

# The iLab Mini: A Low-Cost Experiment Platform for the Developing World

Piotr Mitros

August 1, 2007

## Abstract

This paper describes a system that improves access to instrumentation equipment for electronics experiments in the university system of the developing world. Based on field work in Nigeria, this paper evaluates two complementary approaches. The bulk of this paper presents the iLab Mini, a low-cost data acquisition board capable of supporting basic electronics experiments. In addition, it discusses the uses and limitations of on-line electronics experiments using the MIT iLabs architecture. This paper is targeted at people wishing to develop and deploy on-line and conventional laboratories based on the Mini.

## 1 Introduction

In the tertiary education system in many parts of the developing world, the cost of instrumentation equipment is prohibitively high. Students have limited access to very rudimentary equipment, and are forced to share such equipment among many students. Driven by field work at Obafemi Awolowo University (OAU) in Nigeria, as well as experience in several other developing countries, this paper describes some approaches for increasing access to electronic experiments in the developing world.

Based on the constraints found in the developing world, a low-cost data acquisition board, the iLab Mini, was developed. Given a computer, this \$30 board is capable of replacing a low-speed oscilloscope, function generator, and power supply. Alternative approaches to designing this board, giving either lower cost and lower performance, or high cost and higher performance, are also evaluated.

The main Mini board provides eight analog inputs with a nominal 10 bit accuracy, and acquisition speeds in the low kilohertz range (up to a total of 77KSPS at reduced resolution). It provides eight analog outputs, also with 10 bit resolution, with output speeds of around 30KSPS. It also provides eight digital GPIO pins, and an external SPI interface. It is USB-powered, and is capable of supplying close to 500mA at 5V, and up to 50mA of power at 3.3V to external experiments. It is capable of communicating with the computer at speeds of up to 57.6kbps-2.5Mbps, depending on configuration.

A number of experiments can be connected to the Mini. The most developed uses a daughter-board to transform the Mini into a basic parameter analyzer.

In addition, this paper will discuss the MIT iLabs architecture and software. MIT iLabs is an architecture designed to let students remotely control experiments over the internet. iLabs is primarily intended for pedagogical use – students can have a partial laboratory experience remotely at a much lower cost and time commitment than a traditional laboratory would require. Although iLabs cannot replicate the full laboratory experience, it does allow students to have a partial laboratory experience at much lower cost and with a much lower student time commitment. Students can observe the difference between model and reality in many contexts where a traditional laboratory would be either time or cost prohibitive. As a result, it has the potential to substantially broaden access to electronic experiments, since a single experiment may be shared by hundreds or thousands of students.

Having been initially developed for use in the developed world, however, it has a number of limitations with regards to use in the developing world. The reference implementation iLabs infrastructure also requires expertise in a wide range of technologies, and requires a very significant effort to develop new labs. It requires an expensive stack of proprietary software. The protocols are bandwidth inefficient, and do not operate well over low bandwidth/high latency networks, as found in the developing world. This paper discusses these limitations, and presents solutions to some of them.

Finally, the Mini board is integrated into the iLabs architecture provide a complete solution for rapid, low-cost deployment of on-line electronics experiments. The goal of this system is to enable lecturers to develop and deploy simple iLabs around specific problem sets in a matter of hours. More sophisticated labs may require software and firmware changes, but these should take months, rather than years, to develop. In the developed world, this means that students will be able to compare analytical answers on problem sets to reality with minimal overhead. In the developing world, the impact is obviously much greater.

At present, I have deployed an iLab Mini board with a protoboard version of the parameter analyzer on the Mini lab server for development use at OAU. I also left a number of unpopulated PCBs. Developers at OAU have expressed an interest in working on further developing the Mini both within and outside of the iLabs architecture.

This paper will first present the context of the work – the conditions found in Africa, and why this work is necessary. The next section will describe the design tradeoffs of the iLab Mini board. Next, it will show an on-line laboratory experiment based on the Mini. This will consist of two parts – a board that converts the Mini into a parameter analyzer, and a laboratory server that allows the Mini to be used with the MIT iLabs infrastructure.

## 2 Context of the Work

The technical work in this paper is based on field work at OAU in Ile-Ife, Nigeria, totaling eight weeks over three trips. OAU is our primary partner school, although MIT iLabs has partners in a number of other schools in Africa and throughout the world. It is one of the largest and most prestigious universities in Nigeria.



Figure 1: An introductory physics lecture at OAU.

OAU has extreme resource shortages, both human and material. OAU has only 1,700 faculty educating 25,000-30,000 students. A typical introductory physics lecture, shown in Figure 1, has well over a thousand students. The electronic engineering department has a total of 14 staff – 3 professors, 1 reader, 5 lecturers, 4 assistant lecturers, and one graduate assistant. This staff is responsible for teaching a curriculum consisting of 48 courses (roughly 24 per semester) to 700 students – a student:staff ratio of 50:1.

Due to lack of funding, equipment shortages are also extreme. Introductory students share lab stations consisting of a power supply and multimeter (no function generator or oscilloscope), with 6-8 students per station, as shown in Figure 2.

A significant portion of students have personal computers, and virtually all students have some access to computers. Students generally do not currently have university-provided network access. Many academic buildings have networks, but access is restricted primarily to staff. Students are permitted, in the context of some classes, to use computers in computer labs. The university plans to build a dormitory network soon. All students have access to computers with low-speed internet through internet cafes which, although moderately expensive for students, are not unreasonably so.

University internet access is, and will continue to be, very limited. Due to the lack of a fiber connection, internet is available only through satellite connections. As a result, bandwidth is roughly 300 times more expensive in most African countries than it is in the United States. At the same time, the GDP/capita is roughly fifty times lower<sup>1</sup>. While cost figures for OAU are unavailable, figures for bandwidth cost at Makerere University Kampala in Uganda are shown in Table 1. The unit cost of bandwidth at OAU is similar, although the total bandwidth was recently upgraded to 6Mbps.

---

<sup>1</sup>Most GDP/capita figures given adjust for purchasing power parity. When comparing Internet costs, it is important to use unadjusted figures.



Figure 2: Introductory EE students working on a laboratory. Equipment stations are shared by large numbers of students, and consist of a power supply and multimeter.

