Digital Signal Processing Laboratories in Timisoara
Department of Computer and Software Engineering
POLITEHNICA University of Timisoara
2, Vasile Parvan Bv., 300223 – Timisoara, Romania
Tel: + 40 256 403271, Fax: + 40 256 403214
Http://dsplabs.cs.upt.ro/

PROJECT REPORT

SUCEAVA Hard&Soft

2012 Edition

RONIN. Remote Observation and learNing experImeNts

Team : Timisoara I
Members : Victor ADASCALITEI
Andrei-George STOICA
Adrian Horia OAIDA
Zsolt BIRO

Digital Signal Processing Laboratories in Timisoara
Department of Computer and Software Engineering
POLITEHNICA University of Timisoara
2, Vasile Parvan Bv., 300223 – Timisoara, Romania
Tel: + 40 256 403271, Fax: + 40 256 403214
Http://dsplabs.cs.upt.ro/

PROJECT REPORT

Summary :

This document covers the principles that were considered when designing and building the RONIN laboratory environment (former "STUD*i* Lab" project which was dropped for marketing reasons). More specifically, we, the developers of RONIN, were driven by the following goals :

1. building a remote laboratory environment so that the students, who are also the audience of this service, can benefit from real-life-laboratory knowledge without the risk of damaging expensive laboratory equipment
2. building a project that is, above all, eco-friendly both in the recycled materials used and the power ratings involved
3. building such a system under modern economical constraints : developing it as low-cost as possible but not lower
4. building a system with off-the-shelf parts that can be easily put together an configured
5. making a system that is as easy to use as possible
6. spreading the concept and integrating it into the excellent Moodle[1] academia CMS

And last but not least, our main goal in building RONIN was to bring good educational knowledge and make it available in all the corners of the world from the comfort of the Internet infrastructure.

Now that we've seen our goals and aims for this project, let's go into some more engineering details.

System Architecture :

The project was build and organized into 4 sections :

1. A data acquisition module built around an mbed[2] device
2. Several lab experiments that acquire, communicate and distribute data through the mbed front-end
3. A Moodle site page to make the service comfortably available on the Internet
4. An Android application that carries the RONIN concept through the mobile users segment

Each of these modules were tightly coupled to give the user the feel of an actual experiment environment with the addition of a much simpler configuration settings. This can be seen from the ease of use with which the whole system can operate.

Having this said, we'll take each "subproject" and analyze it in more details.

mbed at the heart of the system … :

Every project has a core, and RONIN is no different. At the very core of the RONIN project there is a mbed microcontroller. Why did we choose mbed? There were several reasons :

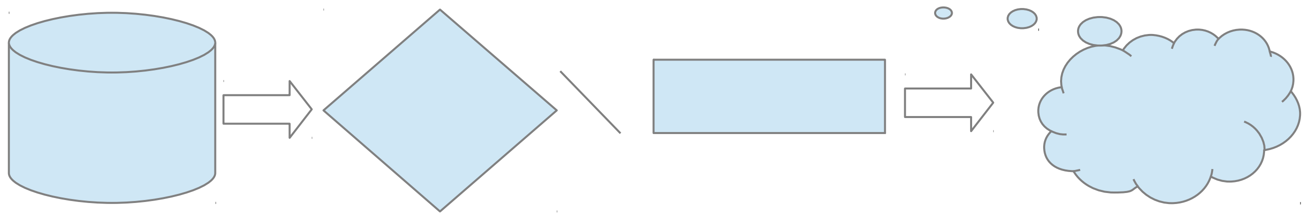
1. it is a cheap device that has a great "value for price" index

[1] The site's homepage : http://moodle.org/
[2] More details about this microcontroller can be found here : http://mbed.org/

2. it has the capability to undertake all kinds of jobs that a virtual laboratory might through at it
3. it is easy-to-use and program and has a great community of users
4. it is self-contain having all kinds of built-in modules ready to be interfaced with the outside world to read it's parameters

All these points helped us conclude that mbed was the perfect tool for the job and that it would require as little external components as possible yet still live up to it's name. In this regards, the economical aspects could also be tempered with.

Schematically, mbed only needed a RJ45 jack to connect to the Internet. It's layout system can be inspected from the following image :



Lab Apps mbed Ethernet Jack Internet

If all the above mentioned items look familiar, this is not the case for the “Lab Modules” block. Let's look into this a little bit more carefully.

