



THE UNIVERSITY OF WESTERN AUSTRALIA

FINAL YEAR PROJECT THESIS

**Advanced Embedded Systems:
Designing an Expansion Board for
the EyeBot Robotics Platform**

Thomas Smith

20515543

Supervisor: Professor Thomas Bräunl

School of Electrical, Electronic & Computer Engineering

October 27, 2014

Thomas Smith
16 Karrul Way
Greenfields WA 6210

October 27, 2014

The Dean
Faculty of Engineering Computing and Mathematics
The University of Western Australia
35 Stirling Highway
CRAWLEY WA 6009

Dear Sir,

I submit to you this final report entitled “*Advanced Embedded Systems: Designing an Expansion Board for the EyeBot Robotics Platform*” in partial fulfilment of the requirement of the award of Bachelor of Engineering.

Yours Faithfully,

Thomas Smith

Abstract

During the course of the year work was undertaken on developing the next generation of the EyeBot robotics platform, to be used at the University of Western Australia as a research and educational tool. This involved three core areas of design implementation. The development of GUI application, the update and porting of the EyeSim simulator, and the design of a new expansion board.

The authors work was focused on developing the expansion board with many changes made to the previous design, correcting errors and increasing functionality all while minimising cost where possible without sacrificing quality.

Two prototypes were manufactured and testing revealed some critical hardware problems, primarily to do with a large power supply ripple, setting back the development the EyeBots until the problems could be fixed and new prototypes manufactured.

The new prototypes have corrected the power supply ripple problem, however, a new problem has emerged with the USB connectivity, yet again delaying development.

Acknowledgments

I would like to thank my supervisor Professor Thomas Bräunl for his guidance and support during the year, and for the opportunity to work on a design project.

I would also like to thank Ivan Neubronner for his assistance throughout the year in both the design stage of the project and during the hardware troubleshooting.

Finally, I would like to thank my family for the support they have given me throughout the year.

Contents

| | |
|--|------------|
| Abstract | ii |
| Acknowledgments | iii |
| Introduction | 1 |
| Expansion Board | 3 |
| Design Considerations | 3 |
| Analysis of Past Recommendations | 3 |
| Design Differences | 4 |
| CPU | 5 |
| Programming the CPU | 5 |
| USB | 5 |
| USB Driver | 6 |
| Power Supply | 7 |
| Motors | 7 |
| Analog to Digital Converter | 7 |
| Microphone | 8 |
| Position Sensing Devices | 8 |
| Servos | 8 |
| Digital Pins | 9 |
| Manufacture | 10 |
| Software | 11 |
| Atmel Software Framework | 11 |
| Test Programs | 11 |
| Application Code | 11 |
| Prototype Testing | 12 |
| Power Supply | 12 |
| PDI Interface | 12 |

| | |
|---------------------------------------|-----------|
| USB Interface | 13 |
| Analog to Digital Converter | 13 |
| Motors and Servos | 13 |
| PSD Sensors | 13 |
| Microphone | 14 |
| Second Prototype | 14 |
| Future Work | 15 |
| Conclusion | 16 |
| Bibliography | 17 |

List of Figures

| | | |
|---|--|---|
| 1 | Embedded Hardware Choices [1] | 2 |
| 2 | Expansion Board | 3 |
| 3 | XMEGA A1U Pinout and Block Diagram [6] | 6 |

Introduction

The EyeBot is a robotics platform used at the University of Western Australia to teach embedded systems, with a particular focus on computer vision. At the time of writing, the EyeBot M5 is being used in labs at UWA and has been for quite some time as it is the last EyeBot revision to get past the design stage. However, by today's standards, the EyeBot M5 hardware is quite outdated, running a 32 bit Motorola 68332 CPU with a 16 by 8 character (or 128 by 64 pixel) monochrome LCD [1]. Many of the M5's are plagued with various problems affecting their performance in the labs, largely due to there being no suitable replacement parts available when old hardware failed, such as the camera modules that were designed specifically for the EyeBot M5's. A solution was required to replace the outdated EyeBot hardware, allowing for more modern and powerful embedded algorithms to be used, and in 2006 the EyeBot M6 was created.