	MUK	MIT	MUK/MIT ratio
Campus Gateway (Mb/s)	2.5	$\approx 2,300$	$\approx 10^{-3}$
Gateway cost (\$/month)	\$28k	\$80k	$\approx 1/3$
GDP per capita	\$300	\$36k	$\approx \$0.01$
Bandwidth cost relative to per-capita GDP			$2.5 \cdot 10^{-5}$

Table 1: Costs of bandwidth at Makerere University, Kampala in Uganda vs. MIT. Figures taken from 2004.

The use of instrumentation equipment that requires an unnetworked computer is very reasonable for most students. Use of remote laboratories through iLabs hosted in the United States is reasonable if limited to a small number of uses per semester. More frequent usage would likely be impossible, due to cost of internet cafes, and shortage of computer labs. Once the dormitory network is in place, frequent use of locally hosted iLabs will also become very reasonable. Use of some remotely-hosted experiments may be bandwidth prohibitive, depending on bandwidth requirements of the experiment.

In addition to work at OAU, the author worked on a team to deploy MIT iLabs at Zhejiang University and Dalian University of Technology in China, and has, independently of this work, visited the Indian Institute of Technology Madras. These also helped motivate the work presented in this paper.

### 3 Mini Hardware Architecture

The digital board consists of an FT232RL USB chip connected to an ATmega48 microcontroller, which is in turn connected to an LTC1660 DAC. The ATmega48 includes an eight channel, 10 bit ADC, capable of 15kSPS at full resolution and 77KSPS at reduced resolution. The LTC1660 consists of eight 10-bit DACs, each independently capable of operating at full resolution at 33KSPS. The USB chip communicates with the microcontroller through an RS232-based interface (although operating at TTL levels, and capable of greater speeds than traditional RS232).

#### 3.1 USB Interface

The USB interfaces uses an FT232RL USB chip. This chip is traditionally used in USB-to-serial converters, and provides an RS232/RS422/RS485-compatible serial interface (although at TTL signal levels, rather than RS232 signal levels, and capable of much higher speeds).

The maximum throughput depends on hardware and software configuration. The FT232RL may be used either with a generic RS232 serial port driver, or with an FTDI-specific driver. Using the generic RS232 driver simplifies development, but limits operation to standard RS232 speeds. Using the FT232-specific driver allows a wider range of speeds, but complicates software development. The Mini board also exports 4 pins on the FT232 that may individually be configured to either act as GPIO pins, or to output a clock at either 6MHz, 12MHz, 24MHz, or 45MHz. These may be used as GPIO pins only with the FT232 driver.

The FT232RL generates a 6MHz or 12MHz clock that may be used by the microcontroller. For higher-speed operations, the Mini board may also be configured to use a crystal oscillator (this is configured by soldering jumpers on the PCB). The driver and clocking scheme will determine the maximum speed possible for communication with the computer. Maximum throughput with different driver and clocking configurations is shown in Table 2. The board has only been tested with the standard serial driver – it has not been tested to the maximum speeds possible with the FTDI driver.

Clock	Driver	Maximum Throughput	UBRR	U2X
6MHz on-board	RS232	57.6kbps	12	1
6MHz on-board	FTDI	750kbps	0	1
12MHz on-board	RS232	115.2kbps	12	1
12MHz on-board	FTDI	1.5Mbps	0	1
7.3728MHz crystal	RS232	115.2kbps	3	0
7.3728MHz crystal	RS232	230.4kbps	1	0
10MHz crystal	FTDI	1.25Mbps	0	1
20MHz crystal	FTDI	2.5Mbps	0	1

Table 2: Maximum clock rates for common clocking configurations, as well as associated register settings. Some possible Mini-compatible microcontrollers are limited to 10MHz clock speed. Some RS232 software is limited to 115.2kbps, while other is limited to 230.4kbps.

USB also provides power to the chip, and potentially, to any external experiments. One caveat is that although USB supports high current levels (up to 500mA), large capacitors may draw out-of-spec in-rush currents. USB is only rated to drive a  $10\mu\text{F}$  load capacitance, or  $50\mu\text{C}$  of in-rush charge. Empirically, however, most computers can drive capacitances significantly above this specification. By USB spec, the voltage level is rated at 4.75-5.25V. Practically, most computers power supplies are very close to the ideal 5V. Powered USB hubs, in contrast, have a fairly high variance in output voltage (largely dependent on load). The model tested outputs 5.15V.

The USB specification requires USB peripherals to implement a sleep mode with a maximum current draw of 2.5mA. The Mini does not implement this mode. This may prevent computers from suspending with the Mini attached. The specification also requires USB peripherals to draw at most 100mA until initialization. While the current draw of the Mini itself is less than 100mA, it has no way to limit the current draw of any external circuits it may be powering. Both of these properties, while out-of-spec, are common among USB-powered peripherals<sup>2</sup>. These features were left unimplemented primarily to reduce cost and complexity of assembly. The FT232R includes facilities for implementing sleep mode – all that is required is an external MOSFET for switching power to the rest of the circuit, and an RC-filter. This would, however, add about \$1 in cost to the board (\$0.30 in quantity), slightly increase board size, and slightly complicate board assembly.

### 3.1.1 Rationale for USB

Although the cost of the USB chip is several times greater than that of an RS232 level converter, USB was chosen for several reasons:

- Many modern computers no longer support RS232 or parallel ports natively – these

---

<sup>2</sup>Indeed, many USB-powered peripherals, such as flashlights, vacuums, and fans, continuously draw over 100mA from USB, and do not negotiate power consumption at all.

require a separate USB adapter.

- USB is designed to power an external circuit. It can supply up to 500mA of current at 5V with over-current protection. For many experiments, this eliminates the need for an external power supply, reducing system cost and complexity significantly. In contrast, neither serial nor parallel ports are designed to power external devices. While both can provide some power, both types of ports are easily damaged if too much current is drawn. In particular, in the case of a serial port, the amount of power available is quite small, not defined by specification, and varies from port to port.
- USB hubs allow multiple Mini experiments to cheaply and easily connect to the same computer – this is very important when developing remote laboratory experiments where it is desirable to have one server host many experiments. In contrast, multiport RS232 and parallel cards are uncommon and expensive.
- The USB chip requires fewer passive components (and therefore simpler assembly) than an RS232 charge pump.
- The specific USB chip can also be used to generate a clock for the microcontroller. In many cases, this eliminates the need for an external crystal.
- USB provides substantially higher bandwidth than either serial or parallel interfaces.

