The PLEB Project

A Platform for Portable and Embedded Systems Research

Adam Wiggins

UNSW-CSE-TR-0105 — September 21, 2001

awiggins@cse.unsw.edu.au
http://www.cse.unsw.edu.au/~disy/
School of Computer Science and Engineering
The University of New South Wales
Sydney 2052, Australia

Abstract

The PLEB project is a student run project aimed at stimulating portable & embedded systems research within the school. This report outlines the projects activities and some of the experiences gained in developing the first hardware platforms. The report also sketches the details for second generation PLEB hardware platforms and the project's future direction.

Contents

1 Project Overview

1.1 Introduction

In July 1998 a proposal was put forward to design and build a "pocket Linux embedded box" (PLEB) as a School sponsored student project at CSE. The proposal was accepted and a budget allocated, hence the PLEB Project was born.

The original motivation for PLEB was two fold. On the one hand a growing interest in portable computing and embedded hardware within the school was creating a demand for a cheap platform suitable for use in researching these new fields; on the other hand I was interested in getting my hands dirty with some hardware design while getting both staff and students interested in hardware research and development through the project.

Since its initiation the project has grown and matured into a number of components with the overall goal of "building an infrastructure (both hardware & software) to aid researchers/developers in the area of portable/embedded systems design"[1]

This report gives an introduction to the PLEB project and focuses on the experiences gained in designing the hardware components of the project.

1.2 Project Organisation

As the PLEB project grew it became clear what components were required to achieve the project's goals. The components of PLEB are grouped into three categories and listed below:

Hardware Platforms

> Each sub-project in this category makes up a number of hardware designs targeting a specific application environment.

Systems Software

> These sub-projects make up the software base that required for applications to make use of PLEB hardware platforms. In many cases they target platforms outside of the project as well.

Development Tools

> A number of tools used in the construction of the software and hardware components of PLEB, these sub-projects look at selection and development of the construction tools used in the project.

Since the PLEB aims to be an aid to further research and development it has from its beginnings embraced the open source philosophy. All software components of the project are released under the GNU General Public Licence (GPL) hardware components are released under GPL like hardware licences and as much as possible open licensed tools are used and developed.

1.3 Hardware Platforms

The experience with designing and constructing PLEB's hardware platforms forms the focus of this report. The two platforms that currently form part of the project are briefly introduced here.

1.3.1 Photon

The original platform of the project Photon forms the main development focus of the project. A cheap, small, light-weight platform capable of delivering a serious amount of processing power based on Intel's StrongARM range of processors.

[1] As paraphrased from 'the PLEB Project mission statement'.

1.3.2 Nova

The Nova platform developed out of the need for a small programmable I/O device connected to a standard desktop/laptop computer or Photon platform as a testing/programming tool for hardware platform development.

1.4 Systems Software

The software infrastructure required for PLEB basically includes all programs that are required to service the applications that will run on the projects, as well as similar hardware. This basically amounts to operating system and BIOS-like firmware development.

1.4.1 Linux Support

The Linux kernel provides a full UNIX clone and more and is a popular operating system kernel even in embedded application areas. One of the main advantages of Linux is the large developer and support base. An aim of PLEB is to port Linux to the project's hardware platforms where suitable.

Currently all Photon platform hardware support has been merged into the ARM port of Linux and is readily available.

1.4.2 Catapult

Catapult is the firmware component of PLEB. The firmware is responsible for booting a hardware platform into a stable known state and then launching the operating system and/or applications of the system. The basis behind Catapult is to provide an extensible and modularised, multi-platform firmware for all hardware used in the project

In the summer of 1999/2000 Simon Winwood under my direction developed a design for Catapult, the following summer Shane Stevens and Harvey Tuch implemented the base as well as modules for loading/booting Linux from FLASH memory.

1.4.3 The L4 Microkernel

Systems research, in particular operating systems research, is one of the strong areas in CSE. The L4 Microkernel forms the basis of most of the operating systems research in the School and my undergraduate thesis outlined the design of a L4 Microkernel for the Photon platform. I also worked on the the L4Ka implementation from Karlsruhe. While neither kernel is functional on PLEB platforms, recent progress with L4Ka should provide a working microkernel for Photon.

1.5 Development Tools

1.5.1 Anvil

The IEEE 1149.1 JTAG Standard (see section 3.3) is a common feature of many modern embedded systems. It facilitates testing and programming of the systems hardware platform. Anvil is a project to develop a JTAG toolkit for utilising the JTAG support of embedded systems to test/program them.

Harvey Tuch will shortly be joining me working on Anvil.

1.5.2 Compilers, etc

All software components of the project are constructed using the standard GNU development tools, gcc, binutils etc.

1.5.3 Schematic/PCB CAD Packages

Computer aided design (CAD) packages have formed one of the major weaknesses of the project. CAD software has until recently had no open source replacements. The GNU Electronic Design Automation project (gEDA) has started in recent years, however while it offers a mature schematic capture package, a suitable printed circuit board (PCB) design package is still not available.

All PLEB hardware designs to date have used the OrCAD software package but experience (see section 3.1) has highlighted the need for an alternative package to be employed.

2 Photon Platform

Photon, initially know as MiniPLEB is the first hardware platform of the PLEB project. The aim was to create a base platform on which the project could grow from. However this first attempt proved more a learning experience in how to turn a design idea into a usable platform.

In this section the goals and design of the first version of the Photon platform are examined. The Photon platform has found use in a number of projects within the School and even within a few institutes outside of the university.