The EyeBot M6 boasts a powerful ARM9 CPU capable of running a Linux Operating System, stereo cameras with image processing being handled by a Xilinx Spartan 3E FPGA. Also on board was a large colour Touch-Screen Display, USB, Ethernet, and Bluetooth [2]. However, while the M6 met all the requirements of a modern, powerful replacement to the M5, it was soon discovered that due to the complexity of the design it was both difficult for new users to work with, and also very costly to repair should there be a problem, on top of the already quite high build cost. It was decided a new solution was required that was cheaper to manufacture and maintain, while also being powerful and easy to use.

With the abundance of low cost Single Board Computers available today, such as the Raspberry Pi and Beagle Board, it was the logical choice to design the next generation of EyeBot's around these in order to significantly reduce both cost and complexity. A comparison check was performed on various Single Board Computers, as seen in figure 1, in order to find the board most suitable for the task of replacing the EyeBot M5's. The two top contenders were the BeagleBoard-xM and the Raspberry Pi, however due to lack of available display and supply issues at the time with the Raspberry Pi, the decision was made to begin designing the EyeBot M7 with the BeagleBoard-xM.

In 2012, Andrew Adamson began designing the first revision of the EyeBot M7 [3] using a BeagleBoard-xM. Adamson realised that in order to satisfy the requirements of the next generation EyeBot, an expansion board extending the functionality of the BeagleBoard was required. Adamson designed and prototyped two expansion boards.

Development perhaps would have continued with the BeagleBoard-xM had it

Embedded Hardware Choices

| | Pi | Beagle | Gumstix | EyeM6 |
|-----------|----------------------------------|--|----------------------|-------------------------|
| Cam | 1✓ | ✗ | ✗ | 2✓✓ |
| LCD/Touch | ✗ | ✓ | ✓ | ✓ |
| I/O Pins | ✗ | ✓ | ✗ | ✓ |
| LAN | ✓ | ✓ | ✓ | ✓ |
| WiFi | ✗ | ✗ | ✓ | ✓ |
| CPU | ARM11 (iPhone 1,3) 700 MHz | ARM CortexA8 (iPhone 3GS,4) 1GHz | ARM CortexA8 1GHz | ARM 5 XScale 600 MHz |
| Price | ✓✓ | ✓ | ✓ | ✗ |

Bräunl 2014

3

Figure 1: Embedded Hardware Choices [1]

not been discovered that the Raspberry Pi could interface a Touch Screen Display via its SPI ports [4], at the cost of reduced framerate. This now made it the more desirable solution due to its lower cost and dedicated camera interface.

Development of the new EyeBot based around the Raspberry Pi began this year with a team of 3, with work being separated into the three main areas:

1. The development of GUI application to run on a Raspberry Pi and provide the top level interface between the user and the EyeBot.
2. The update and porting of the EyeSim simulator to reflect the changes that come with the new EyeBot.
3. The design of a new expansion board to act as the interface between the Raspberry Pi GUI application and the low level hardware, such as motors, servos and sensors.

The authors focus during the year has been on the design and testing of the expansion board. The goals set out at the start of the year were to design a low cost expansion board meeting the requirements of the EyeBot robotics platform. It was also hoped to have the boards ready for use by the time the semester 2 Digital and Embedded Systems labs required them

Expansion Board

The Expansion Board is an interface board that will exist between the Raspberry Pi and the low level hardware found on the EyeBot. This is required as many Single Board Computers including the Raspberry Pi do not provide enough I/O pins in order to interface all the sensors required, and do not allow for interfacing motors and servos without additional hardware.



Figure 2: Expansion Board

Design Considerations

During the design process of the new expansion boards, the previous expansion board was carefully examined to identify any areas where improvement could be made. Past theses pertaining to the expansion board were also analysed and recommendations for future work sections provided many useful design directives for the new board.

