

Using standardised hardware modules to implement complex design projects for teaching Embedded Systems

Mark Beckerleg, John Collins

Engineering Research Institute
School of Engineering
Auckland University of Technology

Abstract

The aim of the second year diploma embedded systems paper is to equip students to design and develop small-embedded systems. Previously it has been difficult to create 'real life' design projects that can be completed within a short duration of 4 weeks. To overcome this difficulty we have created a peripheral input output (I/O) board with a wide range of devices allowing students to produce and test their programs. In the current paper, students are given three design projects to be completed within the semester. Previous successful projects include: clock radio, parking meters, frequency meters, an audio generator, and a remote data logger. These projects require a comprehensive functional and technical description, full circuit diagrams, structure diagram and program. Normally it would not be possible for a student to complete this workload because of the time required to create the hardware. We have developed modularised hardware that can be configured in a variety of ways to overcome this problem. With careful selection of project briefs, the modularised hardware can be incorporated in testing the student's final program design for the specified project.

The modularised hardware has been designed in two parts. The first part is the "microcontroller board" which uses the Atmel Atmega128 microcontroller, a JTAG ICE, and an interface board that brings out the microcontroller ports to standardised connectors. The benefit of integrating a low cost ICE is it provides a flexible learning tool for students to debug their programs. The second part is a peripheral I/O called the "Mega Applications Board". This has been designed to allow for maximum flexibility in projects. It interfaces to the microcontroller via the standardised connectors, and has numerous input and output devices, allowing all the onboard systems of the microcontroller to be used.

The learning outcomes from this approach have been excellent. Students have completed these relatively complex design projects within four to six weeks.

Keywords: embedded systems, project based learning, innovative,

Introduction

There are two broad approaches used to teach embedded systems. The first is to use specialised equipment to create one large laboratory. Examples of this approach are a ping-pong levitator [1], an autonomous car [2] or robots [3]. The advantages of this approach are the types of project undertaken (e.g. a robot creates enthusiasm), and the students can generally implement a difficult and intriguing design. The disadvantages are: the complete

*"Proceedings of the 2005 ASEE/AaeE 4th Global Colloquium on Engineering Education
Copyright © 2005, Australasian Association for Engineering Education"*

system even when broken down into components is complex for beginner students; the system relies on all sections functioning before a result can be seen; strategies are necessary to cope with students who are behind (e.g. given model solutions of non performing sections); because the same laboratory is repeated annually, the students may acquire previous years results to complete the ir tasks. These larger types of projects are better suited to more advanced papers such as a project design course, where a new project is created each year. Examples from previous years of these projects are a propeller clock, inverted mobile pendulum, or a heart-rate monitor.

The second approach in teaching embedded systems is to use a peripheral I/O board with a wide range of I/O devices to create a variety of projects. This has the advantage of: keeping initial problems simple so that the student can progress in small steps; ability to pace the increased problem complexity until the student is completing a comprehensive embedded systems design; and finally this technique allows the lecturer to implement a wide range of projects that can be modified each year. At AUT, we have built a microcontroller board and peripheral I/O mega application board for use with this approach.

Embedded systems design is taught at AUT in a range of papers, including Microcontroller Systems in which C fundamentals and the programming of onboard devices are taught. This paper leads into Embedded Systems, where complete embedded system design is taught. The microcontroller and mega application boards are used in both papers.

Structured programming [4] and C are taught within the microcontroller systems paper. C was chosen because of its popularity in industry, its ability for low level programming (e.g. bit and register manipulation) and its relative ease of use. Assembly programming is not taught (although it has been introduced in an earlier paper) as high level languages are more often used in New Zealand industry.

Although the depth of the programs in this paper is relatively simple, they can still be made to represent real life applications. Example programs would include a bilge-pump control system or a logic probe. These real-life programs are downloaded and tested on the microcontroller and mega application board. The onboard devices programmed include general parallel I/O, serial I/O (USART), high speed serial (SPI), analogue to digital converters, 8 and 16 bit timers for real time interrupts, input capture, and signal generation

To avoid poor programming techniques the students must construct the structure diagram before they begin writing the C program. Historically this has been difficult to achieve as the students would attempt the code first and then retroactively create a structure diagram from the code. Usually this resulted in an inadequately designed and unsuccessful program. We have overcome this problem by developing a software program that can generate the C program outline from a structure diagram. This innovation has been a resounding success in encouraging the student to create structured programs.