2.1 Goals and Constraints

The ultimate goal of the Photon platform was to provide a building block on top of which a portable system could be constructed. A side issue which ended up being a significant factor was that Photon was the first serious hardware design that I had taken from concept to physical construction. As a result the goals and constraints of the platform were hazy until the point that it was being used in other groups projects.

To understand how the design ended up the way it is the constraints that I was working with are explained below.

2.1.1 The SA1100 StrongARM Processor

The core of the Photon platform is the SA1100 StrongARM Processor, originally developed by Digital Equipment Corporation. The StrongARM range was taken over by Intel when Digital went out of business.

The SA1100 was selected because of its extremely low power consumption and on chip peripherals for glue less memory interfacing, serial communication, etc.

2.1.2 Physical Size and Power Consumption

The guiding constraints of the original Photon platform was to make the board as small and as low power as possible. The aim was a credit-card footprint while not worrying too much about the height.

2.1.3 Standard Interfaces

To keep the time and cost of development down the aim of Photon was to provide standard interfaces as much as possible, so that “off the shelf” peripherals could be connected to the platform to provide extra functionality.

2.1.4 Component Selection

The primary focus of Photon’s original component selection was their physical size and power consumption. The design was essentially completed before any other considerations of the components were taken into account. This proved a mistake as the cost and availability of components was not taken into account.

2.1.5 Printed Circuit Boards

What primarily shaped Photon into what it is was the manufacturing restrictions of the platforms PCBs. The original restriction was that of a 4 layer board. Since the Photon consists of a modern processor utilising high speed signals, the two inner layers of the PCB had to be dedicated to the power supply rails (3.3 volts and ground) of the board leaving only the top and bottom layers for signal routing. Since the top and bottom layers are also the layers that the components are placed on, this provides limited area for routing signals.

Another issue relating to the manufacture of the Photon PCBs was that the cost of manufacturing them in low quantities is essentially the setup cost which inflated the apparent per-unit cost of the platform.

2.2 Design Overview

The main goal behind Photon as mentioned earlier was to provide a base system which could be built upon and customised to particular applications quickly and cheaply. This, in combination with the manufacturing constraints, led to the idea of stacking multiple simple boards to produce prototype systems in which hardware debugging and software development could take place cheaply and quickly. This stacking idea forms the basis of the Photon platform, if the stack becomes unmanageable, due to things like signal degradation or drift of the power rails, two or more boards can be merged into a single, more complicated and costly board. Similarly if size becomes a factor boards can be merged.

2.3 CPU Board Design

The CPU board, which forms the base of a Photon system, simply contains the processor, minimal support for booting and supplying the processor with power, as well as the ports and connectors required for communicating with outside world or expanding the functionality of the board.

One of the critical design choices was to not put any memory on the CPU board. The reasoning behind this basically was not to lock down the choice of memory parts used for all systems. Another major factor contributing towards this choice was to make the CPU board layout very simple. With a 4-layer board the credit card outline of Photon could not be maintained if memory chips had been put on the CPU board, however stacking of two 4-layer boards facilitated fairly simple routing while providing a complete system with processor and memory. So the original design was split into a processor board and a memory board.

2.3.1 Power Supply

The power supply constitutes the weakest point of the Photon platform. The design aimed simply at providing the processor with its power rails while keeping the physical size of the supply area of the CPU board to a minimum.

The choice was made to not lock down the power supply to any particular power source so that CPU board could be adapted to a large range of environments. The main power rail can be supplied from a power connector as well as the daughter board or LCD/general I/O connectors. To minimise components, no diode protection/isolation was employed so care has to be taken that only one connector will supply the rail with a voltage in the appropriate range and correct polarity. If a battery supply with integrated charger is needed it can either be provided stand alone or as part of a daughter board.

The power supply is made up of 3 parts:

- A reset circuit is employed to make sure the CPU comes up properly even if the supply is a bit temperamental when I comes up.
- A 3.3 volt supply rail. This supplies power to all components of the CPU board as well as to the daughter board and LCD/general I/O connectors. Up to an amp can be drawn from the supply itself.

- A 1.5 volt supply rail. This supplies the core voltage to the processor.

Integrated into the power supply design are a pair of voltage comparators to report the status of the power source rail. The first trip value indicates a low power rail to a general purpose I/O (GPIO) pin of the processor while the second trip value indicates the power source rail is too low for the processor to operate. If the second trip is crossed, a fault is signalled to the processor which forces it to sleep, conserving energy while preserving the state of the processor and memory, until the supply rail restored to a usable voltage.

One of the flaws of the design is that while the 3.3 volt rail will operate within required parameters with a supply rail voltage down to 3.0 volts the 1.5 volt rail fails once the supply rail voltage drops below 4.0 volts. This makes the power supply design unsuitable for a lot of small and simple battery configurations.

In the original version of Photon a LCD bias voltage supply was provided. This circuit was however dropped because the bias voltage required varies extensively between LCD panels, if in fact one is required at all.

2.3.2 Serial Communications Ports

The SA1100 processor provides a variety of serial communications controllers. The Photon board provides parts to make use of three of these ports, each aiming at particular application areas. All provide full duplex operation.

RS-232 Serial Port

RS-232 is the standard serial port interface found on nearly all desktop and portable machines. It allows the Photon to talk to nearly any other computer, albeit at a somewhat low speed.

The RS-232 port can be configured as either a single serial port with RTS/CTS handshaking signals (controlled by software) or as two independent serial ports without any hardware handshaking.