Analysis of Past Recommendations

Two documents were analysed to find recommendations for changes to the design of the expansion board, these were theses by both Andrew Adamson [3], and Remi Keat [5]. Recommendations they had made that are relevant to the new design are analysed in the table below.

| Recommendation | Implementation |
|--|--|
| Include a JTAG and/or PDI (Program and Debug Interface) interface for programming and debugging. | A PDI interface was incorporated into the new design, allowing for initial programming. However a JTAG header was not included as it would require too much space and would need disabling when the analog pins were required. |
| Implement a bootloader on the CPU. | A USB bootloader was found for the ATxmega128A1U. |
| Increase power supply current so that motors do not brown out BeagleBoard. | Power supply current on both rails increased to maximum of 5A, and motor power is selectable by a jumper between 5V and a direct connection to the input supply. |
| Add debugging/general purpose LEDs. | Four LEDs were included on the new expansion board. |
| Use through-hole pin headers instead of surface mount. | Through-hole pin headers were used. |
| Include a microphone, speaker, and IR receiver in the next revision. | A microphone was included on the new expansion board, however a speaker and IR receiver had already been designed into the adapter board for the Raspberry Pi's LCD, so were not needed. |
| Apply a reference voltage to the ADC. | A 2.7V reference was supplied to the ADC as this was the maximum allowable with a 3.3V supply. |

Design Differences

The newly designed expansion board features some key differences from the previous board designed to work with the BeagleBoard-xM. These are listed below.

- The new board is much smaller than the previous, being designed for the Raspberry Pi, which is smaller than the BeagleBoard-xM.
- The number of board layers has been reduced from four to two, in order to lower production costs. However, this coupled with the reduced size made

board layout more challenging, potentially increasing the chances of a hardware fault.

- The CPU was changed to the newer ATxmega128A1U from the previous ATxmega128A1. The new CPU has fewer silicon bugs and incorporates a built-in USB interface, removing the need for the USB to SPI interface chip, thus reducing component count and design complexity.
- The I2C PWM generation chip, PCA9685, previously used to control the servos, has been removed as there are enough pins with PWM generation circuitry on the CPU to control up to 16 servos at once.
- A PDI header has been included.
- Four general purpose LEDs have been added.
- A microphone and amplification circuitry has been added.

CPU

The CPU chosen to control the Expansion Board was the ATxmega128A1U, as it provided all the capabilities needed built in. Its 100 pins allow it to interface a plethora of sensors and actuators with many spare analog and digital pins brought out for general purpose use. The ATxmega128A1U . While only an 8-bit CPU, it is powerful enough to do the task of interfacing the low level hardware, with heavy processing tasks taking place on the Raspberry Pi. Communication between the CPU and the Raspberry Pi is performed by means of built-in USB hardware support.

Programming the CPU

The CPU can be programmed through the PDI header with compatible programmer such as the AVR Dragon, AVRISP mkii, or the Atmel ICE. Only one initial use of the PDI header should be needed, and that is to flash the USB bootloader onto the CPU, allowing future programs to be uploaded via the USB interface. The USB bootloader is made by Atmel specifically for the ATxmega128A1U to simplify device firmware upgrades. An application is also provided, Atmel FLIP, to connect to the bootloader and upload the firmware with instructions available in Atmel's app note AVR1916 [7].