As the embedded systems paper follows on from the microcontroller systems paper with a complete embedded systems design process, the level of programming complexity is increased. The student will complete three designs, each of which is broken down into four parts. The first part requires a full functional and technical description. The functional description details the features of the system and an operator manual with a step-by-step explanation of how the system will operate; the technical description describes the hardware, type of processor, I/O devices, power requirements and environmental operating conditions.

*“Proceedings of the 2005 ASEE/AaeE 4th Global Colloquium on Engineering Education
Copyright © 2005, Australasian Association for Engineering Education”*

The second part is the circuit design including a complete schematic for the microcontroller, I/O devices such as switches, keypad, motors, and LEDs and power supply. The third part is the development of the program structure diagram, and the fourth part is C code development and debugging. The features of the mega application board are used extensively to allow testing of the real life design.

Resource Development

Specialist training boards have been designed and built in house to support this teaching approach. Previously the course and training boards were based around the Motorola 68HC11. However it was difficult to incorporate into student projects. The problems were relative difficulty in programming as the 68HC11 does not support flash program memory, instead it uses either EPROM (expensive), or EEPROM; the 68HC11 cannot be programmed whilst on the target board requiring the use of a separate programmer; and the 68HC11 has a limited family of devices. The Motorola 68HC11 microcontroller board and the I/O boards are shown in figure 1.