Both ports support speeds of up to 250Kb/s and the hardware can force the ports to shut down, to save power, as well as detect if a valid RS-232 device is connected.

RS-422 Serial Port

RS-422 is another serial port standard which is commonly used. It differs from RS-232 in two key areas. Firstly RS-422 provides differential pairs for transmitting/receiving. This allows lower transmission voltages to be used and also provides better noise immunity allowing longer cables to be used in noises environments.

The second difference is that the encoding scheme used is packet based, allowing addressing of particular units and CRC checks in hardware. The line driver used in the Photon allows a maximum of 32 devices to be connected to the one terminated cable, allowing a mini-network to be set up.

The port supports speeds up to 250Kbaud and again the hardware can disable the port to save power.

IrDA v1.0/v1.1

The IrDA standard is used in all infrared systems. Two versions exist, v1.0 which all infrared devices support, allows speeds of up to 115Kb/s. Devices like palm pilots, printers, etc typically support this. The newer v1.1 uses a different encoding method and supports speeds of up to 4Mb/s. IrDA provides cheap line-of-sight wireless communication at distances of up to a meter with reasonable bandwidth. The Photon platform makes use of a IrDA LED which supports both speeds as well as intensity control to provide 3 levels (and off) to reduce power consumption at closer ranges, etc.

2.3.3 Daughter-Board Connectors

The daughter board connector forms the heart of the Photon design. By providing access to the majority of the SA1100 system bus and peripheral control signals it allows the functionality of a Photon system to be extended by daughter boards which can be stacked. The use of 2 connectors at opposite ends of the board provides structural support as well, so no screws or clips are required to hold the stack together.

This stacking system has allowed simple boards to be individually developed at low cost and in small time frames. Once a number of small single-function daughter boards are available, the hardware bugs are easily located. Software for them can be developed. The user can select a combination of functionality for a particular system, and then merge the daughter card designs into a single daughter board (see section 2.5.2).

The daughter board connector supplies the 3.3 volt rail and can draw on an external power source to provide the power supply rail to the CPU board.

2.3.4 LCD/General I/O Connector

The Photon design makes use of a separate connector which provides the LCD I/O lines from the SA-1100, this connector can be used to drive active or passive LCDs at a colour depth of 1-16 bits at up to 1024 x 1024 pixels. The unused signals of the LCD connector can also double as GPIO lines.

The LCD connector supplies the 3.3 volt rail and can draw on an external power source to provide the power supply rail to the CPU board.

2.4 Memory-Board Design

The memory daughter board of Photon was designed to provide minimal functionality required to get a Linux system running on Photon. The design provides non-volatile FLASH memory as well as volatile DRAM memory to provide file system and RAM respectively. A real time clock was also included to maintain system time across power supply rail outages, but proved to be fairly useless (see section 2.4.3).

2.4.1 DRAM Memory

The original memory board design provided four banks of 16MB each giving 64MB of RAM in total. However with the small size of the daughter board (primarily dictated by the size of the CPU board) it was not possible to fit and route 8 DRAM chips on a 4-layer PCB. This would have given the maximum amount of memory the SA1100 can accommodate without extra logic. As a result only 2 banks of RAM are mounted on the memory daughter board, providing a total of 32MB of RAM.

The DRAM parts supports what is known as low-power self refresh. This allows the SA1100 to “sleep” to save power while preserving the contents of RAM.

Two jumpers select which banks the RAM will occupy. This facilitates the stacking of two memory boards to supply the total complement of 64MB of RAM.

2.4.2 FLASH Memory

The memory board provides non-volatile FLASH memory; each memory board populates a single bank of memory. The current memory boards make use of parts which provide 4MB per bank[2]. As with the DRAM, a jumper selects which bank the FLASH is mapped to. If two memory boards are stacked, a total of 8MB of FLASH is available for booting as well as file system space.

[2] However larger parts providing 8MBs per bank are available.

2.4.3 Real Time Clock and Buffers

As mentioned earlier, originally a real time clock (RTC) and backup battery were included with the design. It was found, however, that problems with the selected RTC chip and battery and the fact that no one made use of the RTC resulted in it being dropped from the manufactured boards.

The memory board also makes use of buffers to reduce the drain on the address and control lines of the processor board in situations where a large number of daughter boards are stacked on a CPU board.

2.5 Other Daughter Board Designs

While the CPU and memory boards form the base Photon platform, a few other boards have been developed. These boards were commissioned to service project needs as they have arisen and are described in the following sections.

2.5.1 Ethernet Board

The availability of ethernet greatly speeds up development time by facilitating fast downloading of data, such as operating-system kernels and RAM based file systems, from host computers. Ethernet also enables use of things like the network file system (NFS), allowing serious development on the currently diskless Photon platform.

For this reason, a stand-alone ethernet daughter board was developed. It is expected that in the future ethernet will from part of a larger and more general development daughter board (see section 3.5).

2.5.2 The “Frog Board”

The first true test of the success of the stackable daughter board idea came with a custom design for a specific research application area. Research being undertaken by Andrew Taylor in monitoring frog populations in the Northern Territory required a small and low-power platform. This platform needed to be equipped with 1-8 audio microphones and capable of running software to detect frog calls, logging the results to non-volatile memory with absolute time/date stamps.

It was decided that a Photon CPU board equipped with a single daughter board providing the below functionality would suit the application:

- 8 analog to digital converter (ADC) channels.
- Backed up real time clock (RTC) in the advent of power outage.
- Battery voltage reading.
- Miscellaneous digital I/O.
- Non-volatile FLASH memory storage, suitable for a small Linux file system.
- RAM storage suitable for running a Linux kernel and applications.