The result is that, in most cases, the overall system cost is much lower with USB than with other interfaces. Furthermore, we expect cost of USB chips to keep falling, while the cost of RS232 level shifters (or additional microcontroller pins for a parallel interface) has remained relatively constant.

It is possible to implement USB directly on an Atmel microcontroller, without an external USB chip. This approach would have removed one of the major cost-items. Nevertheless, we chose to use an off-board USB chip for several reasons:

- While it works with most USB controllers, the Atmel USB hack falls outside of USB electrical specifications. In particular, it does not work with many Sony and Dell laptops. The number of incompatible devices may increase in the future – many working but non-compliant USB 1.0 devices were not compatible with USB 2.0 controllers.
- The firmware-only USB implementation requires substantial amounts of processing power and memory. It is not clear how much processing power and memory would remain for running the experiment. Furthermore, due to the possibility of other devices on the bus, USB timing may not be predictable. It is not clear if a software-only USB implementation would allow for the easy implementation of experiments with deterministic timing.
- The firmware-only USB implementation requires custom drivers on the computer – it cannot be used as a virtual serial port<sup>3</sup>. As a result, driver development is much

---

<sup>3</sup>Although it may be possible to re-implement it to do so.

more complex, and much more platform-specific. It also cannot be used with standard terminal programs for debugging purposes.

- The substantial increase in firmware complexity would complicate the development of new firmware versions for new labs.

Nevertheless, this approach is worth future experimentation – a software-only implementation would significantly reduce the bill of materials – the USB chip is the second most expensive part of the board, and also consumes significant PCB area. It is also the most difficult component to manually solder, which poses particular problems in the developing world.

## 3.2 Microcontroller

The Mini uses the ATmega48 microcontroller. At design time, this was the lowest-cost microcontroller available that could provide adequate I/O for the parameter analyzer.

The ATmega48 provides a single 10-bit ADC, multiplexed to 8 pins. The ADC is capable of 15kSPS at full accuracy, and up to 77kSPS with reduced accuracy. While the Mini's conversion speed vs. accuracy has not been tested imperically, anecdotal evidence about the Atmel microcontrollers indicates that full 10-bit accuracy is rarely achieved in reality.

The microcontroller is pin and feature-compatible with the ATmega88 and ATmega168. Code written for the ATmega48 should compile without changes on both of these microcontrollers. These microcontrollers increase available memory at additional cost. In addition, it is partially pin-compatible with the ATmega8 and ATTiny28, although without maintaining source or feature compatibility. The Mini may be assembled with either of these alternative microcontrollers at lower cost for specific applications that do not require the full functionality of the basic Mini. The available compatible microcontrollers are summarized in Table 3.

## 3.3 Digital to Analog Converter

The Mini board uses the LTC1660 DAC. The LTC1660 consists of eight 10-bit DAC channels, each independently capable of operating at full resolution at 33kSPS<sup>4</sup>. The LTC1660 was primarily included due to design requirements for the parameter analyzer. For many applications, the LTC1660 is overkill – the LTC1660 is the most expensive component of the Mini. There are several ways to reduce the cost of the Mini if the application does not require 8 channels with 10-bit precision at 33KHz.

First, the LTC1665 may be used in place of the LTC1660. The LTC1660 and the lower-cost 8-bit LTC1665 are fully pin and software compatible. Both accept 12 bit values, and

---

<sup>4</sup>Greater speeds are possible at reduced resolution, although this generally requires only using a subset of the channels – if all 8 channels are used, the SPI data rate limits us to under 40kSPS per channel.

Device	Flash	EEPROM	SRAM	Compatibility	Price Ind. (Quant.)
ATTiny28	2k	–	32	No - 4MHz, no ADC	\$1.05 (\$0.88)
ATMega8	8k	512	1024	No (partial)	\$3.55 (\$2.06)
ATMega48	4k	256	512	Yes	\$2.58 (\$1.50)
ATMega88	8k	1024	1024	Yes	\$3.76 (\$2.19)
ATMega168	16k	512	1024	Yes	\$4.00 (\$3.28)

Table 3: Microcontrollers compatible with the Mini main PCB. The ATTiny28 does not provide ADC, and is limited to 4MHz. The ATMega8 is limited to 8MHz or 16MHz, depending on part, and has a slightly different feature set from the ATMega48/88/168. It should be compatible with most applications given a firmware port. The ATMega48/88/168 are mutually interchangeable, only differing in amount of memory. Applications written for the ATMega48 should work on the ATMega88 and ATMega168 without changes.

ignore the lowest order bits. Using the LTC1665 can reduce the cost of the board substantially – the LTC1660 costs \$11.38 individually and \$5.70 in quantity, while the LTC1665 costs \$6.38 individually, and \$3.20 in quantity.

Second, the LTC1660 may be entirely omitted – the ATMega48 has a 6-channel hardware PWM channels. The PWM may be combined with an external low-pass filter to act as a low-quality DAC.

Future design iteration of the Mini will investigate the possibility of cost savings from using lower-cost, lower-speed 4 channel DACs. This would be adequate for the parameter analyzer, and may reduce system cost.

## 4 Hardware Cost

The hardware cost for the iLab Mini depends on the exact configuration assembled. In a typical configuration, the cost for the parts to populate a single board (excluding the cost of the PCB itself) is about \$25.07. In quantity, this cost drops to \$12.46. The bill of materials for a typical configuration is shown in Table 4. Due to the large overhead for tooling (or prototype production methods), PCB costs are very quantity-sensitive. At Advanced Circuits, the per-PCB production costs for the 2x1.68 inch PCB at planned quantities are about 4-9 dollars. The cost at different volumes is shown in Table 5. Lower PCB costs could be achieved by using a low-cost Chinese manufacturer instead of a high-end American one.

