



Murdoch
UNIVERSITY

On-board Communications for the Pioneer 3-AT Robot

A report submitted to the School of Engineering and Energy, Murdoch University in partial fulfilment of the requirements for the degrees of Bachelor of Engineering and Bachelor of Commerce

Dylan Jones

Supervisor: Dr Graeme Cole
Co Supervisor: Dr Gareth Lee

2012

Abstract

Originally the Pioneer Robot was purchased by Murdoch University with an on-board Linux based personal computer to facilitate teaching Software Engineering. As Murdoch's Software Engineering program has since been abandoned, the computer has been replaced with microcontrollers to facilitate Industrial Computer Systems Engineering teachings. Given that existing industrial networking protocols often require large memory allocations and significant processing power, this thesis looks at building a protocol to connect these multiple microcontrollers that uses less processing and memory requirements. Another key requirement is the ability to report exceptional events outside routine polling undertaken by the communications master.

The outcome was the development of a new master-slave protocol over an RS-485 physical layer using an New Micros NMIS 5000 board and an Arduino shield built using a National Semiconductors DS75176BN integrated circuit. The protocol was implemented within a prototype system using personal computers connected by Alfatron RS-232 to RS-485 interface converters. State machine program design was used to develop the prototype protocol in the National Instruments Labview environment. Data collision issues were overcome using a scheduling approach to refine the 'free bus' idea for exceptional event reporting. Although the final prototype would benefit from further refinement, it is advanced enough to be implemented in a functioning network.

Academic Supervisor endorsement

I am satisfied with the progress of this thesis project and that the attached report is an accurate reflection of the work undertaken.

Signed

Date

Acknowledgements

I would like to acknowledge and pay tribute to those who have supported me in successfully completing this project.

Specifically Associate Professor Dr Graeme Cole and Senior Lecturer Dr Gareth Lee, my academic supervisors, as well as the school technical staff, who provided their experience and wisdom to help guide me through this learning experience.

I would also like to thank my family and friends who have provided so much support and understanding over the course of my studies. Most importantly I wish to thank my daughters Georgia and Monique, my parents, and also my dear friend Linda Jenkins. These people will never fully know the depth of my gratitude for both their sacrifices and support, and also the wisdom I have taken from their teachings. Thank you.

Finally I would like to thank my current and previous employers who over the last eight years showed great understanding in providing flexible working arrangements to allow me to meet my family, work and study commitments. Particular thanks to Graham Wilcox, Kim Savage, Chris Traianou and Sandy Dunn.

Contents

Nomenclature	vii
List of figures	ix
List of tables	x
1. Introduction	1
2. Existing protocols.....	5
2.1 Modbus.....	5
2.2 Elektor Message Protocol.....	6
2.3 Easy-A Protocol.....	8
2.4 Scaleable Node Address Protocol (SNAP)	8
3. Development of the communication network	11
3.1 OSI model layers.....	11
3.2 Objective P1: Network development	13
3.3 Objective P2: Specify the required hardware	21
3.4 Defining the protocol	22
3.5 Objective P3: Protocol rules	24
4. Building and refining the prototype network.....	30
4.1 Development outline.....	30
4.2 Confirming communications	33
4.3 Developing a master slave relationship	37
4.4 Building data packets	38
4.5 Error detection	40
4.6 State machine implementation.....	41

4.7	Free bus implementation	44
4.8	Communications latency	46
5.	Conclusion.....	52
6.	Future work.....	53
6.1	Prototyping.....	53
6.2	Implementation.....	54
7.	Bibliography	56

Nomenclature

6811	see 68HC11
68HC11	Motorolla HC6811 microcontroller
ANSI	American National Standards Institute
Arduino	Manufacturer of the Duemilanove, Uno and Mega microcontrollers
ASCII	American Standard Code for Information Interchange – binary code used to represent alphanumeric and other special characters
Baud	Measurement equivalent to the number of bits per second
CAN	Controller Area Network
DGPS	Differential Global Positioning System
GPS	Global Positioning System
I/O	Input and output field devices
I ² C	Inter-integrated circuit
Linux	Open source modular computer and microcontroller operating system
Mb/s	Mega bits per second
MC	Master Controller
Mega	Arduino Mega microcontroller
Microcontroller	Single board computer featuring processing, memory and input

	and output functionality
OSI	Open Systems Interconnect (model)
PC	Personal Computer
Pioneer Robot	Pioneer 3-AT Robot manufactured by MobileRobots
SBC	Single board computer
SNAP	Scaleable Network Address Protocol
SPI	Serial Peripheral Interface
TTL	Transistor-Transistor Logic
RS-232	Electrical characteristics standard (also known as TIA/EIA-232)
RS-485	Electrical characteristics standard (also known as TIA/EIA-485)
USB	Universal Serial Bus

List of figures

Figure 1 - Modbus Query-Response Cycle (Modicon Inc 1996).....	5
Figure 2 - Elektor Message Protocol (EMP) data packet (Nickel 2011)	7
Figure 3 - Elektor Message Protocol 'hybrid' mode (Nickel 2011).....	7
Figure 4 - SNAP data packet (High Tech Horizon 2002)	9
Figure 5 - SNAP header bytes (High Tech Horizon 2002).....	9
Figure 6 - OSI model depicting layers and functions (Zimmerman and Schweitzer III 1996)	12
Figure 7 - RS-485 network (Lammert Biers 2012).....	15
Figure 8 - Simplified waveform example of RS-485 (Wikipedia 2012)	17
Figure 9 - Example of bias resistor installation (Kugelstadt 2011)	18
Figure 10 - Communications networks topologies (Rouse 2010).....	20
Figure 11 - Resultant waveforms from a system with a 3 metre spur (left) and short spurs (right) (Maxim Integrated 2001).....	21
Figure 12 - Max485 wiring diagram (Arduino 2012).....	22
Figure 13 - Alfatron interface converters shown connected.....	33
Figure 14 - Original Master (write message code shown)	35
Figure 15 - Communications configuration settings subVI	37
Figure 16 - Master prototype user interface (final)	38
Figure 17 - Slave prototype user interface (final)	38
Figure 18 - Message packet construction subVI	39
Figure 19 - Original unpack message subVI	40
Figure 20 - Early state machine implementation.....	42
Figure 21 - Free bus scheduling implementation	46
Figure 22 - Send-reply cycle observed in the prototype.....	47
Figure 23 - Poll cycle.....	48
Figure 24 - Free Bus message sent at 9600 baud	50
Figure 25 - Free Bus message sent at 9600 baud	51

List of tables

Table 1 - Primary objectives of this thesis	2
Table 2 - Secondary objective of thesis.....	3
Table 3 - Message packet structure	25
Table 4 - Polled data packet (from master)	26
Table 5 - Polled data packet (from slave)	26
Table 6 – Initiate free bus message packet structure	27
Table 7 – Free bus message packet structure (from slave)	27
Table 8 – Free bus message packet structure (master acknowledgement)	28
Table 9 – Emergency message packet structure	29
Table 10 - Development PCs specifications	31
Table 11 - A450 interface converter dip switch settings	32
Table 12 - Initial communications configuration settings.....	36