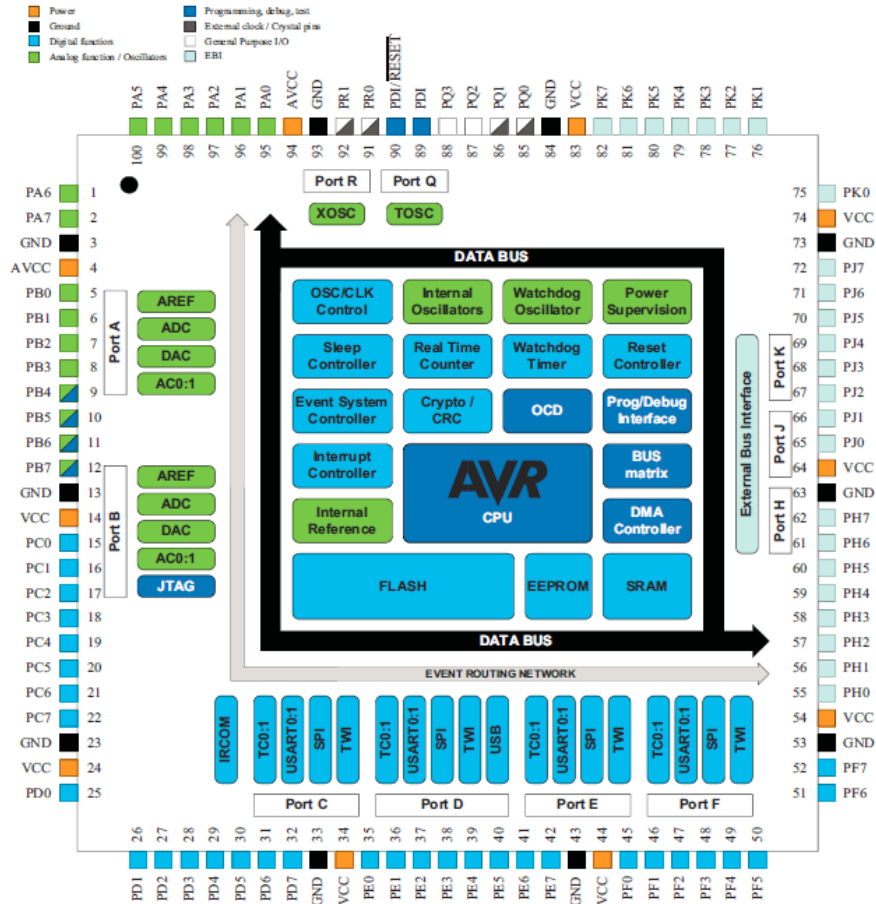


Figure 3: XMEGA A1U Pinout and Block Diagram [6]

USB

The use of a standardised communication protocol such as USB allows the Expansion Board to be used with virtually any computer able to provide the correct communication commands. This means any future upgrades to the EyeBot’s processing capabilities will be as simple as swapping out the Raspberry Pi for another Single Board Computer and then porting the Application across if necessary. It also means the Expansion Board can be used with any laptop or desktop so long as the correct drivers are installed, however no application is currently being developed for Windows or Mac, so at the moment the only method of communicating to the expansion board outside of the Raspberry Pi is via programs such as PuTTY.

USB Driver

In order for host computers to be able to communicate with the expansion board, they must first identify a compatible driver that specifies the USB device’s communication method, in the case of the expansion board this is the Communications Device Class, CDC, which is a serial communication over USB implementation.

However, before a driver could be created for the expansion board, a unique identifier was required, the USB VID (Vendor ID) and PID (Product ID), which would differentiate the expansion board from other USB devices.

Purchasing a VID from the USB Implementers Forum is the recommended way of acquiring the ID's, however the prices are prohibitively high at \$5000 US [8], so another solution was needed. Two other options existed that would solve the problem, 1) Redesign the board with the FTDI chip from the previous implementation, as that had drivers available for it already, or 2) purchase a VID/PID from a third party for a fraction of the cost, although this would mean that the expansion board would not be eligible for USB certification. It was decided that the best option was to buy the VID/PID from the third party.

Power Supply

A big part of the design process for the Expansion Board was spent on getting the power supply right. This was due to the board needing to be able to take in a fairly wide range of supply voltages and provide both 3.3V and a 5V rail of sufficient power capabilities to power all the peripherals, including the Raspberry Pi. Both supply rails utilise a buck converter capable of providing up to 5A of output current and will work with an input voltage up to 15V. Input current is limited by means of a Positive Temperature Coefficient, PTC, resistor.

The 5V regulator is switched on by the MCU after it has powered up, meaning the expansion board is always on before the Raspberry Pi, and the 5V rail can be disconnected if the battery voltage drops too low. For testing the 5V rail before the boards have been programmed a solder jumper was designed in to the new prototype that allowed the 5V regulator to turn on with the power.