Since the Mini may be assembled in different configuration for different purposes, the actual costs may increase or decrease depending on unit. If the unit uses an oscillator crystal instead of using the on-board oscillator, the costs increase by \$0.50 individually, or \$0.28 in quantity (this requires two 33nf 0805 capacitors, one additional 0Ω jumper resistor, and the crystal). For fixed installations, omitting the header connectors and soldering wires directly lowers the cost by \$5.15 (or \$2.35 in quantity). There is also room for two additional

Part	Single	High Quantity
FT232R	\$4.05	\$2.30
USB connector	\$1.04	\$0.37
10n capacitor	\$0.03	\$0.01
5x 0.1u capacitor	\$0.04	\$0.01
Ferrite bead	\$0.12	\$0.06
ATMega48	\$2.58	\$1.50
LTC1660	\$11.38	\$5.70
8x 470 ohm resistor	\$0.04	\$0.02
3x 0 ohm resistor	\$0.04	\$0.02
34-pin header connector	\$3.10	\$1.36
10-pin header connector	\$2.02	\$0.89
Total cost	\$25.07	\$12.46

Table 4: The bill of materials for a typical Mini digital board configuration. The first column shows costs when parts are ordered in minimum quantities. The second column shows costs when parts are ordered with the maximum quantity discount available from the distributor.

Quantity	Unit cost
1	\$33.00
15	\$8.33
50	\$4.05
150	\$1.72
1000	\$0.73
50000	\$0.56

Table 5: PCB production costs at Advanced Circuits for the Mini main board, including academic discounts, excluding shipping costs.

optional bypass capacitors, whose cost would depend on capacitance.

In the developing world, the Mini would be distributed as a kit, so assembly costs could be kept low. Work could either be done by students or by technicians. In the case of student work, the only costs would be solder, solder wick, flux, and periodically, soldering iron tip replacement. If the work were performed by a local technician, assuming 30 minutes for assembly, and a salary of \$6,000/year, assembly costs would be \$1.50-\$3.00 per board<sup>5</sup>. If the Mini were distributed preassembled in large quantities, assuming \$0.01 per solder point, the assembly costs would be about \$1.60 per board. Preassembling in small quantities is uneconomical.

## 5 Mini Assembly

Since the Mini uses fine-pitch surface mount parts, assembly requires high-quality tools, and an adequate level of soldering expertise. Both of these are uncommon in the developing world. The cost of soldering equipment is high, but not prohibitively so (around \$200), although availability of high-quality soldering equipment is limited. Assembly instructions are available on the Mini web site, and a series of soldering videos is being prepared.

The Mini uses a standard 5-pin AVR hobbyist programming connector for programming the ATmega48. This is a single-row connector with the pins RESET, SCK, MISO, MOSI, and GND. The Mini may be programmed through any interface that can toggle 3 output pins and read 2 input pins at 5V TTL levels, including simple serial port, parallel port, and USB programmers.

The FT232R implements all of the pins of an RS232 connector, and the control pins are presently unused. In future iterations of the Mini, these may be connected to allow the Mini to be programmed through the normal USB interface. This would make deploying individual Mini boards much easier. This was not a high priority, since this gives unusably low programming speeds for development work, presumably due to USB latency<sup>6</sup>. We would prefer to encourage users of the Mini to also develop on the Mini, and people relying on this mode of programming would be unable to do so.

Detailed instructions for assembling a Mini are given in Appendix A.

---

<sup>5</sup>\$6,000 is an upper bound on the salary of a technician in Nigeria. The \$1.50 figure assumes no overhead. The \$3.00 figure assumes overhead equal to salary.

<sup>6</sup>USB is moderately high latency. Low latency USB devices have on the order of 1-30 ms latency, depending on device). Assuming AVRdude waits for each operation to finish before the next one, sending one bit requires 3 bit flips (3-90 ms). Sending one byte requires 24 bit flips (24-720ms). This gives ballpark of 1-42 bytes/second. The one-way transfer rate achieved when bit toggling over an FTDI USB-to-serial converter was 5 bytes/second, so close to the middle of this range. Since data is typically verified after an upload, this gives a programming speed of roughly 2.5 bytes/second. For comparison, the USBTiny programmer has a programming speed of roughly 820 bytes/second – 330 times faster. This is not a simple serial port issue – the performance seen with normal serial ports is normal – the issue only comes up with USB→serial converters.

## 5.1 Future Directions

There are a number of possible minor improvements to the Mini main board, as well as several possible architecture changes.

There are several possible directions in which the Mini main board may be improved:

- Cost
- Performance
- Ease-of-use
- Robustness

Driving the cost down most likely requires sacrificing performance. An implementation with software-only USB and PWM DAC could consist of essentially an Atmel, a clock crystal, a USB connector, a header connector, and surface-mount capacitors and resistors. The total cost of the board ought to be half of the cost of the current Mini. The board would also be simple to assemble – if the number of channels were cut, it could use only through-hole parts.

ADC conversion speed is the major performance limitation of the Mini – a lot of student circuits exhibit RF pickup or high frequency oscillations that the Mini is not capable of monitoring. These require bandwidth on the order of 100MHz to observe, and similar conversion speeds. Achieving this is fundamentally expensive – a high-speed 10-bit ADC itself costs over \$8 in quantity, and close to double that individually. Most high speed ADCs have a limited signal range (typically, current input or  $1-2V_{PP}$ ), so require additional electronics to level-shift. In addition, the digital logic would have to be substantially upgraded to be capable of receiving data at 100Msps. The overall cost would most likely more than double that of the current Mini.

The block diagram of a possible design of such a low-cost high speed data acquisition is shown in Figure 5.1. In this design, the ADC is connected directly to a high-speed SRAM. When the system is finished acquiring, the microcontroller disables the ADC, takes over the SRAM clock, and reads the data out (much more slowly). This design eliminates the need for an expensive high-speed microcontroller or FPGA.

As shown, the system is limited to capture at one speed. The ADC on the microcontroller could be used to capture at 15ksps. To capture intermediate speeds, a second clock would be necessary, and a clock select switch. Due to the large capture range of 128k samples, the circuit would only need one intermediate speed (approximately 1Msps) to display a screen of data at any speed.

With the current design, there is some incremental work left on robustness and on ease of use. First, adding over-current and over-voltage protection on most of the pins (especially power output) would be useful. Although the USB spec requires USB ports to include over-current protection, this may not always be robustly implemented<sup>7</sup>. Second, as mentioned

---

<sup>7</sup>One Dell laptop would simply power down on an over-current condition, rather than limiting USB power.