The design process involved two stages:

1. Initially, a photon board equipped with a memory daughter board formed the base platform. A prototype board was designed providing the 8 channel ADC and an Atmel AVR 8bit RISC processor for battery voltage reading and miscellaneous I/O. This board was quickly thrown together and constructed using a 2 layer PCB manufactured within the School.
2. The final stage involved merging the memory and prototype frog boards into a single daughter board for the production system.

The ability to develop prototype boards using existing boards proved highly productive in this situation. A quick prototype could be assembled allowing the application software to be developed while the hardware bugs were ironed out and the final design completed. Hence, once the final production design was available the software development was already well underway.

These units have now been deployed and the extreme environment (high temperatures, unreliable power supply through solar-charged battery cells) provides a good test of the Photon's robustness.

2.5.3 Projects Using Photon

To date the Photon platform has been used in a number of projects within the UNSW as well as a few outside. A few a listed here:

- Frog population monitoring in the Northern Territory.
- A number of CSE thesis projects.
- Bionic eye project at UNSW.
- The Stanford Satellite project.
- The LART project at TU Delft, Netherlands.

While other groups have approached us with proposals to us Photon in their research or applications, problems with making the hardware available (see section 3.4) proved too limiting.

3 Experiences

This section elaborates on some in the design of the Photon platform and highlights paths for future development.

3.1 OrCAD Software

As mentioned in the overview, CAD software tools have been one of the major weaknesses of the project. Why is this so? CAD software has been one of the late bloomers in the open source movement since they are specialised tools used typically in commercial environments. Recently, developers frustrated with the lack of availability of good CAD software have started looking for alternatives, and as a result projects like gEDA have popped up with the aim of developing a complete set of electronic design tools for hardware development. However these tools are still incomplete and buggy.

Initially OrCAD was used for PLEB hardware platform designs. The reason being it was suitable for multi-layered PCB design and integrated schematic capture with PCB board layout. No other commercial packages were explored due to cost (OrCAD licences were already available within the School) and no free tools were mature enough while offering integrated schematic/PCB development.

However, experience with the design of the Photon platform indicates that the package is unsuitable in PLEB development for a number of reasons.

3.1.1 OS Platform Support

The major drawback of OrCAD is it limitation to Microsoft Windows operating system environments, while all other tools used in the project run on Linux and other UNIX platforms. Having to switch over to Windows to use OrCAD presents a number of problems which slow down development and restrict use of the software to machines running Windows. Since the Schools computer system is UNIX-based, this presents a major inconvenience.

3.1.2 Bugs and Inaccessible Functionality

A source of continual frustration with OrCAD is the instability of the software. This instability not only slows down development but has, on many occasions, resulted in loss of work requiring designs to be redone. Since the software runs on Windows and seems to use hidden file references, backing up designs to School accounts (where daily backup are done) have proved impractical.

Another source of frustration is the obfuscation of functionality. OrCAD makes heavy use of the Windows GUI and often displays attributes of the design with no clear way of modify them. The help files fail to shed much light on the situation and the only further documentation available seems to be through costly training seminars. The most distinctive functional limitation is that of back/forward annotation and pin/gate swapping.

Back and forward annotation refers to the synchronisation of the schematic design with the PCB board design. Automation of this is a common feature of all serious CAD packages of this type. Automatic annotation updates the PCB if a change is made in the schematics known as forward annotation while automatic updates of the schematics from PCB edits is known as back annotation. The main application of back/forward annotation is the swapping of component pins and sometimes gates[3]. OrCAD offers no clear pin/gate swapping function and provides severely limited automatic annotation. If faced with any inconsistency between the schematic and PCB design it will happily butcher your design without warning. While the damage can often be undone, restoring the consistency between the designs can prove difficult and time consuming, OrCAD only provides an undo operation with a single entry history.

One of the few strong points of OrCAD is its integration of a logic design and simulation package into the schematic capture program. Logic designs can be created and simulated in schematic sheets. This provides integration of logic designs in platforms making use of programmable logic devices such as complex programmable logic devices (CPLDs) and field programmable gate arrays (FPGAs).

3.1.3 Licensing Restrictions

The School owns a single educational licence for OrCAD. The major limitation of the educational licence is that OrCAD provide no support for the software. Most notably they only supply restricted libraries for schematic symbols and PCB board footprints. This results in a large amount of effort being required to build these libraries before schematic/PCB design can be started.

The restrictive licensing of OrCAD means that not many open hardware developer have access to OrCAD so, while PLEB hardware designs are open, interested parties cannot make use of the CAD files to correct and customise PLEB hardware designs.

3.2 Eagle CAD

Recently Cadsoft have released version 4.0 of their Eagle CAD software. This package is an attractive replacement for OrCAD for the following reasons.

3.2.1 OS Platform Support

One of the biggest advantages of Eagle CAD is its availability for both Windows and Linux Intel[4] based PCs. This means that hardware design development can run on the same system as the software development.

3.2.2 Functionality

The source of functional superiority of the Eagle CAD package is that while the software is GUI based all functionality is available in a command-line format. The command-line functionality is enhanced by the ability to write scripts. This speeds up development by allowing designs

[3] An example of this is a component containing multiple AND gates which all provide the same functionality.

[4] Windows and Linux Alpha based PCs should be able to run Eagle CAD as well with the FX86 emulator.