Motors

Two Allegro A3906 dual H-bridge motor drivers allow the board to control 4 DC motors at up to 1A of current per motor, with overcurrent fault notification to the CPU. Two PWM signals per motor are used to obtain precise drive control over each motor. Each motor pin header contains the positive and negative pins of the motor as well as 5V, GND and two pins for interfacing a quadrature encoder, allowing for precise speed and direction control. To help mitigate excessive power use on the 5V rail, the motor drivers have a selectable input voltage by means of a jumper, allowing them to run directly off of the battery.

Analog to Digital Converter

The Xmega has two feature rich 12 bit differential analog to digital converters built in, which are capable of providing very accurate readings. Seven general purpose ADC inputs are brought out to a header to allow for easy interfacing with various sensors. The first four ADC pins on the header also double as outputs for the two stereo DAC's contained within the chip, although external circuitry will be needed to filter and buffer the signal should it be required.

The ADC requires an external voltage reference in order to measure signals greater in voltage than $VCC/1.6$. The maximum permitted voltage reference for a 3.3V VCC was 2.7V, which was just enough to measure the PSD sensors. This is supplied by a resistor divider network from the 3.3V rail.

Microphone

A microphone is included in the design for the expansion board to allow the EyeBot to listen and record sound to the Raspberry Pi. Significant improvement has been made to the microphone amplifier circuitry over the older EyeBot M5 and earlier, hopefully allowing for clearer sound detection and opening up the EyeBot to such applications as voice and sound recognition.

Previous versions of the EyeBot had an error in the microphone amplifier circuitry, causing too large a gain and subsequently causing an unstable output. This has since been corrected in the new design.

Position Sensing Devices

These sensors emit IR light and use the time taken for the light to bounce back to calculate distance. The sensor translates this distance to an analog voltage which is read by the microcontroller's ADC. There are six positions available for the PSD sensors on the expansion board, however unused pins will still be able to be used as general purpose ADC. The PSD sensors output voltage can extend past the 2.7V ADC reference, however the range lost is minimal when this occurs.

Servos

The servo header is capable of connecting to up to 14 servos, although it is unlikely the expansion board will be able to drive so many at once. Ways around this would be to either power the servos from a separate power source, which would require

modifying the header, or by trying a staggered control approach by only sending a signal to one or two servos at a time.

Digital Pins

Twenty general purpose digital input/outputs are provided on the expansion board, all individually programmable from the Raspberry Pi. Input protection is provided on the digital I/O pins through current limiting series resistors and diodes from GND to the pin, and from the pin to VCC.

Four digital pins are also connected to LEDs, allowing for a simple and easy way to debug or notify the user

Manufacture

After the design had been finalised and the schematic was created, Ivan Neubronner created the PCB layout. The prototype boards were manufactured by ITEAD Studio, a PCB pooling service in China. This was done to significantly decrease the cost of manufacturing the PCB's, as tooling costs are divided amongst many PCB designs. For example, the cost to get 10 2 layer boards made at ITEAD Studio is only \$19.90 US [9]. While the cost of the previous expansion board from PCB Cart was \$210 for tooling plus \$30 for each board [3].

Components were sourced from various distributors based on a range of factors including availability, cost, and quality (where specific components had not been specified, e.g. resistors and capacitors). Three distributors were eventually used to purchase the components from, these were, element14, RS-Components, and Altronics. For the first prototype boards only enough components to populate two boards were purchased, where minimum order quantities would allow, in order to minimise costs in the event of a design change.

Once components and boards had arrived, the expansion boards were assembled by Ivan using a hot air solder method. This is required as the surface mount components are too dense and in some cases have too fine a pin pitch to practically use a soldering iron. The hot air method also speeds up manufacturing time as all components on a layer can be soldered at once.