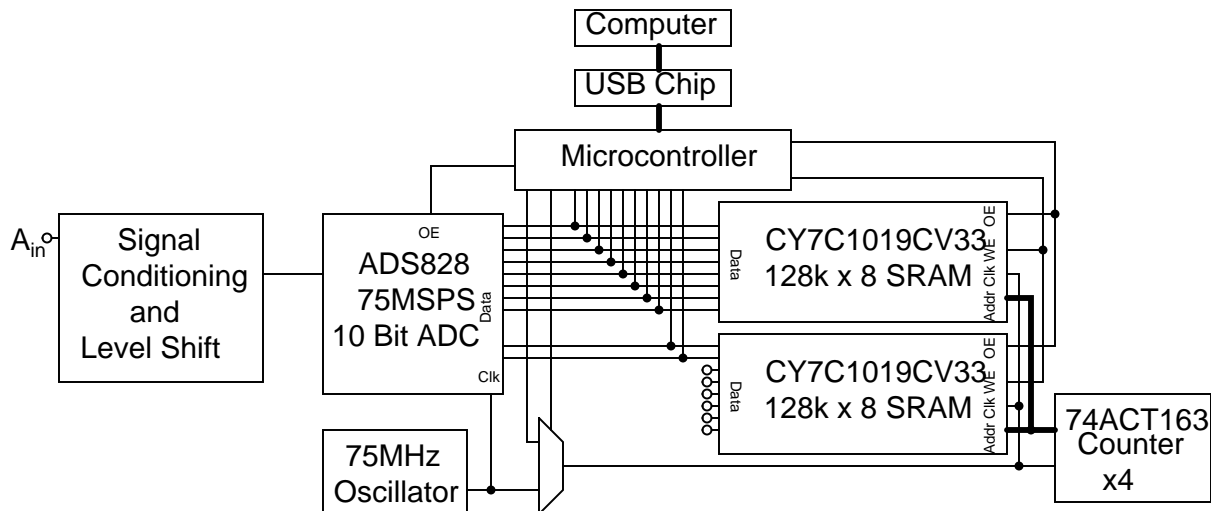


Figure 3: A possible design for a low cost, high-speed data acquisition system.

before, adding the capability to program the ATmega48 from the control pins on the FT232R would eliminate the need for a separate programmer for one-time deployments. Third, adding a  $V_{DD}$  pin to the programming header would allow more convenient use of official Atmel programmers.

## 6 Parameter analyzer experiment

The Mini digital board can be used for a number of experiments. The simplest experiments simply analog inputs as low-speed oscilloscopes, and the analog outputs as function generators. More sophisticated experiments require external circuits and custom firmware. The first such experiment developed is a parameter analyzer.

A parameter analyzer is an instrument capable of characterizing the I-V characteristics of electronic devices. The parameter analyzer daughter-board allows the Mini to control four symmetric measurement unit (SMU) ports. Each SMU port outputs either a controlled current or a controlled voltage, and measures both the current and voltage output.

The parameter analyzer board is targeted at on-line laboratory experiments – not for conventional student use. It is designed to be very low cost. As a result, instructors can use a deploy-and-forget model. Rather than needing to recover equipment by taking down an experiment once a class has finished using it, instructors can leave the experiments on-line for use in future semesters, at other universities, and by distance and self learners. We hope that this will help foster a community around the iLabs concept and architecture.

In contrast to traditional parameter analyzers, the daughter-board has a fairly narrow dynamic range. The Mini only has 10 bit DAC/ADC capability. As a result, a parameter analyzer with a maximum current of 100mA can, fundamentally, at most have a resolution



switches  $f_1$  and  $f_2$ .

Since the differential amplifier has gain, the voltage dividers on the input to IC1 are necessary for stability in current feedback mode. Unfortunately, in voltage feedback mode, they reduce loop gain. Reducing gain of the differential amplifier would reduce the need for these, but would lower the number of bits available for current readings. Increasing the shunt resistance would compensate for this, but decrease headroom. While there are several ways to omit these from the voltage feedback loop, while keeping them in the current feedback loop, all of them involve further increasing the already large parts count or causing other problems.

The board uses the MAX395 switches, which were chosen due to the relative low-cost, and the well-controlled on-resistance. They are also pin-compatible with the MAX335 switches, which may be substituted into the board with an external voltage supply for higher voltage applications.

The operational amplifier used in the reference is the NJM62, which was chosen for a number of reasons:

- FET-input (low input bias current).
- Low-cost. \$0.32-\$0.51 for 1 unit, and \$0.196-\$0.38 in high quantities for a dual op-amp.
- Wide rails  $\pm 2V$ - $\pm 18V$ . This allows us power from the 5V USB power supply, but to also support operation from an external power supply for wider range of operation.

The major problem disadvantages of this op-amp are:

- It is not rail-to-rail, reducing the  $\pm 2.5V$  rails even further.
- It has a fairly low output current. The maximum rail voltage falls off substantially if the output current is increased from 1mA to 10mA. This makes it impractical for power devices.

A number of pin-compatible operational amplifiers exist that address one or both issues. For high current devices, the AD8532 supports up to 250mA output current in a pin-compatible form factor.

The SMU requires a three power rails, while the Mini only provides 0V and 5V at useful levels of current and voltage. As a result, the circuit shown in Figure 5 drives a 2.5V rail (ground for the purposes of the parameter analyzer). In most applications, Q1 and Q2 are omitted, and R1 and R2 are 0 $\Omega$  jumper resistors. The optional configurable output driver circuit is provided in case the circuit being observed requires additional power. In most cases, it is omitted, since the NJM62 can drive adequate current for normal operation, and the crossover distortion of the push-pull pair reduces power supply accuracy<sup>8</sup>. A regulator was not because:

---

<sup>8</sup>If Q1 and Q2 are used, use an operational amplifier with adequate DC gain and gain-bandwidth, as well as adequate bypassing on the power supply rails, to ensure a solid ground in spite of crossover distortion.