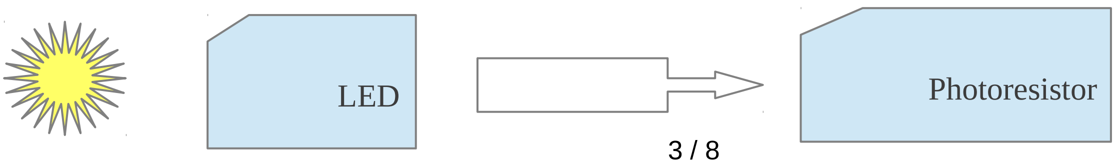
Some laboratory experiments supported by RONIN :

As we previously mentioned, RONIN was built for making real-life-laboratory experiments safely available for the masses. We took the liberty of making some simple laboratory examples. And so, 4 laboratory projects were constructed :

1. An example consisting of a simple voltage divider principle
2. A Whitestone bridge experiment
3. A feedback controlled temperature environment
4. An 8 bit R-2R DAC ladder

First off, to control various circuit parameters, a means of parameter modification was in order. Since capacitor-control or inductive-control was out of discussion (for economical reasons). A some-kind of resistor-controlled loop was searched for. There were several alternatives : to buy one (rather expensive) or to try to build one from scratch. The latter was choose. What we came up with was a PWM controlled photo resistor that could change it's values based on the duty cycle of the input wave.

The following picture captures the thought :

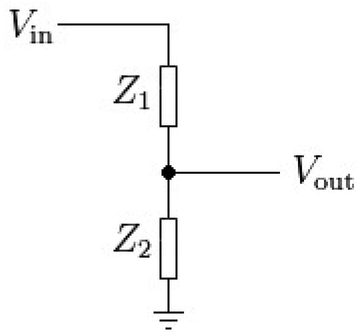


The idea was good as we investigated the concept and found out that the values that could be obtained with such a system were continuous and, in some regards, better then commercial alternatives. There were some difficulties encountered on the road of learning these custom-made electrical pieces :

1. we had problems with the stability of the components (solved later by “welding” the parts to the PCB)
2. thanks to the nature of the photo resistor, the resulting domain was not continuous for all the combinations of inputs applied (solved through incremental analysis techniques)
3. the outputted resistance exhibited a capacitance behavior with a short rise (cca. 200MV p-p) during a transition phase

Let's look at each provided experiment a little bit more closely.

The first experiment in which the new component was used was the humble “voltage divider” :

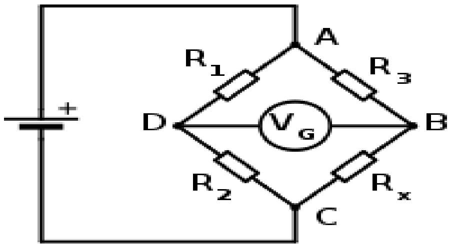


Z1 was a newly created resistor-led pair while Z2 was a fixed experiment parameter.
The voltage obtained through such a device is given by the following relationship :

$$V_{out} = \frac{Z_2}{Z_1 + Z_2} \cdot V_{in}$$

Having the nature of a photo resistor, Z2 exhibited a variable resistance controlled by the MCU. As a user, you could select a desired voltage and the system would “discover” the value of Z2. Reversely, one could select a desired resistance and find out, empirically, how the output voltage varies in this regards.

- Next off was the Wheatstone bridge. A Wheatstone bridge, as wikipedia protrais it, represents a “electrical circuit used to measure an unknown electrical resistance by balancing two legs of a bridge circuit, one leg of which includes the unknown component.”



It's general output schematic looks like the following :
mbed utilized 2 of it's output pins to actually measure the voltage of the voltmeter points. The applied theory was simple : read the 2 data voltages and adjust the resistance so that, through KVL[3], the difference between the 2 points tends to decrease to zero.

In this experiment, the user can choose either to calculate the voltage of the bridge given a resistor value for Rx or to find the resistance given that we manually fix in the value for Rx.

Although these 2 experiments might seem trivial, mbed actually doesn't use a table of values that might match a resistor to a PWM duty cycle as required. Insted, mbed carries out the actual voltage/resistor finding by incremental analysis of the PWM duty cycle along with the model for the circuit.

