  if setmode and not(timing) then setthreshold;
  if updatereq then updatevoltages;
  if dispagereq then displaypage;
  watchdog:=false;
until false;

end. {main}
```

Code for IO driver used by 'master3.pas'

```
$DEBUG
; LCD driver for an 8 character 2 or 4 row LCD connected to a
; Speedrite Electric Fence Monitoring System (EFMS) MAster Unit.
; 8 data bits on P2 , RS on P3.7 , RW on P3.6 and E on P3.5

; By the time we get here , the character to send is in A and the
device (1)
```

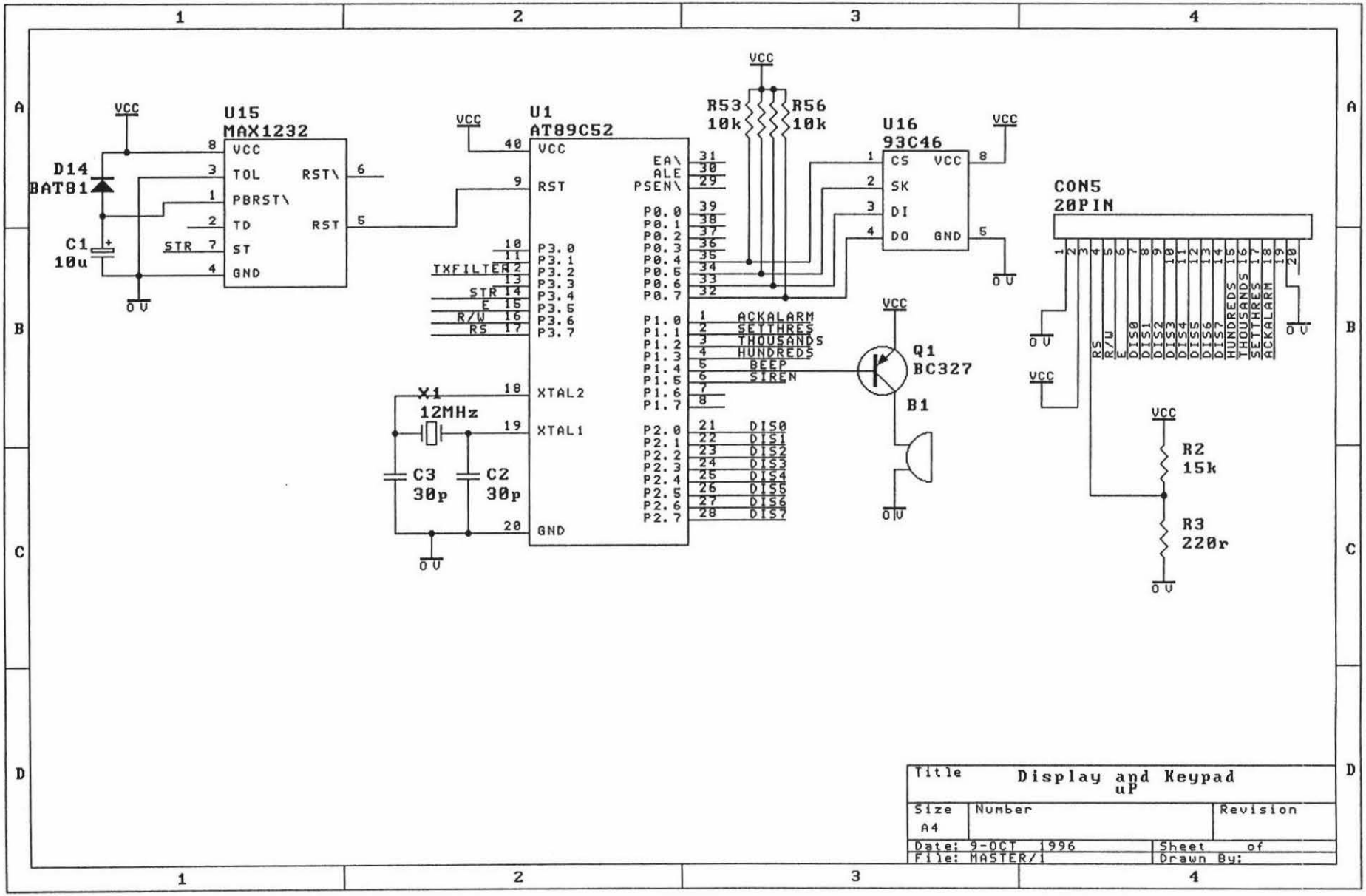
```
; will be in R2

; Version 0.73 - 27 July 1996

LIBRARY EFMSLCD

PUBLIC  LCD_DRIVER,?LCDRESET,?LCDWRITE,LCD_SPECIAL_G

?EFMSLCD?B  SEGMENT     BIT

RSEG  ?EFMSLCD?B

?LCD_DATA_ESC:  DBIT    1       ; Next character to be a data byte
?LCD_CTRL_ESC:   DBIT     1         ; Next character to be a control
byte
?LCD_EA_STATE:  DBIT     1        ; Used to save the state of the EA
bit
LCD_SPECIAL_G:  DBIT    1       ; Determines if user-defined #7 used
for "g"

?EFMSLCD?C  SEGMENT     CODE

RSEG  ?EFMSLCD?C

?LCDRESET:
        CLR     ?LCD_DATA_ESC
        CLR     ?LCD_CTRL_ESC
        CLR     LCD_SPECIAL_G
        ;
