EE318 Electronic Design Lab Project Report, EE Dept, IIT Bombay, April 2009

Wireless DSO Modules

Group 6

07007016, Ameya Damle, *ameyadamle@iitb.ac.in*
07d07009, Sai Kulkarni, *saikulkarni@iitb.ac.in*
07d07010, Chiraag Juvekar, *chiraag@iitb.ac.in*

Project Guide: Prof. Dipankar

“Rumor has it that analog circuit design is dead. Indeed, it is widely reported and accepted that rigor mortis has set in. Precious filters, integrators, and the like seem to have been buried beneath an avalanche of microprocessors, ROMs, RAMS, and bits and bytes. As some analog people see it (peering out from behind their barricades), a digital monster has been turned loose, destroying the elegance of continuous functions with a blitzing array of flipping and flopping waveforms. The introduction of a ”computerized” oscilloscope- the most analog of all instruments - with no knobs would seem to be the coup de grace.”

- Jim Williams, "Analog circuit design - Art, Science and Personalities"

Abstract

The project aims at creating wireless DSO modules for measuring voltage signals, observing signal variations and waveforms and monitoring them from a remote location. The goal was to provide a multichannel, high bandwidth device having isolation and synchronisation between the channels. There are two major functional partitions in the operation:

Capturing the Signal: In order to measure the signals, the input analog signal is first amplified/attenuated (based on the magnitude) using the programmable amplifier block. This amplified signal is then sampled at a maximum sampling rate of 200KSPS, when the trigger signal (enable) is ON. This selectively sampled signal is fed to the microcontroller where it is buffered and transmitted in bulk via the RF link on request.

Control of parameters: The control of parameters, such as ADC sampling rate and amplifier gain is done from the base station. For this, a custom protocol is implemented between the base station and channel modules. The base station transmits the instructions via the RF link. The channel module IDs are incorporated in these instructions and based on those the corresponding modules respond accordingly.

Table of Contents

Introduction

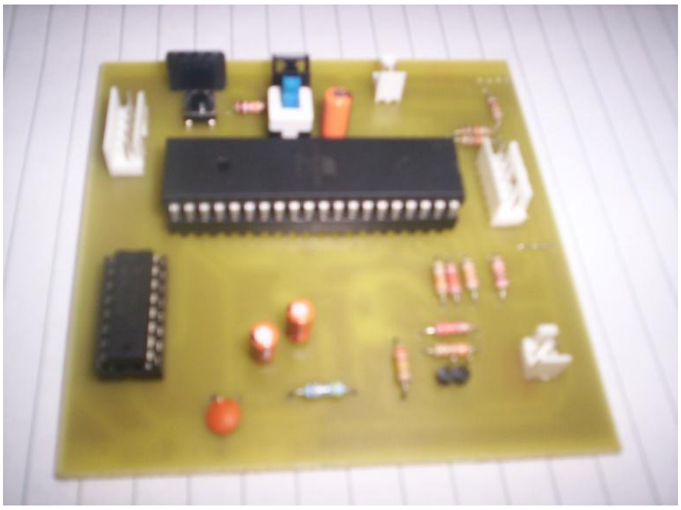


Fig. 1: Picture of the Final Board

In our project, we have created Wireless DSO modules with a PC based base station. The wireless modules act as signal capture and processing devices while the base station displays the signal waveform on the PC monitor.

Rationale and Motivation behind the Project

Anybody who has ever used a CRO or a DSO has faced the following problems:

1. Slightly tricky operation and control
2. The bulkiness of the device makes it difficult to use
3. Isolation between the two channels is not provided

This helped us come up with the idea of Wireless DSO modules.

1. Being PC based, the usage is very simple and handy
2. Being light and wireless, the DSO modules are portable.
3. Being wireless, an operator can monitor multiple distant signals and compare them just sitting at his work station in almost real-time.
4. Since these DSO modules provide isolation between two channels, we can measure and compare independent signals. The RF link also provides isolation between the PC and the modules.