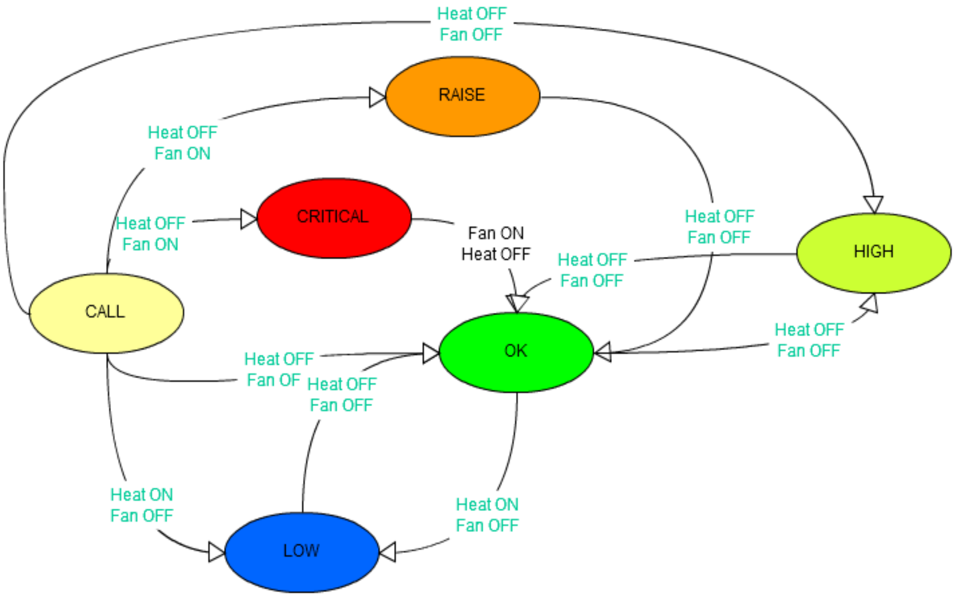
For instance, say that we want to find a voltage of the voltage divider given a known resistor. What mbed will do is the following : it will first iterate the PWM duty cycle from 0.0 to 1.0 in 0.001 increments. For each generated value, it will inspect the outputted value of the circuit and, working backwards, it will approximate the resistor value for the divider.
In this case, the actual formula used would be :

$$R_x = \frac{(R_{known} * V_{in})}{V_{out}} - R_{known}$$

If the calculated resistor is in within some close value around the initial resistance, the iterations will stop and the final result will get printed.

The next experiment uses an oven-like environment to generate and control a temperature in an closed, isolated system illustrating a basic feedback control system. The temperature is generated through a passing current in a 1W, 10Ohms resistor while the information is gathered through a 10mV/degree Celsius. The whole assembly is placed in a close proximity to each other and decoupled from the rest of the system through it's own 3.3V voltage regulator. Besides these 2 elements there is also a fan that is used to extract the heat faster and bring the system to room temperature.

[3] Kirchoff's Voltage Law (http://en.wikipedia.org/wiki/Kirchhoff%27s_circuit_laws)



The whole logic of this experiment is build around a complex state machine architecture that is illustrated in the above image.

Each and any of these elements are controlled directly by mbed through general purpose switching transistors, type 2N2222 to be more exact.

For this experiment, the user can inspect in real-time the temperature of the environment for which various configuration options can be set. These options are : the desired temperature, the sampling rate, the hysteresis margin and the delta critical temperature for which the fan kicks in to rapidly extract heat from the system.

Next, things get a little bit more interesting : our last provided experiment consists of **a home built R-2R ladder DAC**. Basically, we use 8 pins from mbed to push and generate analog values. Since the ladder itself isn't capable of driving big loads, an op-amplifier in buffer mode is used to repeat the signal and feed it through a speaker making it "more" real.

The user can use this experiment to learn about R-2R networks and play a little bit with it by generating sine waves that can be actually heard.

Now that we've seen some RONIN capabilities, let's move a little bit further and talk about …

RONIN, as users see it …

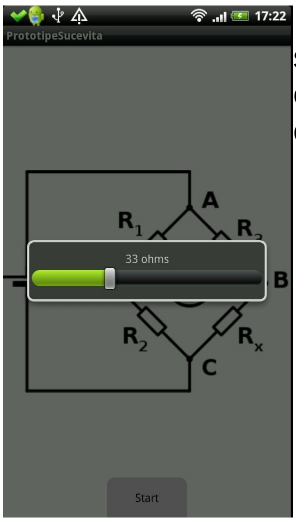
Well, it's a Moodle platform, what did you expect ? It looks like any other Moodle site.

Apart from that, RONIN is integrated into the platform as a plug-in for the site. The user navigates it's way through the interested course and undertakes it's interested experiments in 2 easy and self-explanatory steps :

1. it first configures the parameters for the experiment *and*
2. secondly it starts the experiment.

Now that the web-page is covered, our ever-evolving-world demands that an Android application needs also building. Well, we know and we've done it!

RONIN into the cloud …



The mobile front-end is actually a Java app that communicates "behind the scene" with the Moodle server and mimics it's behavior only in a more expropriate manner : it only displays the available experiments and lets you choose, from each experiment, the customizable parameter(s).

The next picture illustrates this best :

Thus, by this feature, the experience is complete! Moodle is up and kicking.

Conclusions

The RONIN project was a daunting one for us. Our engineering skills were tested to the limit. We covered areas such as electronics, mechanical engineering, programming skills, mathematical skills, physics, DSP, web programming, mobile programming, and others.

RONIN and the idea of a remote laboratory in general can be made into a successful product. This is especially true in these interesting times when successful academia experiments such as Coursera[4] or MITx[5] showed that there is a vivid interest in the area.

[4] See more here : http://coursera.org/
[5] Read more here : http://mitx.mit.edu/