        MOV     R0,#8                   ; 8 bytes to send
        ;
LCD_RLP:MOV     A,R0                    ; Get index
         ADD      A,#DLY_TBL-DLY_GET-1     ; Calculate correct offset
for delay
        MOVC    A,@A+PC                 ; Get delay count
DLY_GET:LCALL   US100_DELAY
        MOV     A,R0                    ; Get index again
         ADD      A,#DAT_TBL-DAT_GET-1     ; Calculate correct offset
for data
        MOVC    A,@A+PC                 ; Get delay count
DAT_GET:LCALL   LCD_CTRL_WRITE
        DJNZ    R0,LCD_RLP
        ;
LCD_DRIVER:                                     ; Dummy entry point if
LCD_DRIVER called
        RET
        ;
;DLY_TBL:DB      1   ,17 ,1,1,1   ,1   ,41 ,150 ; Timings for 1/8 duty
cycle 250 kHz
DLY_TBL:DB       2   ,55 ,2,2,2   ,2   ,45 ,160 ; Pessimistic timings
for 1/16 duty
;DAT_TBL:DB      06H,0CH,1,8,30H,30H,30H,30H ; at 160 kHz - This row
for 1 line
DAT_TBL:DB      06H,0CH,1,8,38H,38H,38H,38H ; This row for 2 lines
        ;
US100_DELAY:
        MOV     R2,#50                  ; 50 DJNZs = 100 uS @ 12 MHz
        DJNZ    R2,$
```

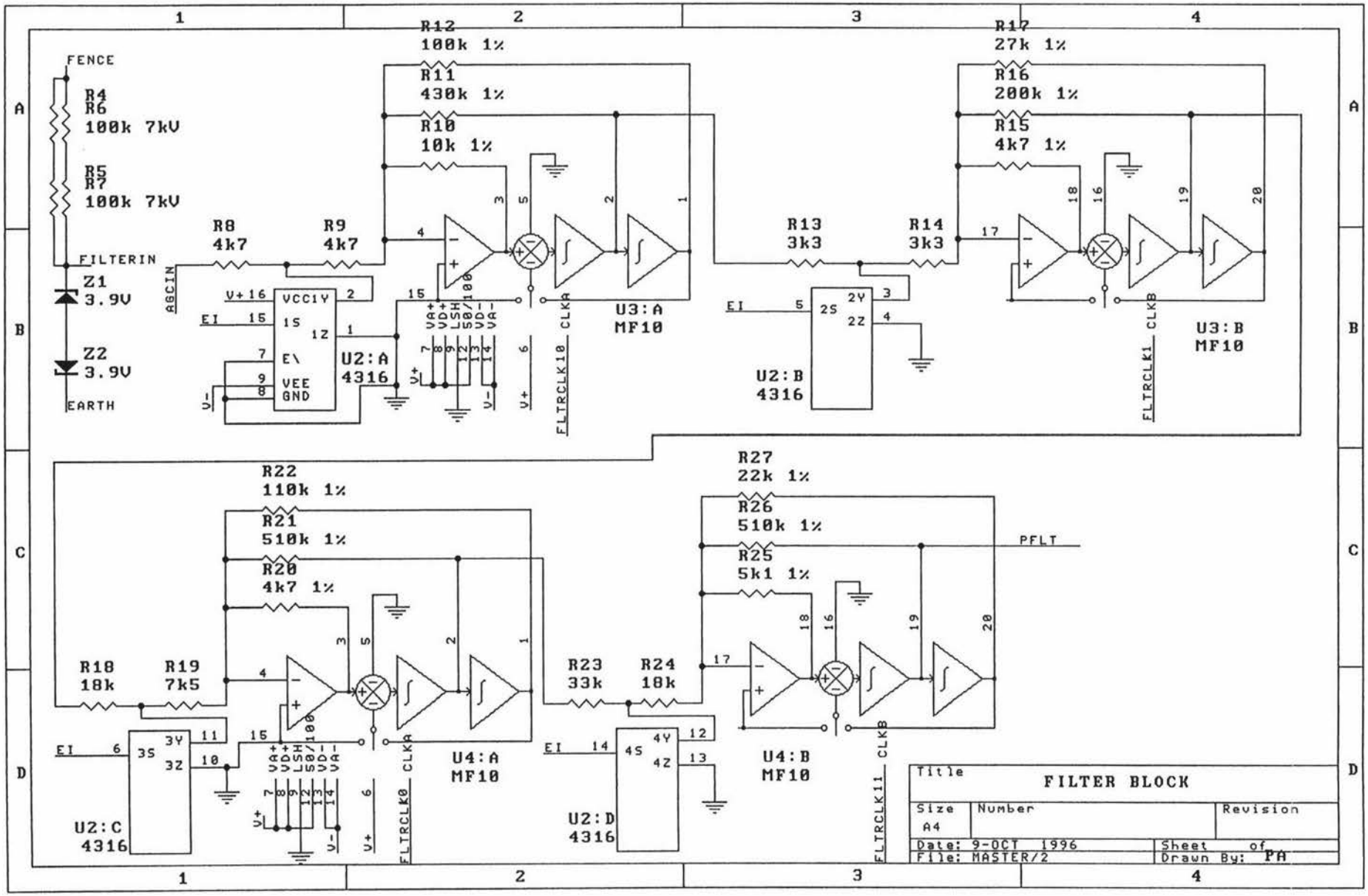
```