(schematic capture, board layout as well as component library) to be specified in a script rather then drawn with the graphical tools. As well as the scripting tool, a user language tool is available which can access the internal data structures of Eagle CAD to generate output files based on designs. Items like bill of materials, drill files and gerber files all used in the manufacturing of PCB boards, can be defined and generated this way. It is even possible to write programs to generate CAD files for other software packages.

An apparently minor but power feature of Eagle CAD is deep undo/redo buffers. These buffers can save a large amount of time in design and are easier to work then unwieldy backups. This and some other small but intuitive features provide ease and efficiency of use.

The only serious feature lacking in Eagle CAD is rotation of items is limited ot 90 degrees. Many packages provide single degree resolution or at the least 45 degree resolution. This is a limitation that will hopefully be addressed in future versions.

3.2.3 Product Availability and Usage

As long as commercial CAD packages provide superior functionality and stability to their open source counterparts, the open hardware development community needs to resort to using these commercial packages. Eagle CAD is an attractive option because of its favourable licensing.

Free Licence

A restricted demo version of Eagle CAD is available for free licensing. The main attraction of this is that only a small number of restrictions are place on the package. Many simple and practical hardware designs can be achieved despite these restrictions. More importantly, component libraries, scripts and user programs can be developed and existing designs can be loaded using the demo version. The restrictions of the free licence demo version are listed below:

- One schematic sheet only.
- 2 Layer PCB boards.
- Limited PCB board size.

Educational/Student Licences

Eagle CAD also has educational licences that provides full package functionality and full support from Cadsoft for reasonable pricing. While the pricing is still restrictive, it is probably the best that is commercially available.

Educational licensing is provided for institutions while a student licence is available for individual student usage. The only restriction is that the packages must be used for non-profit designs only. This suits the PLEB project as all its hardware designs are open.

For the above reasons Eagle CAD is being used by a growing number of open hardware developers, which means greater accessibility to PLEB hardware designs created with Eagle CAD.

The bottom line is that while Eagle CAD is not open source software, its functionality can be easily extended through is scripting tool and user language. This, together with the development of a user base 'community' sharing designs, libraries, etc, means that Eagle CAD provides a suitable CAD tool for PLEB hardware designs. Initial experience with the free demo version in the Nova (see section 4) platform design have supported this opinion. The only draw back of the program seems to be its heavy memory utilisation which can slow a system with a small amount of RAM (my laptop with 32MB has been a prime example of this) to a crawl. This, however, seems to be a common trait of a lot of CAD software.

Eagle CAD provides a suitable package until open CAD tools become available.

3.3 The IEEE 1149.1 (JTAG) Standard

Once the Photon hardware was available, the first step was programming and testing the boards. After a board is assembled, the FLASH memory is in an unknown state. Some method of bootstrapping the system had to be developed. Also early experience showed the need for a reliable method of testing constructed boards. Two major problems where encountered with the memory daughter board:

- A number of mixed up signals in the schematics resulted in some errors that were hard to track using software running on the platform.
- Initial yields of assembled boards were quite low with open/short circuited pins common. The only way to fix these was to re-solder all the components on the board which typically resulted in the board working, while a few boards remained non-functional.

Some systematic method is needed for testing hardware designs. The IEEE 1149.1 Standard, otherwise know as JTAG was developed, with this purpose in mind. Presented with ever shrinking chip packaging rendering physical probing of devices in circuits impractical, the JTAG standard was developed to allow in-circuit probing/testing of board level connectivity with optional support for testing the internal functionality of the devices via a serial chain of JTAG devices exported on a 4-5 wire interface to the board.

The StrongARM processor used on the Photon platform supports the JTAG standard. Using the JTAG port of the processor, the pins of the device can be sampled as well as driven. This can be used to check for open/short circuits on the signals the processor is connected to. The ability to drive signals can be utilised to program the FLASH memory of the device for bootstrapping the system.

The problem faced in using the JTAG standard for testing/programming is that no open source JTAG tools are available, and hardware to interface with the JTAG port was not readily available. The solution initially put forward by Phil Preston was to use a small Atmel AVR processor board equipped with a serial port and some custom software to act as an interface between a host development computer and the target hardware's JTAG port. This initially proved suitable in bootstrapping Photon systems, but was developed only as a tool for programming the Photon FLASH memory, and was not flexible enough to be used for board testing and programming of other platforms. This hardware developed into the Nova platform.

It is a result of this work, the the Anvil sub-project was started with the aim of developing a JTAG software toolkit for testing/programming any platform equipped with a JTAG port. This work is currently in its early stages of development.

3.4 Photon Availability

In the two years since Photon was created, availability (or lack thereof) has prevented wide spread usage of it in research project. In the last year a number of local UNSW project have started to make use of Photon as supplies became available but problems with distributing units outside of the university has made its use in non-UNSW research projects limited.

The bottom line is that the School is not in a position to commercially manufacture the units. The reasons for this are as follows:

- We do not have the man-power to manufacture a large run of units.
- We do not have the collateral to purchase components for a large run.
- The university is not set up to 'sell' products commercially.

So what are the alternatives for the future? Why don't interested parties just manufacture their own boards? The problem lies in the fact that making units of this kind requires a large up-front cost which is only sensible in medium to large runs. Also the problems involved in sourcing components and assembling them restrict most interested parties from "brewing their own".

The other reason external projects have not been able to make use of Photon is that the CAD files have not been made available. Basically this is because they have not been suitable for release. This is due to the following:

- The CAD files themselves can only be used by groups with access to OrCAD.
- The design makes use of components which are very difficult to stock.
- The design still has a few bugs that need cleaning up.