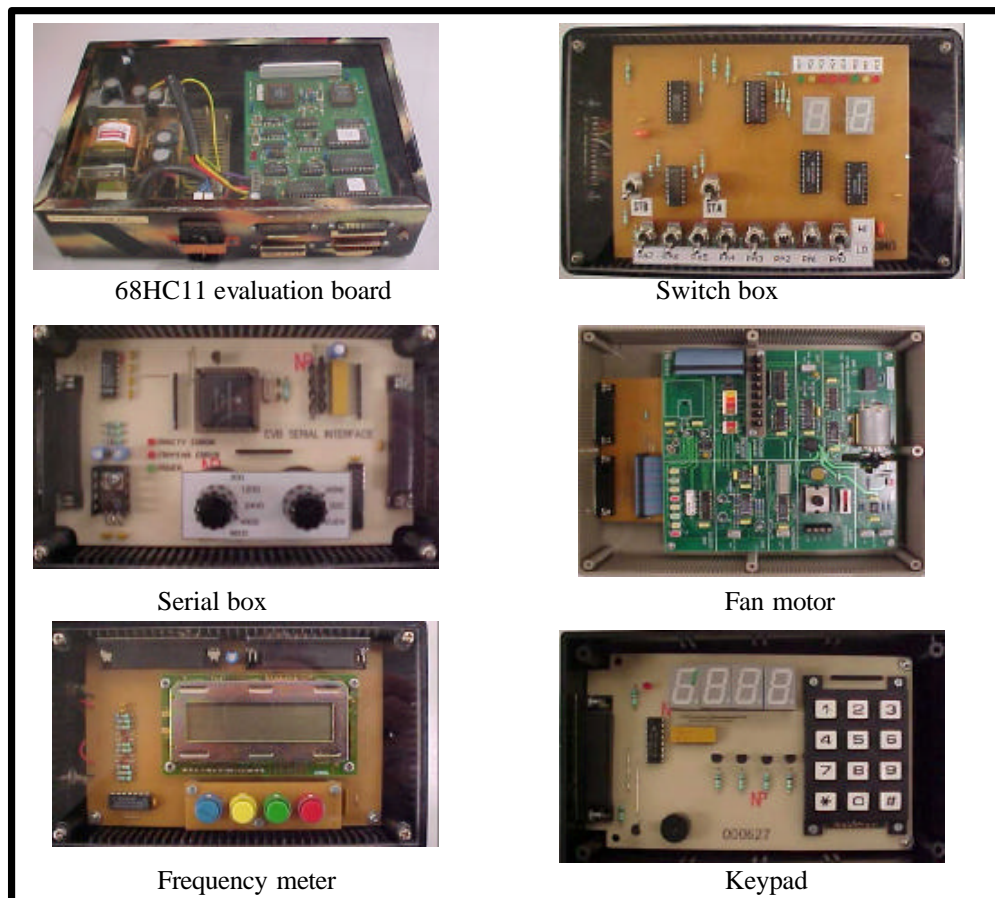


Figure 1 68HC11 evaluation board and peripheral I/O boards

Due to the limitations previously discussed the 68HC11 microcontroller board has several restrictions; the board requires a monitor program to download and debug the C or assembly code which is clumsy and would not be used in industry, as well the monitor program is not suitable for students who wish to create their own microcontroller based project; The peripheral I/O boards were developed over a period of years as stand alone experiments and because these boards could not all interface to the microcontroller at one time they could not be combined to make more complex projects. This also increased the overall cost of the laboratory equipment as we have twenty boards for parallel I/O, 20 boards for serial I/O, 20 boards for the timer systems etc. Thirdly an ICE for the 68HC11 is expensive making class sets impractical, which means that they cannot be used for debugging projects.

The laboratory and course was upgraded from the Motorola 68HC11 to the AVR Atmel ATmega128 in 2000 (figure 2). The ATMEL AVR processor range was chosen based on several factors: the on-board devices in the ATmega128 are similar to that of the 68HC11, the AVR has a large supporting family of devices, ranging from the 8 pin Tiny11 to the 100 pin ATmega2560, the ATMEL AVR uses flash memory which can be programmed in system (In System Programming ISP), a cheap C compiler (CodeVision) was available and the JTAG ICE was economical enough to buy in quantity (figure 2).

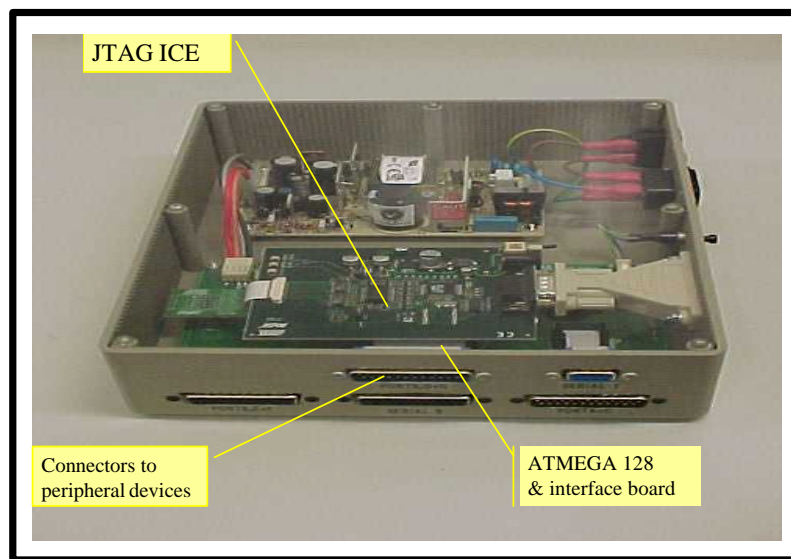


Figure 2 ATmega128 Microcontroller Board

The ability to program flash straight from a PC means that students in project design papers can easily use microcontrollers to build simple projects, and program the devices quickly. The common devices that we stock for student projects are the ATtiny2313, ATmega8, ATmega16 and ATmega128. The compiler used is the CodeVision C compiler which is inexpensive, but more significantly has a free demo version that is limited only by the maximum file size. Thus the student can develop many of their programs at home before debugging them at the university. The JTAG ICE is around \$NZ600, which is cheap enough to include as part of the board design. It allows all internal registers and memory to be

*“Proceedings of the 2005 ASEE/AaeE 4th Global Colloquium on Engineering Education
Copyright © 2005, Australasian Association for Engineering Education”*

viewed, allows breakpoints and has single stepping. Spare ICEs are available for students to debug other projects.

The onboard devices that are programmed in the projects include parallel I/O lines, the UART for simple serial communication, the analogue to digital converters, the timer s for square-wave and PWM generation, and for frequency measurement and the SPI for high-speed communications between other circuit devices such as a real clock chip.

The alternatives to this processor were the 8 bit PIC and 8051 families or to redesign the course around a more powerful 32-bit processor [5]. At the time these were not suitable for simple embedded projects because of their cost and high pin count. However the cost has now come down, with ATMEL's AT91SAM7S32, sub-\$3 flash ARM processor [6]. This processor may be incorporated into the papers in the future.