        DJNZ    ACC,US100_DELAY         ; Number of 100 uS periods
        RET
        ;
?LCDWRITE:
        LCALL   LCD_BUSY_WAIT
        ;
        JBC     ?LCD_CTRL_ESC,LCD_CTRL_WRITE ; Jump and clear bit if
set
        JBC     ?LCD_DATA_ESC,LCD_DATA_WRITE
        ;
LCDWRT2:CJNE    A,#0FFH,LCDWRT3         ; LCD control escape
        SETB    ?LCD_CTRL_ESC
        RET
        ;
LCDWRT3:CJNE    A,#07FH,LCDWRT4         ; LCD data escape
        SETB    ?LCD_DATA_ESC
        RET
        ;
LCDWRT4:JNB     ACC.7,LCD_DATA_WRITE    ; 0..126 = data bytes
        ADD     A,#30H                  ; Carry if in range D0-FE
        JNC     LCD_NR2                 ; Not a row 2 selector
        ;
        ADD     A,#10H                  ; Correct for row 2
        ;
LCD_NR2:ADD     A,#0B0H                 ; Subtract 50H
        JNB     ACC.7,LCD_CMD           ; Value from 60-7F
        ;
        ; Fall through to position LCD cursor ( 80-AF,C0-EF )
        ;
        ;
LCD_CTRL_WRITE:
        MOV     R2,#0                   ; RS value for control port
        SJMP    LCD_BYTE_WRITE
        ;
LCD_CMD:ANL     A,#1FH                  ; Isolate table index