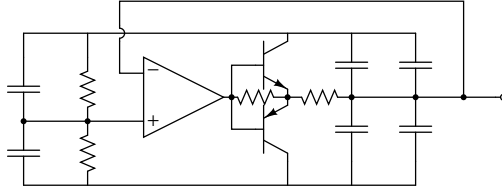


Figure 5: This circuit is used to generate a ground rail between the 0V and 5V rails provided by the Mini digital board. In most applications, the transistors would be left unpopulated, and the output resistors would both be  $0\Omega$  jumper resistors.

- Most regulators are designed to either source or sink current, but not both.
- The prototype was built the board from parts available on-hand.

The circuit may also be run with external power rails, for either cleaner power supply rails, or for more headroom. In this case, the regulator circuit is entirely omitted when populating the PCB. The circuit is designed to be compatible with rails of up to  $\pm 15\text{V}$ , but when operating with higher rails, it requires a small level-shifter and protection circuit between the main Mini board and the parameter analyzer. In addition, for rails above  $\pm 8\text{V}$ , the parameter analyzer board must be populated with the MAX335 switches instead of the MAX395 switches.

## 6.2 Parts list

The parts required for the reference version of the parameter analyzer are shown in Table 6. Total cost is approximately \$15.40 in individual quantities, and \$7.80 in quantity, excluding PCB and assembly costs. This is for the reference implementation, which is designed to be minimum cost. The board may cost slightly more if assembled for high-current or high-voltage uses. Possible changes include:

- Better operational amplifiers.
- MMBT3904 and MMBT3906 driver transistors for the middle power rail.
- More, bigger bypass capacitors.
- MAX335 instead of MAX395.

## 6.3 Status

The first iteration of the parameter analyzer board is completed and tested. There is one bug – the reset pin on the MAX395 is connected to the wrong rail. This is easy to manually

Qty	Part	Reference Value/part	Location	Cost
1	Switch	MAX395	All SMT ICs	4.87 (2.22)
7	Dual op-amp	NJM62	DIP IC	0.37 (0.29)
1	10-pin header connectors			2.02 (0.89)
1	20-pin header connectors			2.39 (1.13)
10	bypass capacitor	0.1uf	C1-C10	0.12 (0.06)
2	resistors (voltage divider for $v_{gnd}$ )	10k	R1 R2	0.05 (0.02)
4	shunt resistors	280 $\Omega$	Rn4	0.05 (0.02)
8	resistors for front of diff amp	4.12k	Rn1, Rn8	0.05 (0.02)
8	resistor for back of diff amp	2.05k	Rn0, Rn9	0.05 (0.02)
12	divider resistors, top	10k	Rn6, Rn2, Rn3	0.05 (0.02)
8	divider resistors, bottom	5.1k	Rn5, Rn7	0.05 (0.02)
4	jumper resistors	0 $\Omega$	J1, J2, R3, R4	0.05 (0.02)

Table 6: Bill of materials for the parameter analyzer daughter-board. Total cost is approximately \$15.40 in individual quantities, and \$7.80 in quantity, excluding PCB and assembly costs. Prices are taken from Digikey, where available. The NJM062 is purchased through Mouser, and the MAX395 through Maxim Direct.

jumper on the board. Otherwise, the board is functional, although there is a large number of possible functional improvements.

The low gain of the amplifier, reduced loop gain, and large shunt resistance contribute to give the circuit a small, but noticeable, output resistance in voltage output mode in certain configurations of the circuit. The output voltage, therefore, does not perfectly track the desired voltage. The effect of this is not huge, since the user can plot the measured output voltage, rather than the desired voltage. Nevertheless, one of the tasks remaining is to compensate for this in firmware – in addition to the analog feedback loop, the firmware should implement a digital feedback loop which compares the desired output voltage to the actual output voltage, and adjusts  $v_{in}$  to compensate.

Operational amplifiers are currently grouped by SMU, rather than by function. This allows boards requiring fewer than four SMUs to be assembled omitting parts for the additional SMUs. Overall, this is a poor tradeoff, since the design requirements for the three operational amplifiers are different. IC1, when testing power devices, needs to be capable of sourcing large currents. IC2 needs to be a FET-input device, since any bias current will translate to current measurement error. On the other operational amplifiers, the input bias currents simply need to be relatively matched. IC1 and IC2 preferably use rail-to-rail devices. Since the current board design uses dual operational amplifiers, and groups them by SMU, it does not allow the use of different amplifiers for each stage.

Better implementations for the regulator for the ground rail are possible and should be investigated.

The layout of the board requires some clean up. In particular, standardizing the layout of individual SMU ports would be simplify assembly.

## 7 Software Architecture

The Mini is designed to work with the MIT iLabs software architecture. MIT iLabs is an architecture and associated set of protocols for implementing an sharing on-line laboratories. The iLabs architecture consists of three major components. A lab server interfaces to hardware. It verifies and executes experiments. A client provides a graphical interface to control the experiment. A service broker sits between the client and lab server, and is primarily responsible for authenticating users. Each lab server may communicate with multiple service brokers, and each service broker may communicate with multiple lab servers. The service broker is separate from the lab server for a number of reasons. Most importantly, with shared experiments, each university has a local service broker, and so universities can manage their own users – the school hosting the experiment does not need to manage user data for remote universities. The service broker also allows the system to provide a more unified user interface to multiple lab servers, reduces inter-university bandwidth consumption, and simplifies administration. In most cases, each university will only need one service broker, but may host many lab servers.

MIT provides reference implementations for all three pieces of software. The reference iLabs lab server provided by MIT requires an expensive software stack, and complex installation and maintenance. As a result, it is inappropriate for use for either light-weight laboratories, or in the developing world. For the Mini, a preliminary lab server was developed whose goals are:

1. Low-cost.
2. Ease of installation, deployment, and maintenance.
3. Ease of development.

These place some constraints on the flexibility and power of the server.

### 7.1 Cost

One of the primary goals of developing a new lab server was to reduce cost. The MIT reference lab server requires a software stack consisting of:

Product	NEWEGG.COM price	Academic price
Windows Server 2003	\$690-\$2400	\$609-\$2319
SQL Server	\$710-\$1625	\$479-\$8449
Visual Studio	\$270 (Std) \$700 (Pro)	\$50 (Std) \$129 (Pro)

In Nigeria, the per-capita GDP is \$632/year<sup>9</sup>. Using the original lab server would require one of several approaches. First, a Nigerian school may spend roughly double the annual GDP/capita on software. Second, as is now common, the school may use illegal copies of the

---

<sup>9</sup>Source: CIA World Factbook

software. Neither of these approaches is appropriate. Finally, the school may use donated copies of the software. This approach does not scale.

