MDP Balloon Board: An open source software & hardware system for education (Work In Progress)

P.R.A. Fidler, T.G. Froggatt, M.J. Morley, A.J. Green, E.J. Greveson and P.J.G. Long
Cambridge University Engineering Department
Trumpington Street, Cambridge CB2 1PZ, United Kingdom
{praf1,tgf20,mjm80,ajg72,ejg45,pjgl}@cam.ac.uk

Abstract

As part of a Cambridge-Massachusetts Institute [1] funded project a set of teaching resources is being developed around the ARM based Balloon Board [2] microcomputer. The system is being designed to give students and academics a preinstalled system with support material and low cost hardware for control and data acquisition projects. Aspects of the hardware and teaching resource are discussed, together with experiences gained in configuring the system for use with a design and build autonomous robot project for second year undergraduates. Extensions to the current system are also presented including work extending the system using RTAI, additional low cost hardware and inclusion in the MDP Resource CD (Multidisciplinary Design Project) which is also being developed,(www-mdp.eng.cam.ac.uk)

1 Introduction

The teaching and running of cross-discipline undergraduate 'design & build' projects is often restricted by the lack of suitable hardware and supporting software and teaching material. The complexity of such projects and the restrictions on budgets and course time often result in bespoke solutions at individual educational establishments.

The Multidisciplinary Design Project (MDP), funded by a UK government initiative the Cambridge-MIT Institute (CMI), is an attempt to collate and develop best practice for 'design & build' projects. The plans include the development of a design information resource, a documented set of prototype materials/components together with associated teaching material. The requirement for computer data acquisition and control is central to these types of projects. In many situations the reqirement to start at a relatively low level, combined with budgetary and support constraints, has resulted in the use of PIC (inc. BasicStamp, AXE) systems. Projects requiring a little more performance are often based around an external unit connected to a PC using a standard interface e.g Serial/Parallel and more recently USB/Ethernet. High performance systems tend to be derived from industrial systems plugin cards or bus systems, e.g. PC104. The resulting systems are very capable but are costly and often specific to a particular software package and OS.

Attempts to reuse lower cost hardware, e.g. PDAs, as controllers has lead to to some intriguing concepts but are constrained by a major requirement for education - sustainability. The availability of replacement units, repairs and support is critical given the relatively long lifetimes of projects in Universities.

Although the hardware availability often drives system selection the operating system and available software are also critical in teaching. The time required to master a new operating system or particular control language can significantly reduce the learning experience and outcomes. At another extreme the use of a 'Windows like' point and click graphical environment can leave the students with a lack of understanding of the time critical nature of the software and associated hardware.

A review of the system required to support the plans of the MDP together with extensive discussions with other universities and faculties lead to a proposal for an 'ideal' university control and data acquisition teaching system.

Common processor
(e.g. PC 104, PIC, ARM, Industrial relevance)
Flexible
Expandable
Low Cost
Reasonable performance
sufficient for low→ medium image processing
Common OS
Multiple Languages
Development environment
Multiple interfaces
(e.g parallel, serial, I^2C [3], CAN)
FPGA
(for Direct programming and bespoke interfaces)
Low Power
(Safety/battery operation)
Sustainable
Support resources

TABLE 1: *A specification for an 'ideal' University control and data acquisition system*

A number of existing systems were considered as the base system for the MDP but most had major constraints often relating to the hardware and software costs/student. During the investigation the team became interested in the 'Tower'[4] computer system being developed in the Media laboratories at MIT. This offered a level of scalability combined with low cost. Discussions were held to extend the functionality, especially to support enhanced processing capabilities, but unfortunately the timescales were unachievable. However, discussions with the Balloon board development team lead to an interest on both sides to become involved in the development of a similar system.