        ADD     A,#2                    ; Offset over SJMP
        MOVC    A,@A+PC                 ; Look up in table
        SJMP    LCD_CTRL_WRITE
        ;
        ; Table of LCD commands
        ;
        DB      1                       ; 80 - Clear display
         DB       2                          ; 81 - Home cursor and
display
        DB      4,5,6,7                 ; 82 - Entry mode set
          DB        8,9,10,11,12,13,14,15    ; 86 - Display + cursor
control
        DB      0,0                     ; 8E - Padding
           DB         40H,48H,50H,58H          ; 90 - CGRAM address
commands
        DB      60H,68H,70H,78H
        DB      16,20,24,28             ; 98 - Cursor/display shift
         DB       48,52,56,60              ; 9C - Number of lines ,
font
        ;                                 ; A0 - Row 1 ( continues to
3 )
        ;                                 ; D0 - Row 2 ( continues to
4 )
```

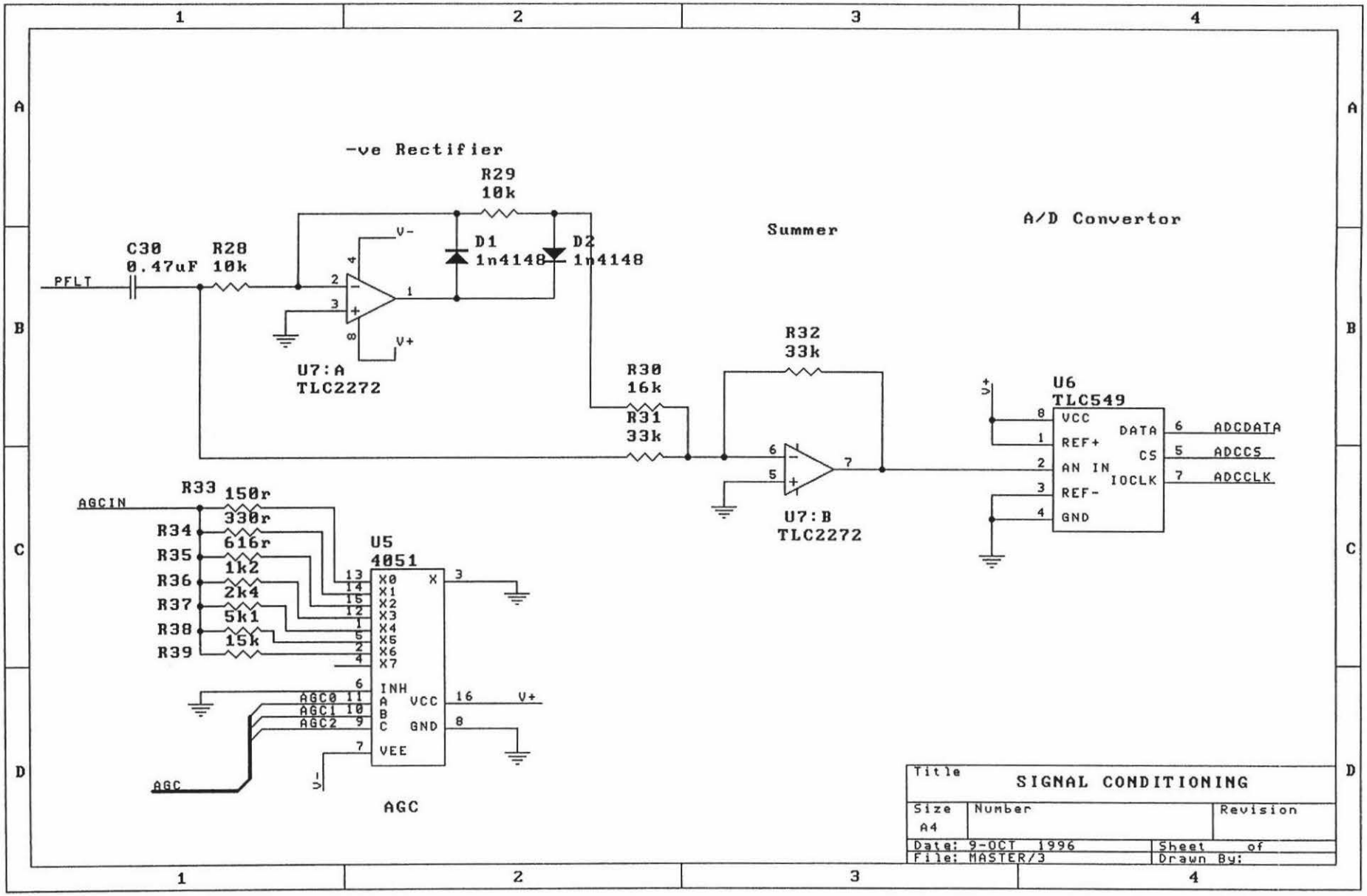
```
        ;
LCD_DATA_WRITE:
        MOV     R2,#80H                 ; RS value for data port
        JNB     LCD_SPECIAL_G,LCD_BYTE_WRITE
        CJNE    A,#67H,LCD_BYTE_WRITE   ; Lo-case g ?