The new lab server developed uses an entirely free software stack. It is written in Java. It can run on the GNU/Linux operating system, although it is Windows-compatible as well<sup>10</sup>. It does not, at present, require a database. If database functionality is eventually added, it will be added using the free Berkeley DB database.

The new lab server also significantly reduces hardware costs. The Java server can run on a system with much lower system requirements. SQL Server alone requires 512MB memory, and recommends 1GB. In contrast, the iLab Mini lab server consumes approximately 20MB on top of the operating system requirements – under Debian GNU/Linux, it ought to be completely functional with roughly 64MB RAM. Indeed, the development machine for the Java lab server is a Athlon/750 with 384MB of RAM, purchased used for \$20, running a number of unrelated servers at the same time.

## 7.2 Ease of installation and maintenance

The iLabs reference lab server requires specific versions of Windows, IIS, and Microsoft SQL Server. As a result, it is highly constrained with regards to what machines it can run on. In most cases, this results in a dedicated server configured specifically for iLabs use. An existing machine used for a different task cannot be easily re-purposed to also host an iLabs lab server. The installation process requires a significant amount of time and expertise to properly configure and integrate these pieces of software. Installation is a fairly lengthy and tasking process. Maintaining the machine then requires managing the security of IIS, SQL Server, as well as the iLabs-specific code. This prevents many of the types of possible uses of iLabs.

The lab server developed for the Mini, in contrast, requires nothing beyond Java. The web server used for web services is embedded and runs within the Java sandbox. As a result, the system is much easier to install and configure, and less prone to security issues.

This limits functionality somewhat. In contrast to the reference implementation, the lab server cannot currently be administered through a web interface. In addition, since there is no database, if the server needs to be restarted, the experiment queue is lost, and students need to resubmit experiments.

## 7.3 Ease of development

Developing traditional lab servers around the MIT reference lab server is complex. One goal is to see lab servers developed by students at universities in Africa. These students' skill level is generally somewhat behind that of their American counterparts. As a result, we wanted to design our software infrastructure to be as simple to develop with as possible. The new lab server achieves this in two ways:

---

<sup>10</sup>Running under Windows would require some trivial changes to the hardware-interface code.

- Lower learning curve – the new lab server requires developers to know fewer technologies than the reference lab server
- Better abstraction – the web services code is hidden from most developers.

The traditional lab server requires developers to learn a thick stack of technologies:

- C#
- VisualBasic
- SQL and Microsoft SQL Server
- ASP and IIS
- Web services
- XML
- Extensive knowledge of the iLabs architecture

Most students do not learn many of these technologies as part of a normal curriculum. In contrast, the lab server developed for the Mini requires:

- Java
- Preferably, some XML (not required)
- Fairly minimal knowledge of the iLabs architecture

Java is the most common introductory programming language in American universities. It is one of the more common outside of America, although C/C++ is more common in both Africa and China. Java is taught to computer science students in the computer science departments of both Zhejiang University in China, OAU in Nigeria (although electronics students at OAU are only taught Fortran). Since knowledge of Java is required for the client, learning Java is a necessary step to developing iLabs regardless of choice of server language. Eliminating the dependence on C# reduces the number of languages students need to learn.

In addition, Java is cross-platform. Using a cross-platform language opens iLabs development to users of GNU/Linux and MacOS/X. While Windows is the most common platform at both African and Chinese universities, the upper education system in India strongly favors the use of GNU/Linux and free/open source technologies. The ability to function under GNU/Linux is likely a requirement for cooperating with many institutions there.

Next, the use of web services is hidden from the user in a library. A basic lab server can be implemented with no knowledge of either XML or web services. A simple lab server is shown in figure 7.3.

This server will return the experiment in lower-case letters. More advanced functions can be added to the server by overriding more function calls.

```

import ilab.server.*;

class MiniServer extends iLabServer {
    public MiniServer(String confFilename) throws IOException {
        super(confFilename);
    }

    public String runExperiment(String experiment) {
        return experiment.toLowerCase();
    }

    public static void main(String[] args) throws Exception {
        MiniServer s=new MiniServer("/etc/ilabconfig.properties");
        s.go();

        // Work-around for platform-specific bug that causes Java to
        // terminate before all threads finish.
        while(true){
            Thread.sleep(1000000);
        }
    }
}

```

Figure 6: The source code to a basic lab server

## 7.4 Status

The Java lab server is functional but not yet production-quality. The remaining problems are:

- The security infrastructure is incomplete
- It currently uses HTTP instead of HTTPS for incoming communications. The architecture supports HTTPS trivially, but this has not been implemented.
- It has not been tested whether or not it properly calls NOTIFY on the service broker, as required by the MIT iLabs specification. The lab client operates normally if this call is unimplemented.
- It is very approximate in the way it estimates time remaining for labs in the middle of the queue
- Logging functionality is not implemented
- General clean-up is required
- Some configuration parameters need to be moved into the configuration file
- An easy installer/uninstaller needs to be written

The specific implementation for the Mini parameter analyzer currently has fairly basic functionality. It parses XML input from the client, performs the associated experiment, and sends data back. It sends more data than is necessary (from all ADC channels). It does not have very good fault recovery if the firmware gets out-of-sync with the lab server. It uses a thin platform-specific layer to interface to the Mini that should include more error-catching code to be robust. Unit conversions are not yet calibrated.

A number of fundamental tasks also remain:

- Better memory management.
- Similar abstraction of client-end code. This code should allow people to develop clients without knowing web services. In addition, it should allow the development and testing of labs without access to a service broker<sup>11</sup>.
- Integration of Berkeley DB, an in-memory database, for robustness.

---

<sup>11</sup>In the iLabs architecture, the SB is a server that sits between the client and server, and manages users and security. It is necessary for deployment, and especially, for sharing labs among schools.