So how can these problems be addressed? One possibility that has works well with the LART project. Started at around the same time as PLEB, the LART project at Delft TU, Netherlands is a cousin project to PLEB, also based on the StrongARM processor. The LART project cooperates with a commercial setup who manufacture and sell units to interested developers. Setting up this kind of a relationship make boards available to interested parties while removing the pressure from the School to supply them, so the focus can be given to developing new hardware designs and the software to run on them. This addresses the problem of availability.

The problem of design availability is still outstanding. While the design itself is now available, it is not really practical because it makes use of components which are hard to get. The original design constraints are also no longer valid, and with the experience gained from the original Photon board, a new design is underway. The aim of a redesign is not only to improve upon the original design, but to make it more accessible by using more “off the shelf” components, etc.

Another issue that needs to be addressed is licensing of PLEB platform designs. Currently a hot topic of hardware developers is a GPL like hardware licence. While wanting designs to be freely available for research and commercial groups alike, developers want to insure that any derived work remains freely available as well. While there has been a lot of talk, no appropriate, legally sound licence has really been put forward. Currently PLEB platforms plan to use the LART licence.

3.5 Missing Pieces

While the Photon platform is now available with memory and ethernet boards, more is needed. The following section lists some of the missing pieces required to make Photon more accessible as a development platform.

Firstly a developer’s daughter board is still not available, something with everything including the kitchen sink for developers to play with. What is needed? Such a board might include the following:

Audio

A large area of the research into portable devices is multimedia applications. Enabling Photon with stereo audio input/output would open up this kind of research.

Battery supply/charger

Usability of Photon is currently limited to desktops with lab supplies as no one has yet developed a battery based power supply and charging unit.

PCMCIA and/or compact flash slots

One of the original design goals of PLEB was to make as much as possible use of generally available hardware for portable devices, and to avoid specialist designs, where “off the self” peripherals like PCMCIA ethernet/modems, etc are readily available. A particular need for PCMCIA has grown from the interest in using Photon in wireless communications environments. The easiest way to enable wireless communications on Photon are PCMCIA wireless ethernet cards which are cheap and effective.

IDE hard disk drive interface

> More serious applications require substantial non-volatile memory and the cheapest non-volatile memory are still hard disks. Laptop IDE drives, in particular, offer relatively low power and small size.

These and possibly other items would make up a general developers board, but other things are still missing. Specialist boards like ones offering GPS, gyros, accelerometers, etc for projects requiring mobile positioning or field programmable gate arrays (FPGAs) for reconfigurable computing or other hardware design synthesis should also be explored.

A final issue is the packaging of the unit. Photon lacks a housing, possibly equipped with a battery supply/charger. Such a housing would provide protection and increased portability but is yet to be designed.

4 Nova Platform

As mentioned in section 3.3 the need for a cheap JTAG unit resulted in the design of a simple Atmel AVR board for testing and programming JTAG equipped hardware. This resulted in the Nova platform. The Atmel AVRs are a range of simple 8-bit RISC processors which are equipped with a variety of input/output capabilities and are in-system programmable. This makes them ideal as little hobbiest processors.

4.1 Design Overview

The goal of the Nova platform two-fold. On one hand there is a need for a cheap and simple testing/debugging unit that interfaces between the Photon and a host or development computer. On the other hand, the AVR is an interesting processor for simple control applications, so the Nova should be a simple experimental/learning platform suitable as an introduction to microprocessors, assembly programming and digital logic interfacing.

In satisfying these goals the following design was adopted:

- 4MHz, 3.3 volt AVR with 8KB FLASH and 512B RAM.
- Parallel and serial (RS-232) communications ports.
- 2 I/O ports supplying digital I/O, analog to digital converters (ADCs) as well as pulse width modulated (PWM) I/O.
- High-efficiency voltage regulator which draws its power rail from the communications ports as well as the primary I/O port. The regulator can supply 3.3 volts from communications port lines with voltages down to the 3.3 volts.

The final design is still being routed on a double sided board. The Nova design was done and is being completed on the freely available demo version of Eagle CAD by myself.

4.2 Serial Communications Port

The original JTAG design simply made use of an AVR with a serial port interface to the host for issuing JTAG commands. This also doubled as the power source. This feature was kept in the Nova platform since RS-232 serial ports are both standard and common to nearly all desktop and portable systems.

The new Nova provides a single RS-232 serial port equipped with RTC/CTS handshaking lines (which must be driven in software). Diodes connect both input lines as well as the looped back DTR/DSR to the power supply rail. The AVR can force the serial buffers either off or on (for power savings) as well as detect if a valid serial terminal is connected to the port. The CTS handshake line is also capable of generating an interrupt to the AVR.

4.3 Parallel Communications Port

An extension of the original JTAG design on the Nova board is the addition of a parallel port connector. The parallel port server's two purposes. The first is to allow the in-system programming of the AVR, this is handled by the printer handshaking signals of the parallel port, the inputs of which are again connected through diodes to the power supply rail. One of these handshake lines is capable of generating an interrupt to the AVR. The in-system programming signals can also double as an SPI serial interface, to be emulated over the parallel port signals of the host or driven by a true SPI devices connected to the port, such as the Photon platform or another Nova platform.

The second purpose of the parallel port is for exchanging data between the host or development computer and the AVR or I/O port. The 8-bit data bus of the parallel port[5] is connected to the AVR and I/O port via a tri-statable transceiver. When the parallel communications port is not in use, the AVR isolates the data bus.