         MOV     A,#7                     ; Use user-defined #7 for
lo-case g
        ;
LCD_BYTE_WRITE:
;        PUSH    IE                      ; Save interrupt enables
;        CLR     EA                      ; Disable interrupts
        SETB    ?LCD_EA_STATE           ; Set EA state flag
        JBC     EA,LCD_BYTWRT_EA        ; EA was true - now cleared
         CLR     ?LCD_EA_STATE            ; EA wasn't set , so clear
the flag
        ;
LCD_BYTWRT_EA:
        MOV     P2,A                    ; Byte to data bus
         MOV     A,R2                     ; Get RS value ; RW = 0 =
Write
        ANL     P3,#01FH                ; Clear LCD control bits
        ORL     P3,A                    ; Set RS value
        SETB    P3.5                    ; Turn on Enable bit
        CLR     P3.5                    ; Turn off Enable bit
;        POP     IE                      ; Restore interrupt state
        MOV     C,?LCD_EA_STATE         ; Fetch previous EA state
        MOV     EA,C                    ; Restore it
        RET
        ;
LCD_BUSY_WAIT:
        PUSH    ACC                     ; Save byte to send
        ;
LCD_BUSY_LOOP:
        SETB    ?LCD_EA_STATE           ; Set EA state flag
        JBC     EA,LCD_BUSYLP_EA        ; EA was true - now cleared
         CLR     ?LCD_EA_STATE            ; EA wasn't set , so clear
the flag
        ;
LCD_BUSYLP_EA:
        MOV     P2,#0FFH                ; Float output pins
        ANL     P3,#01FH                ; Clear LCD control bits
        ORL     P3,#040H                ; RS = 0 = ctl-status ; RW =
1 = Read
;        PUSH    IE                      ; Save interrupt enables
;        CLR     EA                      ; Disable interrupts
        SETB    P3.5                    ; Turn on Enable bit
        MOV     A,P2                    ; Read busy bit & save it
        CLR     P3.5                    ; Turn off Enable bit