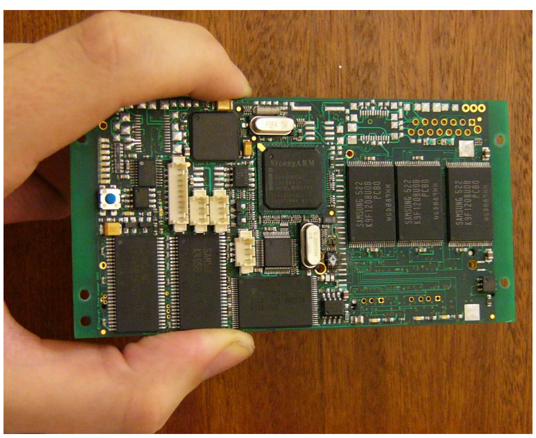

FIGURE 1: *Existing Balloon Board (ver. 2.05g)*

2 Balloon Board

The existing Balloon system was designed in Cambridge (UK) by Steve Wiseman in collaboration with a number of other individuals and local companies under an Open Hardware Licence. It measures 56 x 110mm (including optional mounting holes) and is based around a 200MHz StrongARM . The operating system is a derivative of Familiar Linux [5] and is supported. Although the boards are available commercially via Aleph1 [6] the gerbers (board layout files) and bill of materials etc are openly available for individuals/institutions to source their own boards. In fact, a number of companies make their own variants often sharing production runs to minimise costs.

Processor	Intel StrongARM SA-1110 running at 200MHz
Operating System	Currently Ships with Linux 2.4.19 kernel installed (with. 2.4.25 and 2.6.8.1 kernels also available)
Memory	64MB RAM 192MB NAND Flash memory 8 MB NOR Flash (for a bootloader)
Interfaces	200 way data/address buses CompactFlash socket 3x Serial ports USB Host and Slave ports I^2C bus Audio
Features	Low power (typically less than 1 Watt)

TABLE 2: *Balloon 2 Specification*

The open nature of the system and the use of Linux was an major attraction to the MDP, as the open source resource pack[7] being assembled as part of the project is based on a hybrid Knoppix/Debian Linux distribution. In addition, an enhanced Balloon board was planned offering the opportunity to further extend the functionality and expandability at relatively low cost.

3 MDP Balloon (beta)

3.1 Hardware

Combining the concept of the 'Tower' computer and the Balloon 2 the MDP has developed a system that uses 9-way 'D' plugs as a cheaper and more robust alternative to the 2x20way IDC connectors used in the 'Tower'. The connection supplies +5v, +supply ± 15v + I^2C clock and data lines.

The initial test bed for the hardware has been the 2nd year robotics course run at the Cambridge University Engineering Department (CUED) in which teams of six students are give 24 timetabled hours over a 4 week period to design and build an autonomous robot. The course is run 4 times a year with each with a different task related to a typical factory operation such as picking a item, sensing one or more physical parameters and then transporting the item to a prescribed location. The original system used a bespoke 80522 based micro-controller with student interfaces being controlled via an I^{2}C bus. The requirement to match the capabilities of the old system has resulted in the standard motor control board, line following and prototype boards described below.

The flexibility of the system and the relative low cost of single board development has led to a number of additional boards being developed for general use and possible inclusion in the robotics course.

- Power supply board + 8 bit I/O
  - 6-16 v in
  - 5v, 12v, ± 15v out
  - USB/Serial/I^{2}C breakout
  - 16 channels dig I/O
  - 8 channels 8 bit ADC, 2 channels 8bit DAC
- Motor control board
  - 4 channel 10 bit ADC
  - 4 channel PWM controllers
- LCD Display Board
- Prototyping Board
- Optical line following
- Prototyping Board (with I^{2}C programmable PIC)*
- High Speed Analogue Board*
- Inductive 'wire' following*

* Still under development

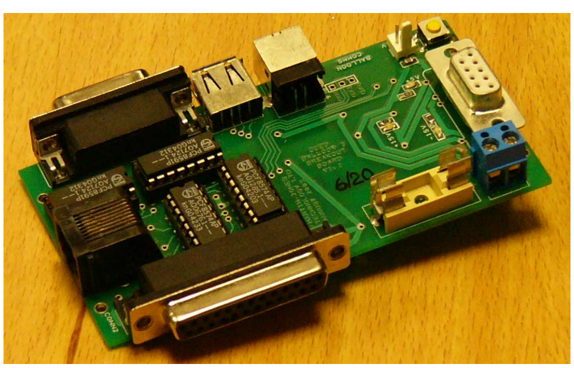

FIGURE 2: *Power Supply and 8 bit analogue/digital I/O board*

All the design details are open and have been developed using the gEDA[8] electronics design suite. Wherever possible 1 or 2 sided boards with no plated through holes have been designed to enable low cost production techniques.

The I^{2}C bus has limited bandwidth and the system also allows access to the higher speed serial and parallel buses via FFC cables. One board using the 16bit 'SAMOSA' bus is a 8 channel 100ks/sec/channel ADC board with anti-aliasing filters.