## 7.5 Bandwidth Limitations

As mentioned, the MIT iLabs protocols were originally designed for use in the developed world. As a result, the protocols are extremely bandwidth inefficient. Experiment information is transmitted in plain text format, wrapped in XML, wrapped in web services transactions. Each layer substantially increases bandwidth. Running a typical experiment requires about 30kB of data sent between the client and the server. Simply compressing this data with gzip would reduce bandwidth requirements by a factor of three. A protocol designed to be bandwidth-efficient protocol could reduce it much further.

Work on improving the bandwidth of iLabs experiments has been in progress for some time. The addition of the service broker allowed the client download and the web accesses preceding it to be moved from the institution hosting the experiment to the institution where the experiment is run. Colleagues at OAU are currently investigating ways of transparently compressing the web services transactions to further cut international bandwidth.

## 8 Conclusion

This paper has shown solutions to some of the equipment problems facing the electronic engineering departments of universities in the developing world.

The use of remote electronics experiments can give students the opportunity to experience the difference between theory and reality. Students can see the effects of noise, of parasitics, of instrumentation limitations, of device variations, and of imperfect models. While students cannot yet use iLabs for circuit design labs, work on reconfigurable iLabs experiments may bring that into existence in the near future. Nevertheless, iLabs will most likely never be able to capture circuit construction skills, such as soldering, debugging wiring errors, proper use of ground planes, or neat wiring.

The design of lower-cost, lower-performance instrumentation can have a dramatic impact on the educational system there. It has the capability to enable course faculty to easily deploy many low-cost on-line laboratories for student use. In addition, it may also broaden access to conventional laboratories.

## Appendix A: Mini Assembly

The first step in assembling the Mini is to solder the Mini. The digital parts of the Mini are always USB-powered. The ADC and DAC may, optionally, use an external power supply for cleaner analog power. If the entire board is USB-powered, jumpers J4 and J5 need to be soldered. If it uses a separate power supply, they should be omitted. If the microcontroller is to be clocked from the USB chip, jumper J3 should be soldered, while J1, J2, C3, C4, Q1 should be omitted. If it uses a crystal, then J1, J2, C3, C4, Q1 should be included, while J3 should be omitted.

Once the Mini is assembled, the FT232 chip needs to be programmed. FTDI provides a utility called MProg to do this. There are several settings which may need to be changed:

- If using the 6MHz on-board clock, C0 needs to be set to CLK6
- The device should always be set as bus powered. The power requirements should be set appropriately to the application (or 500mA, for general purpose use).
- On some chip versions, USB remote wakeup, and plug and play should be disabled (some chips may not include these settings).
- The remaining CBUS pins may either be left unconfigured, or configured appropriately for specific applications.

Once the FTDI chip is programmed, the microcontroller must be programmed.

### A.1 Programming the ATMega48

The Mini uses a standard 5-pin AVR hobbyist programming connector for programming the ATMega48. This is a single-row connector with the pins RESET, SCK, MISO, MOSI, and GND. The Mini may be programmed through any interface that can toggle 3 output pins and read 2 input pins at 5V TTL levels.

The simplest programmers simply use the control lines from a parallel port connect directly, as shown in Figure 8. The control lines from a serial port may be used as well, but require level shifters to convert the  $\pm 12V$  down to 0-5V signal levels, as shown in Figure 8. The 0-5V output of the MOSI port is technically inside the RS232 noise margin, but empirically works with most modern RS232 controllers.

In addition, a number of USB programmers exist. The USBTiny<sup>12</sup> programmer is the simplest. It consists of an ATTiny2313, 12MHz crystal, resistors, and capacitors. The USB stack is implemented in firmware. This programmer is only compatible with some USB controllers, but where it is incompatible, it can often be used through a USB hub. A schematic of the USBTiny is shown in Figure 8. Several programmers exist that use an FTDI chip and an Atmel microcontroller. These have a number of advantages:

---

<sup>12</sup><http://www.xs4all.nl/~dicks/avr/usbtiny/>

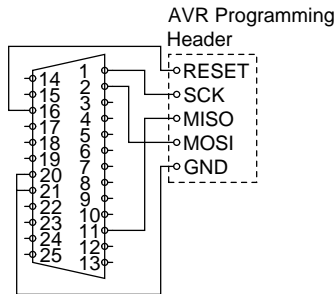


Figure 7: Mini parallel port programmer schematic.

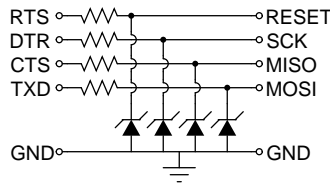


Figure 8: Mini serial port programmer schematic. All zener diodes are approximately 5V. Note that this does not level shift back to RS232 levels – this is compatible with all serial ports we have tested on, but is outside of RS232 spec, so may not be universally compatible.

- They are compatible with all USB ports
- The firmware may be easily ported to the Mini, so that one Mini may be used to program others
- The most sophisticated<sup>13</sup> implements the full Atmel STK500 specification, and so may be used to calibrate the Atmel’s on-board oscillator.

The official Atmel programming interface has an additional  $V_{DD}$  pin that is used to differentiate devices running at different voltages. Atmel-made programmers can be used with the Mini by using the 5V pin from the interface header. An official Atmel USB programmer is available from Digikey for 34 dollars under part number ATAVRISP2-ND.

All of the programming pins have 470Ω protection resistors – although the Atmel microcontroller is fairly resilient to current overflow condition, this prevents potential damage to fragile parallel and RS232 ports potentially used for programming the Mini.

<sup>13</sup><http://www.tuxgraphics.org/electronics/200510/article05101.shtml>

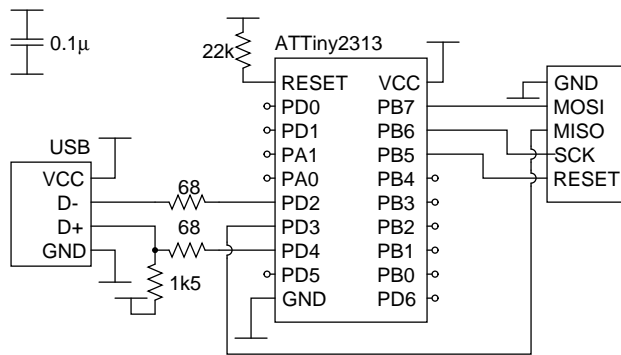


Figure 9: Mini parallel port programmer schematic.