;        POP     IE                      ; Restore interrupt state
        MOV     C,?LCD_EA_STATE         ; Fetch previous EA state
        MOV     EA,C                    ; Restore it
;        MOV     A,R2                    ; Get busy bit
         JB       ACC.7,LCD_BUSY_LOOP      ; Go around again if busy
bit set
        ;
        POP     ACC                     ; Retrieve byte to send
        RET
        ;
```

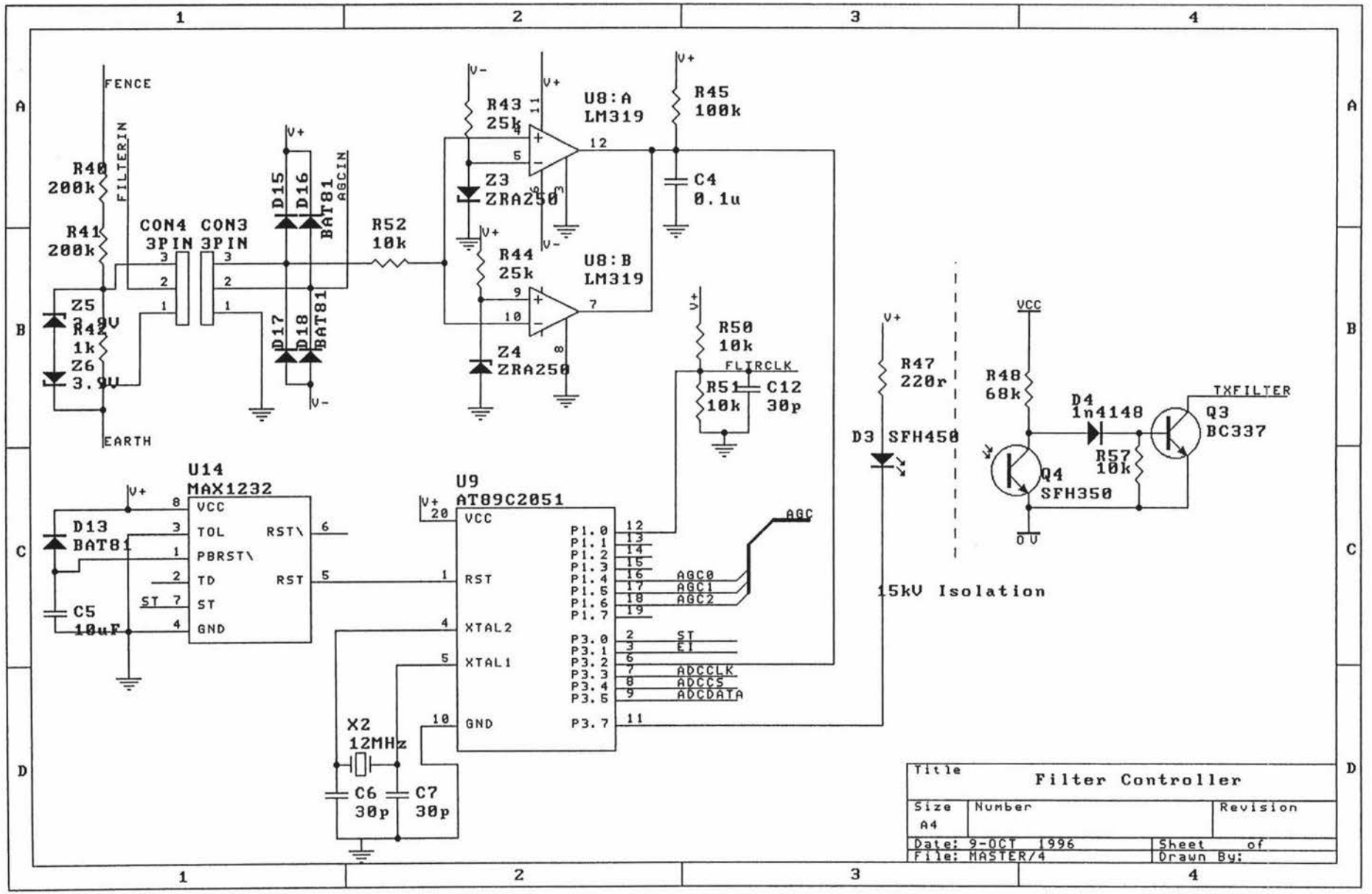
```
LIBEND

END
```

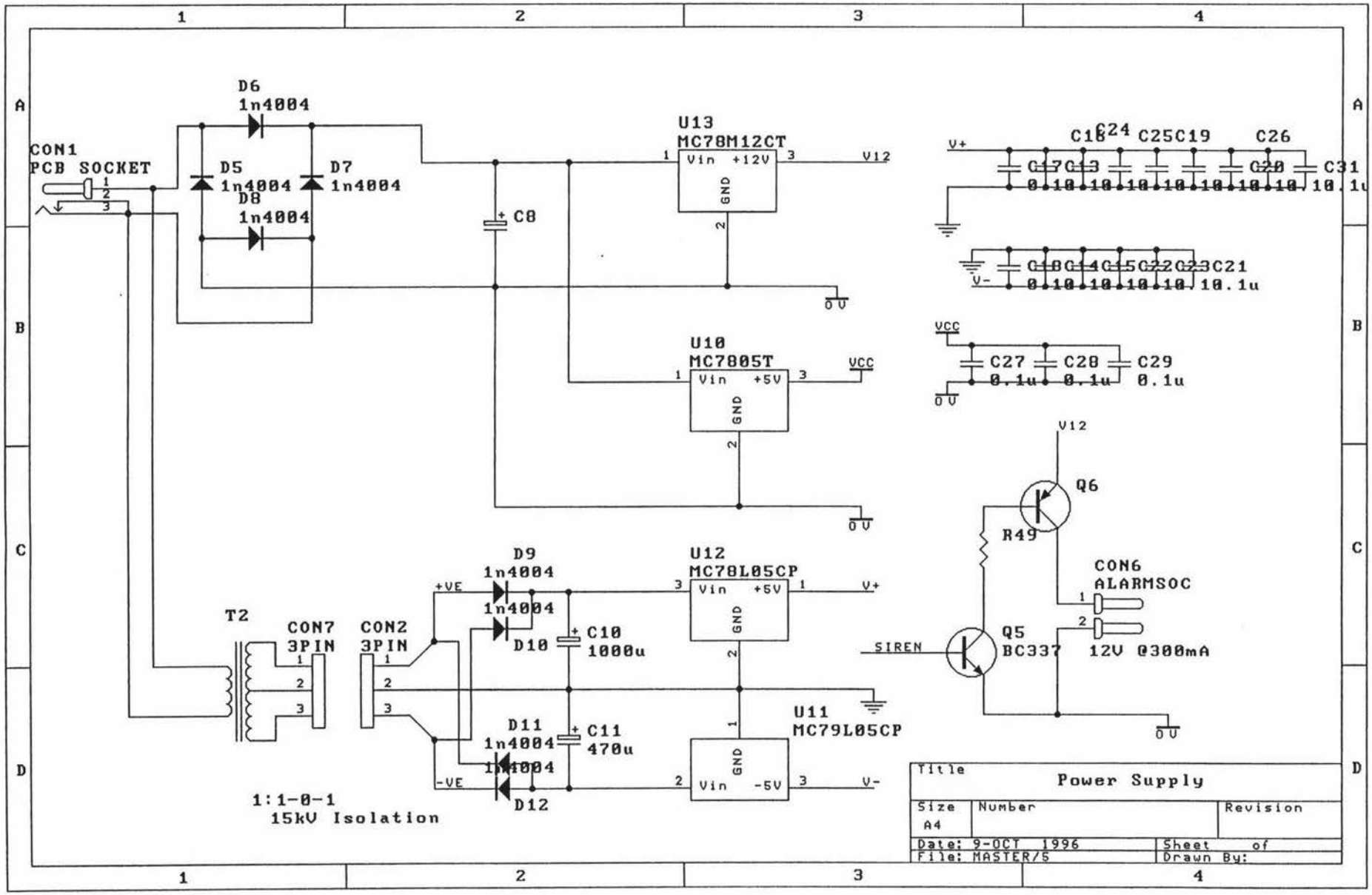
Appendix 8
Master Unit Schematics


U15
MAX1232
VCC
TOL
PBRST\
TD
STR
ST
GND
RST\
RST
D14
BAT81
C1
10u
U1
AT89C52
XTAL2
XTAL1
X1
12MHz
C3
30p
C2
30p
TXFILTER
STR
E
R/W
RS
P3.0
P3.1
P3.2
P3.3
P3.4
P3.5
P3.6
P3.7
EA\
ALE
PSEN\
P0.0
P0.1
P0.2
P0.3
P0.4
P0.5
P0.6
P0.7
P1.0
P1.1
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
P2.0
P2.1
P2.2
P2.3
P2.4
P2.5
P2.6
P2.7
ACKALARM
SETTHRES
THOUSANDS
HUNDREDS
BEEP
SIREN
DIS0
DIS1
DIS2
DIS3
DIS4
DIS5
DIS6
DIS7
R53
10k
R56
10k
U16
93C46
CS
SK
DI
DO
Q1
BC327
B1
CON5
20PIN
R2
15k
R3
220r
0 V
Title
Display and Keypad uP
Size
A4
Number
Revision
Date: 9-OCT 1996
Sheet of
File: MASTER/1
Drawn By:


FENCE
R4
R6
100k 7kV
R5
R7
100k 7kV
FILTERIN
Z1
3.9V
Z2
3.9V
EARTH
AGCIN
R8
4k7
R9
4k7
R12
100k 1%
R11
430k 1%
R10
10k 1%
U2:A
4316
U3:A
MF10
FLTRCLK10
CLKA
R13
3k3
R14
3k3
U2:B
4316
R17
27k 1%
R16
200k 1%
R15
4k7 1%
U3:B
MF10
FLTRCLK1
CLKB
R22
110k 1%
R21
510k 1%
R20
4k7 1%
R18
18k
R19
7k5
U2:C
4316
U4:A
MF10
FLTRCLK0
R23
33k
R24
18k
U2:D
4316
R27
22k 1%
R26
510k 1%
R25
5k1 1%
PFLT
U4:B
MF10
FLTRCLK11
Title
FILTER BLOCK
Size
A4
Number
Revision
Date: 9-OCT 1996
Sheet of
File: MASTER/2
Drawn By: PA


-ve Rectifier
R29
10k
C30
0.47uF
R28
10k
PFLT
V-
V+
D1
1n4148
D2
1n4148
U7:A
TLC2272
Summer
R30
16k
R31
33k
R32
33k
U7:B
TLC2272
A/D Convertor
U6
TLC549
VCC
REF+
AN IN
REF-
GND
DATA
CS
IOCLK
ADCDATA
ADCCS
ADCCLK
AGCIN
R33 150r