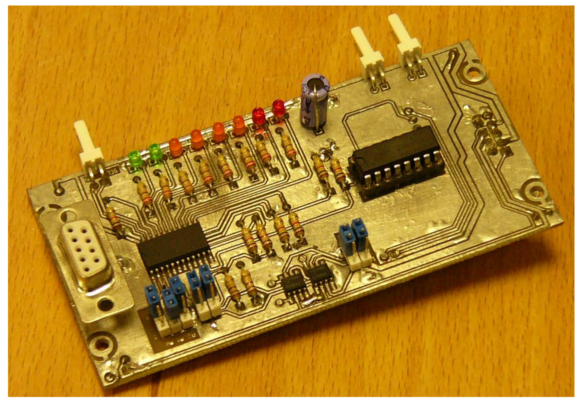

FIGURE 3: *Example of a bespoke board, laid out in gEDA/PCB. (Digital I/O + PWM motor controllers)*

3.2 Software

The MDP Balloon stacking system uses the I^{2}C Bus lines on the D-Plugs to control compatible devices on the upper layer boards. A number of I^{2}C devices are available: digital I/O, analogue I/O, PWM controllers, LCD panels etc. Controlling these from the Balloon board requires control of the I^{2}C bus.

Linux provides support for the I^{2}C Bus as part of the kernel, with user-space support provided via device nodes. A number of different I^{2}C bus controllers are supported, and there is also a 'bit-banging' driver capable of using any two available digital I/O lines as an I^{2}C bus. A server program t has been developed to provide convenient access to devices connected to the I^{2}C bus on the Balloon board. Programs that use the bus may be run either locally on the Balloon, or remotely via a network connection. An extensible support library provides a low-level C and C++ interface to the bus, and higher-level interfaces for the individual I^{2}C devices used in the MDP

Balloon system. The I^2C server has also been run on x86 desktop system connected to a proprietary USB→I^2C adaptor.

Additionally, software has been developed to emulate the interface provided by the the 80522-based micro-controller used for the second-year robotics course that is taught at the department.

A key requirement for use of the MDP Balloon system in education is a development environment. There is no off-the-shelf development environment for the Balloon board at present. CUED currently supply all of its students with the Anjuta IDE on the MDP Resource CD and the possibility of customising Anjuta to provide the additional functionality necessary to compile, upload and run code on the Balloon board is now being considered.

For less time-critical tasks a platform neutral scripting language may be sufficient. The possibility of using Brandy (a BBC Basic interpreter) for simple tasks, or Octave (an open source Matlab clone) for complex mathematical analysis haas also be investigated.

4 Real Time Software

To date the MDP Balloon (beta) systems have successfully been used for the control of simple robots. There is, however, significant potential for extending its use in experiments requiring high levels of control and project involving remote or battery powered data acquisition.

In order for the MDP Balloon system to be suitable for the for such applications, a degree of determinism is required beyond that provided by a stock Linux kernel. A search revealed a number of possible solutions, with RTLinux and RTAI being the two most prominent [9]. Requiring a completely open-source solution, and preferably one with an active developer community, it was decided to pursue RTAI as the potential candidate.

The installation and configuration of RTAI for a desktop PC proved relatively simple. However the same was not true for the Balloon board:

Difficulties encountered:

- The Linux kernel must be patched to incorporate the Adeos nano-kernel/interrupt pipeline system. The closest Adeos patch for a Linux 2.6 kernel was for the XScale PXA processor. The patch unfortunately does not apply cleanly to the 2.6.8.1-tcl kernel that the MDP uses on the Balloon.

- Support for the StrongARM appeared to have been removed from RTAI starting with version 3.2 (Vulcano).

- The user-space side of the liblxrt library (on the ARM) used functions only available in the kernel (eg the atomic compare and exchange operations).

Current progress:

- Adeos has been patched into the Linux 2.6.8.1-tcl1 kernel. This was achieved by using the pxa255_2.6.7-bk6-karo.patch in RTAI source tree as a starting point, and then adding support for the SA-11x0 processor in addition to the PXA.

- RTAI 3.2. can now be compiled and the support for SA-11x0, that had been removed from the official source tree, has been reinstated.

