

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.

# Intelligent Medical Device Integration with Real Time Operating System

by

© Zaid Jan

A thesis submitted to the  
School of Engineering  
in partial fulfilment of the  
requirements for the degree of  
Master of Engineering

Department of Electronics and Computer Syetem Engineering  
at Massey University, [Albany],  
New Zealand

April 2009



## Abstract

Many commercial devices now being produced have the ability to be remotely monitored and controlled. This thesis aims to develop a generic platform that can easily be extended to interface with many different kinds of devices for remote monitoring and control via a TCP/IP connection. The deployment will be concentrated on Medical devices but can be extended to all serial device interfaces.

The hardware to be used in the development of this platform is an ARM Cortex M3 based Micro-Controller board which has to be designed to meet the requirement set by the Precept Health the founder of this platform. The design was conducted at Massey University in collaboration with senior engineer from the company.

The main task in achieving the aim was the development of the necessary software layers to implement remote monitoring and control. The eCosCentric real-time embedded operating system was used to form a generic base for developing applications to monitor and control specific devices. The majority of the work involved in this project was the deployment of the operating system to the Micro-Controller.

During the development process, several hardware issues were discovered with the Ethernet interface and were corrected. Using the generic platform, an application was developed to allow the reading of Bi-Directional pass through a communication protocol from 4 isolated serial input channels, to an Ethernet channel using TCP protocol.

## Acknowledgments

The success of this project would not have been possible without the guidance, assistance and dedication of a number of people. I would like to give many thanks to my supervisor, Dr Tom Moir for his advice, feedback and guidance and also for giving me the opportunity to conduct this thesis.

I would like to thank Tony Blomfield for offering the opportunity and funding to develop this platform, and for his on-going support throughout the project. His knowledge and experience has helped me to avoid many pitfalls along the way

Furthermore I would like to thank Nestor and Philip from eWatch for all their help and guidance through-out the project. Special thanks to Philip for explaining some off the most complex parts of the real time operating system and avoid some of the pitfalls which I have encountered during the development cycle

Finally, I would like to thank my family for their ongoing support, and I am certain they are happier than me that it's all over!

# Contents

<b>Abstract</b>	<b>iii</b>
<b>Acknowledgments</b>	<b>iv</b>
<b>List of Tables</b>	<b>x</b>
<b>List of Figures</b>	<b>xi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Thesis Overview . . . . .	1
1.2 Design Specification . . . . .	3
1.2.1 Requirements . . . . .	3
1.2.1.1 Understanding of eCosCentric . . . . .	3
1.2.1.2 Development Setup . . . . .	4
1.2.2 Constraints . . . . .	5
1.3 Major Contributions . . . . .	6
1.4 Thesis Outline . . . . .	7
<b>2 Background and Literature Review</b>	<b>8</b>
2.1 Embedded System Overview . . . . .	8
2.2 Embedded Computing Design . . . . .	9
2.2.1 Different operating system solution . . . . .	9
2.2.2 Managing embedded devices . . . . .	11
2.3 Hardware . . . . .	13
2.4 Programming embedded devices . . . . .	13
2.5 Medical Device Safety Requirements . . . . .	15
2.5.1 Software Safety . . . . .	15
2.5.2 Hardware Safety . . . . .	16