R34 330r
R35 616r
R36 1k2
R37 2k4
R38 5k1
R39 15k
U5
4051
X0
X1
X2
X3
X4
X5
X6
X7
X
INH
A
B
C
VCC
GND
VEE
AGC0
AGC1
AGC2
AGC
AGC
Title
SIGNAL CONDITIONING
Size
A4
Number
Revision
Date: 9-OCT 1996
Sheet of
File: MASTER/3
Drawn By:


FENCE
FILTERIN
R40
200k
R41
200k
CON4 3PIN
CON3 3PIN
Z5
3.9V
R42
1k
Z6
3.9V
EARTH
D15
D16
BAT81
AGCIN
D17
D18
BAT81
V+
V-
R52
10k
R43
25k
U8:A
LM319
R45
100k
C4
0.1u
Z3
ZRA250
R44
25k
U8:B
LM319
Z4
ZRA250
R50
10k
FLTRCLK
R51
10k
C12
30p
R47
220r
D3 SFH450
15kV Isolation
VCC
R48
68k
D4
1n4148
TXFILTER
Q3
BC337
R57
10k
Q4
SFH350
0V
U14
MAX1232
D13
BAT81
C5
10uF
VCC
TOL
PBRST\
TD
ST
GND
RST\
RST
U9
AT89C2051
RST
XTAL2
XTAL1
GND
P1.0
P1.1
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
P3.0
P3.1
P3.2
P3.3
P3.4
P3.5
P3.7
AGC
AGC0
AGC1
AGC2
ST
E1
ADCCLK
ADCCS
ADCDATA
X2
12MHz
C6
30p
C7
30p
Title
Filter Controller
Size
A4
Number
Revision
Date: 9-OCT 1996
Sheet of
File: MASTER/4
Drawn By:


1
2
3
4
A
B
C
D
CON1
PCB SOCKET
D6
1n4004
D5
1n4004
D7
1n4004
D8
1n4004
C8
U13
MC78M12CT
Vin +12V
GND
V12
0V
U10
MC7805T
Vin +5V
GND
VCC
0V
V+
C16 C24 C25 C19 C26
C17 C13 C20 C31
0.1u 10u 10u 10u 10u 10u 10u 10u 10u 10.1u
V-
C18 C14 C15 C22 C23 C21
0.1u 10u 10u 10u 10u 10u 10.1u
VCC
C27 C28 C29
0.1u 0.1u 0.1u
0V
V12
Q6
R49
CON6
ALARMSOC
12V @300mA
SIREN
Q5
BC337
0V
T2
CON7
3PIN
CON2
3PIN
1:1-0-1
15kV Isolation
+VE
-VE
D9
1n4004
D10
1n4004
D11
1n4004
D12
1n4004
C10
1000u
C11
470u
U12
MC78L05CP
Vin +5V
GND
V+
U11
MC79L05CP
Vin -5V
GND
V-
Title
Power Supply
Size
A4
Number
Revision
Date: 9-OCT 1996
Sheet of
File: MASTER/5
Drawn By:

References

Armstrong, R. H., Banks, C. H., Gill, M. P., **A Guide To Electric Fencing Relating To Principles Installation & Safety,** Hill Farming Research Organisation, (1981).

Billings, Keith H., **Switchmode Power Supply Handbook,** McGraw-Hill, (1989).

East, David, **Fences & Gates, Design & Construction,** Inkata Press, (1993).