High Level Description

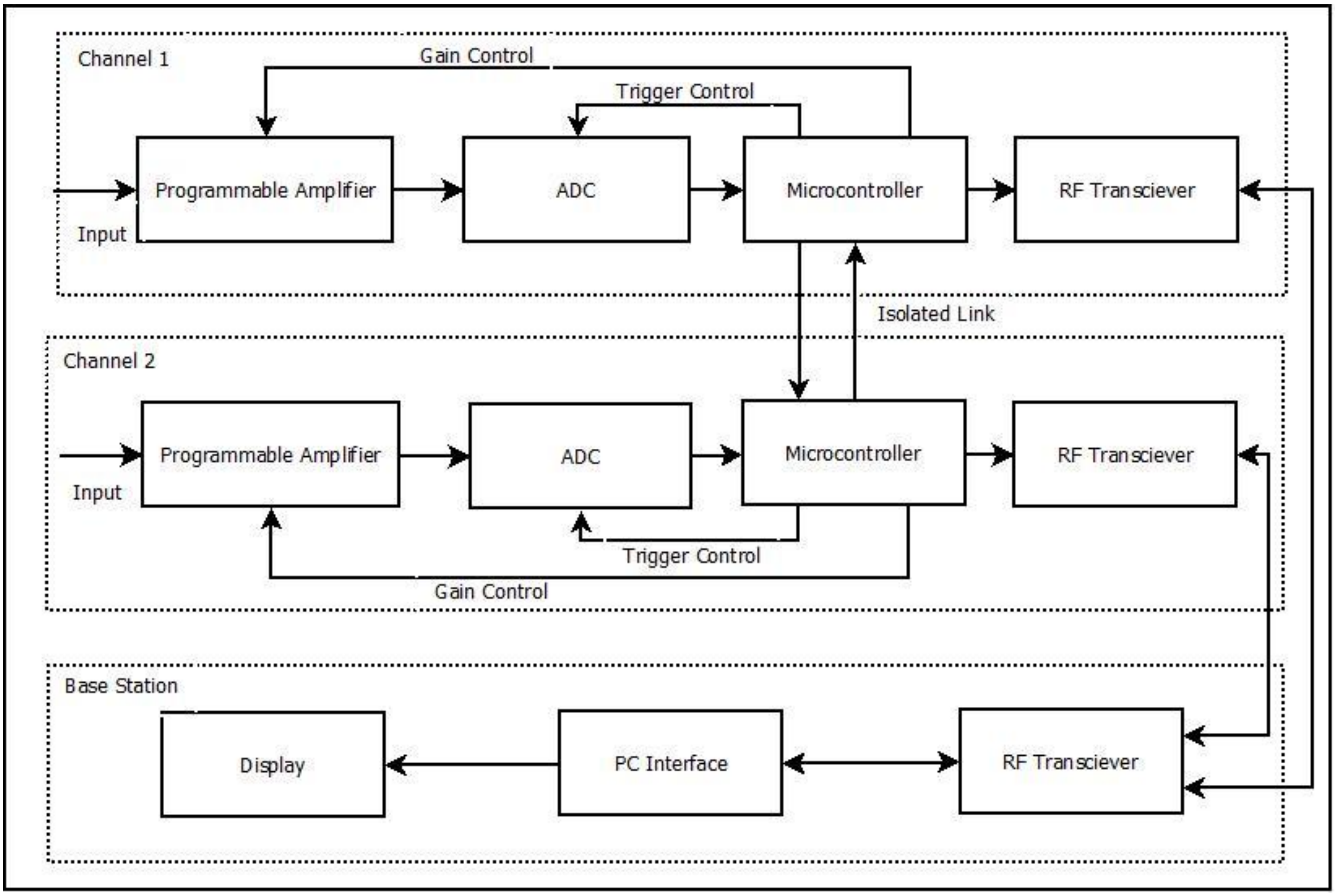


Fig. 2 Block Diagram of a 2 channel DSO unit

The two major components of a DSP module are the wireless module and the base station. All of the processing such as triggering, gain control, scaling, etc is done on the signal receptor module. The base station module is merely an interface between the DSO modules and the PC.

The GUI is provided with user control facilities to control the display scales, the trigger level and changing the trigger channel. Whenever the user changes the operating settings, the base station unit sends the corresponding instruction in the customized protocol via the TF channel. According to the instruction sent, the corresponding wireless module responds. The triggering in two channels is synchronised via the isolated link between the two muCs.

The muC controls sampling rate of the ADC, triggering and the gain of the programmable amplifier. According to the operating settings, it sends the processed bulk data to the base station where it is displayed through the GUI.

Hardware Description

Programmable Gain Amplifier:

The programmable gain amplifier stage is a very simple OP-AMP non-inverting amplifier assembly in which 4 options have been provided to vary the gain, which are controlled by the microcontroller.