<b>3</b>	<b>Hardware design</b>	<b>17</b>
3.1	Hardware Architecture and Schematic Design . . . . .	17
3.1.1	Central Processing Unit CPU . . . . .	18
3.1.1.1	STM32F103 ARM Cortex M3 Thumb2 Processor . . . . .	19
3.1.1.2	Interrupt Controller . . . . .	19
3.1.1.3	Timer/Counter . . . . .	19
3.1.1.4	USART's . . . . .	20
3.1.1.5	On-Chip SRAM . . . . .	20
3.1.1.6	Bus Matrix Interface . . . . .	20
3.1.1.7	Programmable Input Output PIO . . . . .	20
3.1.1.8	Watchdog Timer . . . . .	21
3.1.2	Power Supply(Switch Mode) . . . . .	21
3.1.3	Serial port with optical isolation . . . . .	22
3.1.4	Ethernet Controller (Wiznet W5100) . . . . .	23
3.1.5	Micro SD Card Storage . . . . .	24
3.1.6	JTAG . . . . .	25
3.1.6.1	JTAG Adapter . . . . .	26
3.1.7	Final Implementation of the Prototype Boards . . . . .	27
3.2	Printed Circuit Board Design (pcb) . . . . .	30
3.2.1	Overview procedure . . . . .	31
3.2.2	PCB Design Considerations . . . . .	32
3.2.2.1	Trace width and trace clearance Requirements . . . . .	33
3.2.2.2	Type of vias on the PCB . . . . .	33
3.2.2.3	Floorplanning . . . . .	33
3.2.2.4	PCB Mask layer design consideration . . . . .	34
3.2.3	PCB Implementation . . . . .	34
<b>4</b>	<b>Software Design</b>	<b>37</b>
4.1	Real-time Operating System eCosCentric . . . . .	37
4.1.1	eCosCentric Source Tree Roadmap . . . . .	38
4.1.2	The eCosCentric Architecture . . . . .	39
4.1.3	Hardware Abstraction Layer . . . . .	39
4.1.3.1	HAL Start-up . . . . .	41
4.1.4	The Redboot ROM Monitor . . . . .	42

4.1.4.1	Redboot feature . . . . .	43
4.1.4.2	Virtual vector calling interface . . . . .	44
4.1.5	Kernel . . . . .	44
4.1.5.1	Kernel Boot Procedure . . . . .	45
4.1.5.2	Startup Modes . . . . .	45
4.1.5.3	Schedulers . . . . .	47
4.1.5.4	Interrupt Handling Mechanism . . . . .	47
4.1.5.5	Exception Handling . . . . .	48
4.1.6	Thread Synchronisation . . . . .	49
4.1.6.1	Mutex . . . . .	50
4.1.6.2	Semaphores . . . . .	50
4.1.6.3	Flags . . . . .	51
4.1.6.4	Spinlocks . . . . .	51
4.1.6.5	Condition Variables . . . . .	51
4.1.6.6	Message Boxes . . . . .	51
4.1.7	i/O Control System . . . . .	51
4.1.7.1	I/O Subsystem . . . . .	52
4.1.7.2	Device Drivers . . . . .	53
4.1.8	Network Support . . . . .	53
4.1.8.1	TCP/IP Stack . . . . .	53
4.1.8.2	Supported Protocol . . . . .	54
4.1.9	Configuration Tool . . . . .	55
4.1.10	eCosCentric building process . . . . .	56
4.1.11	eCosCentric Support . . . . .	58
4.2	Implemtation of Micro-Controller Bootstrap . . . . .	58
4.2.1	Boot Table . . . . .	59
4.2.2	Micro_Controller Peripheral Initalisation . . . . .	59
4.2.2.1	PIO . . . . .	59
4.2.2.2	Watchdog . . . . .	59
4.2.2.3	USART's . . . . .	59
4.2.3	Copy Program Image to SRAM . . . . .	60
4.2.4	Set Register to Define Values . . . . .	60
4.3	Implementation of eCosCentric . . . . .	60
4.3.1	Redboot . . . . .	60