The motivation behind having an 8-bit data bus as well as SPI serial port capabilities is that the speed of the JTAG port (which dictates the speed of JTAG testing/programming) is limited by two factors:

1. The clock speed of the AVR, limiting the speed it can bit-bang out the JTAG signals.
2. The communications bandwidth between the host or development computer and the AVR.

Both the SPI port and the 8-bit data bus of the parallel port provide speeds greater then those available via the RS-232 serial port, however both are less common then RS-232.

4.4 I/O Ports

The Nova platform has two I/O ports. The primary I/O port is equipped with an 8-bit digital I/O port connected to the AVR/parallel port data bus as well as 8 ADC channels which can be configured to be digital I/O lines as well. The I/O lines of the port can be used for a JTAG interface, while the ADC/digital lines can be used for a touch screen interface, etc.

The second I/O port is a simple 3-bit digital I/O port. Each digital I/O signal is connected to one colour of a RGB LED which may be driven by the AVR or a device connected to the port. The unique feature of the second I/O port is that each of the 3 signals is capable of providing PWM signals of 10/10/8 bit resolutions. The 8-bit PWM is connected to the blue component of the LED. The PWMs can be used for providing a 28-bit colour LED, or for more useful things like a 3 channel servo controller.

4.5 Speeding up the JTAG Interface

As mentioned earlier, the two factors limiting the speed of the JTAG interface as provided by Nova are the communication bandwidth to the AVR as well as its clock speed. Atmel also provide a range of USB-equipped AVRs similar to the AVR used in Nova. The two notable features of the USB AVRs are:

USB port

> This can be used to communicate with a USB equipped host providing speeds in excess of RS-232, SPI and parallel ports. USB also provides 5 volts at up to 500mA, which is more reliable then "leaching" power off the signal pins of the communications ports as the current Nova platform does.

Higher clock speed

> The USB AVRs can run at up to 20MHz, providing much faster bit-banging capability of the JTAG interface then the the 4MHz speed of the current Nova board AVR.

[5] The parallel port interface is not standard and some hardware only supports 4 of the 8 bits.

A USB port version in place of the parallel port version of the Nova platform would provide a faster platform for experimentation, the one limitation being the loss of the ADC support provided by the AVR. Both version, however, would prove useful development tools as well as experimental/teaching platforms.

5 Photon Design Revisited

While the first version of the Photon board has started finding its way into use in a number of projects, the design is limited. As mentioned earlier, there are a number of problems with the current design that limit its availability. The most significant issue, however, is the fact that many of the original design constraints are no longer valid. In particular limiting PCB manufacture to 4-layer boards no longer necessary, allowing the CPU and memory boards to be merged in a slightly larger board outline.

For this reason I have started work on a second version of the Photon design. This section outlines the new design and the changes made from the original. While the design is still being finalised, the choices listed here are not expected to greatly differ in the final version.

5.1 Stand-Alone Board

The biggest change from the original original design is the decision to have a stand-alone base. That is, a board which requires only a power and serial cable to boot and talk to the outside world. This requirement sets the scene for the design. The final PCB should measure approximately 70mm by 80mm at 6-layers, providing the same functionality as the original CPU board with a bank of FLASH and DRAM on the one board.

The reason behind this change is to provide a smaller and cheaper base for building a Photon system. While 6-layer PCBs cost more to manufacture, a single 6-layer board is still cheaper then two 4-layer boards in reasonable sized production runs. Since the hope is to get the boards manufactured by a third party, this seems like the best way to go.

A slight change from the original design is the dropping of the RS-422 serial communications port in favour of providing connection to the USB slave function of the SA1100.

One of the big limitations of the original design was the components selected for the connectors on the board. The LCD/general I/O connector was changed in a minor revision because the component was hard to find and even harder to make cables for. Also, in many situation users simple wanted to connect a simple PCB board and a cable provides no support. For this reason, all connectors with the exception of the daughter board connector make use of 2mm dual inline sockets. These sockets, while being larger then the headers that mate with them, provide protection against shorts and allow PCB boards to be docked side-by-side or at a 90 degree angle, allowing a large degree of flexibility in how a system is constructed. Even the RS-232 serial and USB communications ports make use of these connectors to minimise board space and maximise flexibility. The LCD/general I/O connector is known as the side-card connector in the new design, to highlight its new flexibility.

5.2 Power supply

A typical area of portable and embedded systems research is how systems consume power over different loads and how to minimise this consumption. Newer processors not only provide clock speed scaling (like the SA1100) but provide scaling of the processor's core voltage rail. In fact, the LART project has shown that even though it doesn't officially support it, the SA1100 will work with voltage scaling to provide quadratic power consumption reductions.

To facilitate this kind of research, the new design aims at providing a high efficiency power supply and the following features:

Isolated power supply rail

Unlike the original design, the power supply rail is diode protected. The supply rail can draw power from the daughter board, side card and USB connectors simultaneously.

Variable core voltage rail

Voltage scaling can offer quadratic power reduction, and since the processors core consumes a large percentage of the processors power consumption, scaling of the core voltage rail is an attractive feature. To make use of this, the board must also be able to get an accurate reading of the current core voltage. The new design provides a core voltage range of 1-2 volts.

Current sensing on 3.3 and core voltage rails

To support calculation of power consumption of the whole system, current sensing (or measurement) is provided for both the 3.3 and core voltages rails.