Mega Application Board

This board is designed to give a wide range of simple projects facilitate student learning. A list of the I/O device is detailed in table 1 and a picture is shown in figure 3

I/O device	I/O system	I/O device	I/O system
8 x Toggle Switches	Parallel I/O	DC Fan Motor	PWM
8 x Push Buttons	Parallel I/O	Lamp	PWM
3 x 4 Keypad	Parallel I/O	Speaker	PWM
Switches/Buttons/Keypad	Parallel I/O	Opto-coupler	Input capture
8 x LEDs	Parallel I/O	BNC (frequency input)	Input capture
2 x 7 Segment Display	Parallel I/O	Temperature Sensor	ADC
Motor Direction	Parallel I/O	Light Sensor (LDR)	ADC
Piezo Buzzer	Parallel I/O	Variable resistor 1	ADC
Motion Sensor	Parallel I/O	Variable resistor 2	ADC
Heater Resistor	Parallel I/O	Humidity Sensor	ADC
LCD Display (24 x 2)	Serial	Microphone with AGC	ADC
Real Time Clock	Serial (SPI)		

Table 1 I/O Device Summary for Microcontroller Applications Board

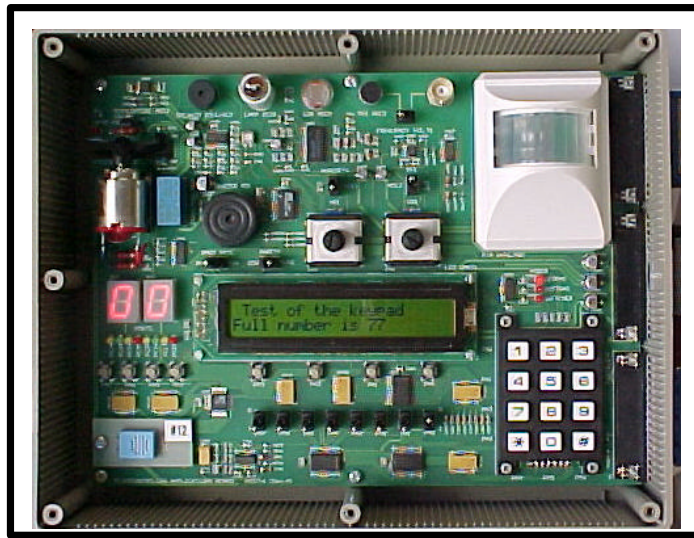


Figure 3 Mega Application Board

The features of the mega application board are

- 1) **Toggle / Momentary/keypad Switches** These three inputs are multiplexed onto one port and are switched by an 8x1 decoder. The toggle switches and LEDs can be used for simple I/O, the momentary switches positioned under the LCD display can be used for menu selection, and the keypad can be used for numbers for security or preset settings.
- 2) **LCD display** is connected to a serial port. This gives the student a simple interface to the display and to use the serial I/O. The baud rate is switch able between 9600 and 56k baud, and odd or no parity. The LCD allows the student to display readings from on board devices such as the ADC or timer systems.
- 3) **Transducers** for the ADC including a light, temperature and humidity sensor, two variable resistors and a microphone.
- 4) **Motor/Opto-coupler/Light/Speaker** connected to the timer systems to allow for speed sensing and control of the motor, dimming of the light and generation of tones to the speaker.
- 5) **Miscellaneous I/O** such as a motion sensor, signal input for frequency measurement and a buzzer.

Examples of small microcontroller systems projects

These one or more of these smaller projects are completed within a week.

- Bilge pump, using the switches as inputs for water level sensors and a LED to indicate that the bilge pump is on.
- Car headlight on warning, using switches to represent the ignition, car door and headlight switches and the beeper to warn the motorist
- Serial LCD driver, using the serial I/O with options for varying the baud rate and parity.

*“Proceedings of the 2005 ASEE/AaeE 4th Global Colloquium on Engineering Education
Copyright © 2005, Australasian Association for Engineering Education”*

- Electronic dice using the 8-bit timer to generate a random number between 1 and 6. A switch triggers the dice, and the number is displayed on the hexadecimal seven-segment display. The numbers 'roll' for three seconds before coming to a stop.
- Clap switch to turn on or off a light. This uses the microphone connected to the ADC to detect the clap, sampling and averaging must be used to detect the clap above background noise.
- Automatic light switch using the LDR to sense the light level and turn the light on when dark.
- Temperature control using the keypad to set the temperature, the fan motor to cool a resistor and the LCD to display the set and actual temperatures.
- Variable speed motor using a variable resistor to set the speed and PWM to control the motor.
- Tone generator uses the keypad to select a frequency and the timer systems to generate a variable square wave.