Harris Semiconductor Literature Department, **Linear IC's For Commercial Applications 1990,** Harris Semiconductor, (1990).

Haykin, Simon, **An Introduction To Analog & Digital Communications,** Wiley, (1989).

Hughes, Edward, **Electrical Technology,** Longman, (1973).

KSC Software Systems, **Manual For 8051 Pascal Pack,** Mandeno Granville Electronics, (1989).

Lowdon, Eric, **Practical Transformer Design,** Sams, (1981).

Malmstadt, Howard V., Enke, Christie G., Crouch, Stanley R., **Electronics and Instrumentation for Scientists,** Benjamin-Cummings, (1981).

McCutchan, John, **Electric Fence Design Principles,** Electrical Engineering Department, University of Melbourne, (1980).

National Semiconductor Corporation, **Linear Databook 1986,** National Semiconductor, (1986).

New Zealand Agricultural Engineering Institute, **Electric Fencing,** New Zealand Agricultural Engineering Institute, (1986).

Slemon, G.R. and Straughen, A., **Electric Machines**, Addison-Wesley, (1980).

Speedrite International, **Speedrite Dealer Manual,** Speedrite, (1995).

Standards Association Of New Zealand, **Battery Operated Electric Fence Controllers (NZS 2210:1968),** Standards Association Of New Zealand, (1968).

Standards New Zealand, **Approval and Test Specification-Electric Fence Energisers Pt 1: Saftey Requirements For Mains-operated Electric Fence Energisers AS/NZS 3129.1:1993,** Standards New Zealand, (1993).

Standards New Zealand, **Approval And Test Specification- Safety Of Household And Similar Electrical Appliances AS/NZS 3350.1:1194,** Standards New Zealand, (1994).

Stevenson, William D Jr., **Elements Of Power System Analysis,** McGraw-Hill, (1962).

Studman, C. J., **New Zealand Fencing,** Massey University, (1989).

Williams, James M. (ed.), **1990 Linear Applications Handbook Volume 1, A Guide To Linear Circuit Design,** Linear Technology, (1990).