Software

The software side of the Expansion Board is being written in Atmel Studio 6 version 6.2, a Visual Studio environment designed specifically for programming Atmel's CPUs. A USB bootloader is initially flashed onto an empty CPU which allows future code to be simply uploaded straight from the Raspberry Pi with no need to change anything.

Atmel Software Framework

Atmel Software Framework, ASF, is a collection of libraries written by Atmel to provide services and drivers for the peripherals of the CPU. Using ASF helps speed the production as complex code, such as the USB stack, has already been written and is fully tested and supported with example documentation. At the time of writing, the project code is being used with ASF version 3.19.0. ASF first requires the user to specify a board that is being used from a list of development boards, or a user board. The user board was selected for this project as the design is unlike Atmel's development boards and will need to specify some features of the board.

Test Programs

A program was written to test the core functionality of the expansion board, built on top of the ASF, it decoded basic serial commands arriving over USB and sent back data to be displayed in a PuTTY window for debugging purposes. Various drivers were included, allowing quick implementation and testing of timers, PWM signals, ADC readings, USB CDC, etc. The test code used on the prototype boards is included in the appendix, however only top level source files have been created by the author, along with the config files and board initialisation file. All other files in the directory are imported into the project by ASF when a driver or service is included.

Application Code

Application code has not yet been completed, largely due to the delays incurred by the hardware faults and the inability to completely test various parts of the expansion board. The application code will be able to reuse most of the test code, as it was written to be used that way. Only small parts will need to be modified, such as the communications encoding and decoding of commands.

Prototype Testing

Two initial expansion board prototypes were created for testing with key test criteria covering

- Power supply voltages
- PDI interface
- USB interface
- Analog to digital converter
- Motors and servos
- PSD sensors
- Microphone

Power Supply

Initial testing of the power supply hardware began well, with voltages measuring well within specification on a multimeter. However, upon testing both motors and servos it was discovered that the CPU would freeze. After ruling out a software issue the voltage rails were checked with a digital storage oscilloscope. With a small load both the 3.3V and 5V rail exhibited a fairly large amount of noise. Much more than the switching regulator should according to its datasheet. However, it became clear why the CPU was freezing when a motor or servo was used after applying a load to the 5V rail and measuring the resultant ripple voltage. Peak to peak ripple voltage exceeded 1V on both voltage rails when a load was applied to the 5V rail. Which is more than enough to cause the CPU to enter a fault state and shutdown. This was a major problem, as it rendered the boards essentially useless for more stringent testing as they could not control motors or servos, and the ripple also interfered with taking accurate ADC measurements. It was quickly identified as a problem with the ground return of the switching regulators and the modification was made to the design ready for the next prototype. Input capacitance was also increased for the second prototype, along with a slightly bigger inductor on the 5V rail, in order to minimise noise as much as possible.

PDI Interface

The Program and Debug Interface is very simple to test, it just requires access to a working piece of software. In this case, it is the USB DFU bootloader. When testing with an AVR Dragon, which supports PDI on the ATxmega128A1U [10], programming success was sporadic and often crashed Atmel Studio. The solution to this is to wire up the dragon to the expansion board via the JTAG interface. This is still possible despite not including a JTAG header as the required JTAG pins are brought out on the PSD header, although if JTAG is active four of the six ADC pins on the PSD header cannot be used. It is believed that the fault with the PDI interface lies with the AVR Dragon, and as such it is recommended that an alternative programmer is purchased going forward.

USB Interface

Once the USB DFU bootloader was programmed onto the chip, the application code was then flashed over via the USB port. It was then possible to install the required driver and test the USB connection using PuTTY. No problems were encountered with the USB interface.

Analog to Digital Converter

Due to the ATxmega128A1U's ADC being quite complex, testing progressed slower here than with other parts. A problem was encountered when the ADC settings, such as voltage reference, clock speed, etc, were changed while the application was running, where it would seem to randomly freeze the CPU. It was suspected that the excessive voltage ripple was to blame for this, however, there had been a multitude of silicon bugs related to the ADC in previous versions of the CPU, so it could also be possible that this is the cause.