Examples of larger embedded systems projects

Each section of these larger projects, functional/technical description, circuit diagram, structure diagram and C code takes a week to complete, thus the entire embedded design will take four weeks. The technical description details the type of I/O, how much memory, (program, data and EEPROM), what onboard systems are needed and from this the type of processor required. The type of power supply (battery or mains) and environment are also considered. The circuit diagram will show the processor and supporting hardware, such as crystal and programming interface, (either serial, parallel or JTAG). It will show the I/O and how it is interfaced to the microcontroller and the power supply.

- **House Alarm with keypad entry.**
An operators manual is produced, describing the features and operation, for example. how to activate and deactivate the alarm, how to alter the pin access code and change entry and exit delays. The program can be tested using the mega application boards keypad for PIN and timing settings, switches for door sensors, motion detector, beeper for access and siren.
- **Clock Radio.**
The manual will detail how the day and alarm time is set, how the snooze will operate and how the alarm and radio is turned on and off. The program can be tested using the real time clock IC to keep the time, the momentary and toggle buttons for the clock functions, a LED to indicate that the radio is on, the beeper for the alarm and the LCD to display the time.
- **Frequency Meter:**
The manual will detail how the menu system is used to select the type of measurement to be performed, and what error messages will be displayed. The program is tested by connecting a signal source to the BNC connector and using the LCD with the momentary buttons mounted below the display to select the menu options. The period, pulse width and frequency of the signal can then be displayed.
- **Data Logger;**
The temperature, humidity and light level along with the time and date are recorded at one minute time intervals. This data is stored in EEPROM so that the data will not be lost if power is disconnected. A PC can be connected to the RS232 interface and when

the appropriate command is received the data will be uploaded from the data logger to the PC.

Conclusions

Replacing the Motorola 68HC11 with the AVR ATMEL has allowed an ICE to be used in the microcontroller boards. This has the advantage of allowing the student greater debugging features to quickly fault find their code. The development of structure diagram generator software application has encouraged students to begin their software design with a structure diagram rather than with code. The introduction of this microcontroller with its ISP flash memory and the wide family range has allowed students to more easily implement microcontrollers into their projects. Combining all of the separate peripheral I/O boards into one Mega application board enables both simple and complex projects to be completed within a restricted timeframe. The resulting embedded systems course is a highly practical paper allowing the students to perform real life designs.

References

- [1] Turpin, R. H. Top-Down Instruction, Bottom-up Learning in a Microcontroller Lab. Proc ASEE Annual Conf. 2004
- [2] Schoch P. M, Golwelkar A, Lim L, Lewis D, J Kokernak J. Laboratory Introduction to Embedded Control. Proc ASEE Annual Conf. 2004
- [3] Smit A, Heer D, Traylor R, Fiez T.S. A Custom Microcontroller System used as a platform for learning in ECE. Proc ASEE Annual Conf. 2004
- [4] ISO/IEC 8631:1989. Information technology - Program constructs and conventions for their representation. (1989)
- [5] Caceres T. A, Combs Z, Ochoa J.A, An Advanced Microcontroller Systems Course for Upper-Level Undergraduate Curriculum. Proc ASEE Annual Conf. 2002
- [6] Online: http://www.atmel.com/dyn/corporate/view_detail.asp?ref=&FileName=IARDevkit_3_7.html&SEC_NAME=Product

Bibliography

Mark Beckerleg

Mark Beckerleg received a BE (Hons) degree from the Auckland University in 1988. He is currently a senior lecturer in the school of engineering at the Auckland University of Technology. Previous experience includes engineer for Television New Zealand and an embedded systems design consultant. He is currently studying for his PhD

Dr John Collins

Dr John Collins received the BSc(Hons) degree in physics and mathematics and the PhD degree in electronic engineering from the University of Auckland. He is currently a senior lecturer in the School of Engineering at the Auckland University of Technology. He has industrial experience in the development of embedded systems and has worked as a consultant on a number of embedded systems projects. His research interests include the design and verification of embedded systems, and embedded systems education.

*“Proceedings of the 2005 ASEE/AaeE 4th Global Colloquium on Engineering Education
Copyright © 2005, Australasian Association for Engineering Education”*