4.3.2	HAL Port . . . . .	61
4.3.2.1	Structure of the Port . . . . .	61
4.3.2.2	Platform Initialisation . . . . .	61
4.3.2.3	Memory Layout . . . . .	62
4.3.2.4	CDL-File . . . . .	63
4.3.2.5	Modification of the <code>eCosCentric</code> database . . . . .	64
4.3.3	Implementation of specific drivers . . . . .	64
4.3.3.1	Ethernet Controller <code>Wiznet W5100</code> . . . . .	65
4.3.3.2	Serial interface Driver . . . . .	65
4.4	Development of Configuration Utility . . . . .	65
4.4.1	Board Configuration utility . . . . .	66
<b>5</b>	<b>Development Setup</b>	<b>68</b>
5.1	Software Environment Setup . . . . .	68
5.1.1	getting <code>eCosCentric</code> Source . . . . .	68
5.1.2	Getting <code>eCosCentric</code> Configuration Tool Ver2 . . . . .	69
5.1.3	Eclipse IDE . . . . .	70
5.1.4	OpenOCD . . . . .	70
5.2	Supported Medical Device . . . . .	71
<b>6</b>	<b>Demonstration and Testing</b>	<b>72</b>
6.1	Procedure Steps . . . . .	73
6.2	Evaluation . . . . .	76
<b>7</b>	<b>Conclusions And Future Work</b>	<b>77</b>
	<b>References</b>	<b>79</b>
	<b>Appendix</b>	<b>84</b>
A	Schematics . . . . .	84
A.1	Main Board . . . . .	84
B	Application Source Code . . . . .	85
B.1	Setip Application . . . . .	85
B.2	Virtual serial port . . . . .	90
C	Embedded Firmware . . . . .	98
C.1	Linking and building the application . . . . .	98

C.2 RedBoot Configuration File . . . . .	99
C.3 Ethernet driver W5100 . . . . .	101

# List of Tables

3.1	Code space required by different Ethernet controller using SPI interface . .	23
4.1	I/O API . . . . .	52

# List of Figures

3.1	Cortex arch M3 . . . . .	18
3.2	Switch mode power supply . . . . .	21
3.3	Isolated serial port . . . . .	23
3.4	Ethernet controller . . . . .	24
3.5	Implementation of the Micro-SD card . . . . .	25
3.6	JTAG interface . . . . .	25
3.7	JTAG adapter interface . . . . .	26
3.8	Top layer of the Final Prototype Board . . . . .	27
3.9	Bottom layer of the Final Prototype Board . . . . .	27
3.10	Final implementation of the JTAG Adapter . . . . .	28
3.11	Full picture overview of the system . . . . .	28
3.12	Initial design of this project - top layer [4 layer board] . . . . .	29
3.13	Initial design of this project - Bottom layer [4 layer board] . . . . .	29
3.14	Top and bottom layout of the main board . . . . .	30
3.15	Top and bottom layout of the JTAG adapter . . . . .	31
3.16	Top layout of the initial board . . . . .	34
3.17	Bottom layout of the initial board . . . . .	35
3.18	Ground plane of the initial board (inner layer) . . . . .	35
3.19	Power plane layout of the initial board (inner layer) . . . . .	36
4.1	eCosCentric File Directory Roadmap . . . . .	38
4.2	Hardware Abstraction Layer[27] . . . . .	39
4.3	eCosCentric startup procedure[27] . . . . .	41
4.4	Redboot ROM Monitor Architecture . . . . .	43
4.5	Kernel stratup procedure . . . . .	45
4.6	eCosCentric exception handling and execution flow[27] . . . . .	49
4.7	The eCosCentric I/O subsystem Architecture[27] . . . . .	52

4.8	Network Architecture . . . . .	53
4.9	eCosCentric GUI Configuration Tools . . . . .	55
4.10	eCosCentric Package Database Structure . . . . .	56
4.11	eCosCentric Build Process[27] . . . . .	57
4.12	Configuration utility . . . . .	66
5.1	OpenOCD layers and interface . . . . .	71
6.1	Board Configuration Utility . . . . .	73
6.2	Serial Terminal Program Configuration . . . . .	74
6.3	Network Terminal Program Configuration . . . . .	74
6.4	Received Data by Network Terminal Program . . . . .	75
6.5	Device Terminal Program . . . . .	76