Motors and Servos

Both motors and servos could not be tested properly on the expansion board, as if they drew too much current they would cause excessive voltage ripple on the power rails causing the CPU to either brown out or switch off due to high voltage spikes.

PSD Sensors

Only two of the PSD sensor pin header connections could be tested as the remaining four were being used for the JTAG interface. Testing of the PSD sensors went well

with the 2.7V reference being just high enough to get a usable range of readings.

Microphone

As the communication interface between the expansion board and the Raspberry Pi has not yet been completed, the only way for the microphone to be tested is to probe the output of the amplifier. Testing indicated that the output had quite a low noise level and swung the full range from 0V to 2.7V, when a loud sound was present.

Second Prototype

Recently the new prototype boards have been put together by Ivan, with initial voltage testing showing much better ripple performance more in line with what should be expected. However, a new problem has arisen where the USB now does not work for an, as at the time of writing, unknown reason. The ADC will also still freeze the CPU if the settings are changed, however, that can be avoided by leaving the settings at defaults after initialisation. Further testing still needs to be conducted, particularly around the communication between the expansion board and the Raspberry Pi.

Future Work

Some recommendations for future work are given below.

- Program a safety feature into the expansion board that monitors connection status and activates a safe mode in the event of communication loss from the controlling computer.
- Incorporate more of the advanced CPU features available on the ATxmega128A1U into the software, such as using DMA to free up the CPU, as well as trying out sleep modes in order to conserve power.
- Maintain documentation, reflecting any changes or updates made to the EyeBot GUI, expansion board or the EyeSim simulator.
- Create demo programs that utilise the added features in the new expansion board.
- It is possible that someday the Raspberry Pi foundation will release a display for the Raspberry Pi's, making the adapter board that exists between the LCD and the Pi redundant. This would mean the IR receiver and speaker would no longer be available to the EyeBot. In that case it may be better to design a small speaker and IR receiver into the expansion board if and when the next design is underway.

Conclusion

The goal of this project was to design a low cost expansion board that would interface with a Raspberry Pi to create the next generation of EyeBots. The result is an expansion board that is relatively low cost to manufacture, especially when compared with the cost of previous versions of EyeBots, yet has all the features expected of the EyeBot platform.

Various hardware issues arose over the course of testing, pushing the expected completion date back. Which unfortunately meant that the EyeBot M7's were not ready by the time the Digital and Embedded Systems labs were underway. However, with a little bit more troubleshooting and testing to overcome some new hardware problems, the current batch of expansion boards should perform well in the labs. So next years students will hopefully see the new EyeBot M7's for the first time.

Bibliography

- [1] T. Bräunl. *EyeBot M5 Documentation*. 2014. URL: <http://robotics.ee.uwa.edu.au/eyebot/>.
- [2] T. Bräunl. *EyeBot M6 Documentation*. 2014. URL: <http://robotics.ee.uwa.edu.au/eyem6/>.
- [3] A. Adamson. “Design, Manufacture and Testing of a Robot System for UWA EyeBot Program”. University of Western Australia, Nov. 2012.
- [4] M. Williams. *Raspberry Pi with a 3.2 TFT with Touch control*. 2014. URL: <http://marks-space.com/2013/05/23/raspberry-pi-with-a-3-2-tft-with-touch-control/>.
- [5] R. Keat. “Design of the RoBIOS Library for the new EyeBot”. University of Western Australia, 2013.
- [6] *8/16-bit Atmel XMEGA A1U Microcontroller*. Atmel. 2012.
- [7] *Atmel AVR1916: USB DFU Boot Loader for XMEGA*. Atmel. 2012.
- [8] USB-IF. *Getting a Vendor ID*. 2014. URL: <http://www.usb.org/developers/vendor/>.
- [9] ITEAD Studio. *PCB Prototyping*. 2014. URL: http://imall.iteadstudio.com/open-pcb/pcb-prototyping.html?pcb_size=126.
- [10] Atmel. *AVR Dragon*. 2014. URL: <http://www.atmel.com/tools/avrdragon.aspx>.