High supply rail voltage range

The new power supply design can operate on voltages from 3 to 28 volts, making it usable across a wide range of power supply sources from the common single cell lithium-ion batteries to lead-acid batteries hanging off solar cells.

It is hoped that with the above functionality, the power supply of the new Photon can form a reference for future designs both within the PLEB project as well as outside of it.

5.3 AVR Peripheral/Power Controller

The voltage and current sensing of the new power supply design requires the use of ADCs to do the actual measuring. Having ADC channels available to the daughter board and side card connectors allows for extensions like power supply voltage and current sensing and touch screen interfaces.

Since the ADCs are primarily involved in the power supply, and a number of power saving features of the design require the processor to control them, the decision was made to adapt an AVR for the task. The design allows the AVR to independently measure and control the power usage of the board. The AVR can also force the sleep or reset of the SA1100 processor, to act, for example, as a watchdog. The AVR talks to the SA1100 via the SPI serial interface, and the AVRs RS-232 serial interface can drive one of the two RS-232 drivers to control/report the state of the power supply rails. Since the AVR is in-system programmable, the design makes use of this fact by allowing the SA1100 to program the AVR in-system.

An interesting aside is that the Compac IPAQ platform uses an AVR for its touch screen interface controller and other features. The choice to use an AVR was made independently of the IPAQ design and it is encouraging to find similar choices were made.

5.4 Daughter-board and side-card connectors

A few significant changes have been made the design of the daughter-board connectors that proved so popular with the original design. Unfortunately, the changes are incompatible with the original board, but have been planned to form the basis for future designs.

The first change is the connector components used. The original design made use of two 70 pin connectors which can only be purchased special request to the manufacturer. In aiming for "off the shelf" parts, two 100 pin high density connectors of the same size as the original were used. These parts are readily available and are general cheap. The same layout, using the connectors as the structural support for the stacking daughter boards, was maintained. However, a larger spacing will be utilised to allow for less congested daughter boards.

The second major change is the adoption of a complex programmable logic device (CPLD) for routing the GPIO signals from the SA1100 and the daughter board and side card connectors. The reason for this is that the GPIO signals that are used for 16-bit LCD displays double as dedicated interrupt signals. The daughter-board connector on the original Photon had very few

GPIO signals for I/O or interrupts, which severely limited the functionality of the daughter boards attached. The CPLD design allows routing of the SA1100 GPIOs to the daughter-board and side-card connectors, as required, and also allows the synthesis of logic to multiplex signals or implement simple digital circuits.

As well as connecting a number of SA1100 GPIOs, the CPLD is also connected to the AVR, so that power management of the daughter-board and side-card connectors can be managed by the AVR. The CPLD can also be set up as the arbiter of the SA1100 system bus allowing daughter boards to bus master. The CPLD also controls the SPI communication bus, so that the SA1100 or AVR can talk to each other (or extra SPI devices connected to the system) via the daughter-board and side-card connectors.

The CPLD used is a low-power device with in-system programmability via a JTAG interface connected to the SA1100. With this, the SA1100 can program the CPLDs functionality as well as probe its signals for testing. For this reason, the CPLDs JTAG chain is isolated from the SA1100 JTAG chain, otherwise the SA1100 could control its own JTAG interface which could lead to problems.

The final major change to the new daughter-board connector design is the buffering of all signals that are not routed by the CPLD. This is basically the system bus. The buffers were added to reduce the signal degradation caused by daughter board stacking. Since CPLD and buffers are 5-volt tolerant, daughter cards may make use of 5-volt logic (although they must derive their own 5 volt rail from the power supply rail). This allows, eg, like single slot PCMCIA daughter cards to be made with a minimum of parts, as well as interfacing with other processor boards.

As mentioned before, the side-card connector is simply the replacement for the old LCD/general I/O connector. This port is similar to the daughter-board connector, however it only supplies ADC channels from the AVR and I/O signals from the CPLD. No system bus signals from the SA1100 are presented to the side-card connector. Unlike the daughter-board connectors, the side-card connector is designed to facilitate a single card. The use of 2mm dual-inline sockets means that the simple and cheap-to-manufacture 2-layer PCBs can be used. The aim of the side-card connector is to make it possible for cheap and simple boards for, eg, LCD interfacing (with touch screen support), or battery supply plus charger circuits, can be developed by groups without access to advanced PCB manufacturing.

5.5 Testing/Programming Port

The testing/programming port provides access to all the programming facilities of the platform, as well as to a number of test points to calibrate things like the ADCs of the AVR and the watch crystal of the SA1100. These calibrated values can then be stored in the platform's FLASH memory or in the AVR's non-volatile memory.

The port exports the JTAG chains for both the SA1100 and CPLD as well as the in-system programming port of the AVR. This allows the whole system to be configured and programmed with the one connector. Also, the write protection feature of the FLASH devices used is made accessible only via this port. This feature allows the locking of certain blocks of memory within the FLASH. By making use of this feature, the firmware of the system can be programmed into the platform and then locked in to avoid software corruption, while the unlocked blocks maybe be erased and updated by software.

6 The Projects Future

The aim of this report has been to outline the progress that the PLEB project has made in providing a generic building block for hardware- and software-based research into the growing field of portable and embedded research. While PLEB has been fairly low key to date, student and staff interest in the project is now at a point that the project needs to become an organised group rather then an individual working on it in his spare time. With continued support from the

School I believe that we can produce a commercial grade platform for researchers and hobbiests alike and that it can provide a platform for thesis and course work projects within the School.