- Two out of the six tests in the RTAI test suite now work (the kernel-space preemt_rt.ko and switches_rt.ko tests), see tables 3 - 5 for sample results.

min	max	average	fastjit	slowjit
-12	15	0	37	50
-21	24	0	39	50
-41	43	0	58	59
-41	43	0	58	59
-41	43	0	58	59
-41	43	0	58	59

TABLE 3: *Results of running RTAI preempt_rt.ko test with no load*

min	max	average	fastjit	slowjit
-73	87	0	80	66
-73	87	0	80	66
-75	87	0	88	72
-78	87	0	88	72
-78	87	0	88	72
-78	87	0	88	72
-78	87	0	88	72

TABLE 4: *Results of running RTAI preempt_rt.ko test with under load*

```
Wait for it ...

FOR 30 TASKS: TIME 394 (ms),
SUSP/RES SWITCHES 120000,
SWITCH TIME 3281 (ns)

FOR 30 TASKS: TIME 414 (ms),
SEM SIG/WAIT SWITCHES 120000,
SWITCH TIME 3447 (ns)
```

TABLE 5: *Results of running RTAI switches_rt.ko test*

It is hoped that if a working real-time enabled system can be provided for the Balloon as part of the development tools that this will greatly increase the potential user base for the board.

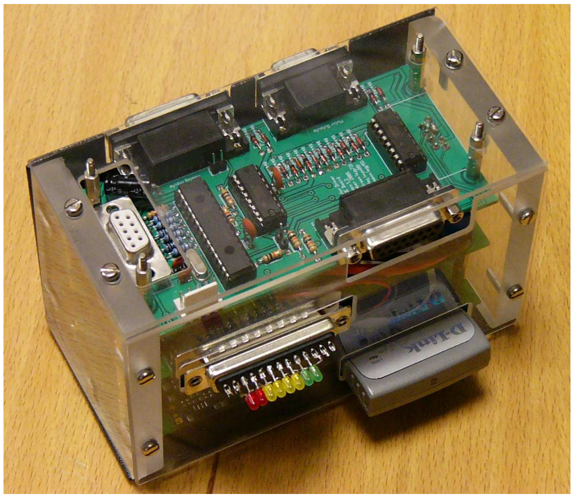

FIGURE 4: *Balloon 2 based 'stack'*

5 Balloon Board (ver. 3)

The Balloon 3 board has been designed as a more powerful slot-in replacement for the existing users while offering enhanced facilities.The current specification includes:

Processor	XScale PXA270 processor (max: 520Mhz)
Memory	128Mb Mobile SDRAM (max: 512MB)
	NOR flash ROM (max: 32MB)
	NAND-flash, including 16bit variants
	Maximum of 2GB
FPGA/CPLD	FPGA - 400k gates
	or CPLD with 256 macrocells
Power	Single 5V Supply
	<1W (typ.)
weight	20-30g
Interfaces	LCD (STN & TFT displays)
	Serialx3, Bluetooth, I^2C
	USB host, slave and OTG
	Compact Flash
	~4MHz 8/12 bit data/address bus
	Camera (Quick Capture)

TABLE 6: *Balloon 3 Specification*

6 Conclusions & Future Developments

A flexible computer system, based on the open design Balloon 2, has been developed for use in real time control experiments and control projects in Universities. The system has been demonstrated for high latency tolerant systems and work is underway to port RTAI to the system. To assist in the use of the associated interface system a new $I^2 2$ deamon has been written that allows the boards to run from both the Balloon or a x86 system via a USB→I^2C or serial→I^2C interface. In addition, by using the MDP Resource CD, an open source working environment is being developed offering a range of software from interface design through to an IDE for the included high and low level languages.

It is hoped to have pre-production versions of the faster, Xscale based, Balloon 3 by the end of 2005 and that the faster processor will assist in decreasing some of the timing results seen as well significantly improving the real-time computational abilities of the system. Work will continue on porting RTAI to the new PXA based system and the results will be released. It is also planned to develop a series of educational resources to aid the inclusion of the system in a undergraduate engineering course.

References

[1] http://www.cambridge-mit.org

[2] http://balloonboard.org

[3] Philips Semiconductors,2000, The I^2C Bus Specification v2.1

[4] http://gig.media.mit.edu/projects/tower/

[5] http://familiar.handhelds.org

[6] http://www.aleph1.co.uk

[7] http://www-mdp.eng.cam.ac.uk

[8] http://www.geda.seul.org

[9] Aarno D., Evaluation of Real-Time Linux Derivatives for Use in Robotic Control, 2004. http://www.e.kth.se/ e98_aar/resources/rtos.pdf