The gain of the amplifier shown in *Figure 2* is:

$$A = \left(\frac{R_{selected}}{R_{selected} + R_p}\right)\left(1 + \frac{R_f}{R_i}\right)$$

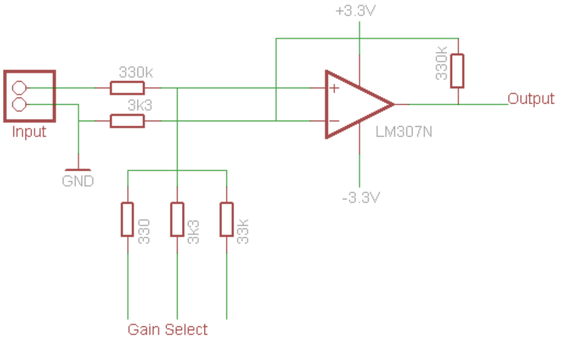


Fig. 3: Programmable Gain Amplifier Schematic

In the above equation, $R_{selected}$ is the resistor combination selected from among 330Ω, 3.3kΩ and 33kΩ. Additionally we set it to inifinity by leaving all three open.

$$A = 101 \times \left(\frac{R_{selected}}{R_{selected} + 330{,}000}\right)$$

The structure of the amplifier is such that when a specific resistor is selected that terminal is grounded. By default all resistors are left open. An easy control is possible because the uC port pin can be set into high-Z mode and can be driven high and low in an output mode.

The following gain settings are possible as a result of the control:

Resistor Selection	Desired Gain	Actual Gain
330Ω	0.10	0.10
3.3kΩ	1	1
33kΩ	10	9.18
None	100	101

Op-Amp Selection for the programmable gain amplifier:

The following criterion was used when selecting an Op-Amp for the programmable gain amplifier:

1. Speed of the Op-Amp (Slew Rate)
2. Dual Supply with 3.3V being inside the V_{dd} range
3. Ease of availability and feasible package for board design and assembly

The final op-amp chosen was TI's OPA725. The maximum frequency of operation targeted was 50kHz.With a maximum input range of -3.3V to +3.3V. This corresponds to a maximum slope of 2V/µs. The OPA725 has a slew rate of 30V/µs and thus meets our speed requirement. Dual supply Op-amp was required and the power supply constraints allowed only a maximum voltage of -3.3V to be generated. OPA725 has a rail to rail supply range of 4-12V. As such our range of 6.6V was within the range. Finally the device is available in convenient 8-pin SOIC package as a sample from Texas Instruments.

ADC:

The ADC selected is at the heart of this project. The project targets a maximum of operating frequency of 50 KHz. To meet this requirement we need to sample around 500 KHz. The task of finding an ADC was primarily decided by the above factor.

Criterion under consideration for selection of ADC:

1. Final instrument requires 2 analog channels, preferably with independent ADCs
2. Ideally low pin-count devices with leaded packages (SMT packages preferred)
3. Local availability (less than 5 day delivery time in case of external orders)
4. Preferably serial data output
5. Sampling rate of the order of 1 MSPS

ADC comparison table:

Device	Manufacturer	Vendor	Resolution	Max Sampling Rate	Package
ADS7884/5	TI	Sample	10	3 MSPS	6SOT/23
ADS900	TI	Sample	10	20 MSPS	SSOP28
THS1041	TI	Sample	10	40 MSPS	SSOP28
ADS5522	TI	Sample	12	80 MSPS	TQFP64
ADC10080	NI	NA	10	80 MSPS	SSOP28
ADC1175	NI	NA	8	20 MSPS	SSOP28
AD9203	Analog	NA	10	40 MSPS	SSOP28

1. Vendors looked at are TI, National Semiconductors and Analog Devices. Preferred Vendor: TI due to ability to order sample within 4-5 working days Other vendors only to be chosen in case of significantly better devices which were not found.

2. If everything else works out as planned, increasing data rate is not a significant challenge. Hence in case of a compromise, data rate is a less important factor.

3. Given a time and resource constraint, ease of use and lack of need to prototype and test is very important. Hence QFN packages shall not be used. Similar grounds were used to discard TQFP-44 packages too. Prefered packages are all leaded packages with less than 28 pins.

4. Serial communication is preferred in SPI or I2C as there is no learning curve.
Parallel communication is given second priority in order to simply board layout.
All proprietary data format have been discarded foreseeing unnecessary complications later

TI THS1040 was chosen. A brief overview of its features:

1. 40 MSPS sampling rate
2. 2V peak to peak inout range
3. Internal reference generation
4. 10 bit ADC with overflow detection

The ADC was first tested on a separate module and was then incorporated into the main board.

The ADC has a parallel interface and the final circuit for the ADC can be seen in the adjoined figure. Note that the input has a simple low-pass filter of the appropriate band-width

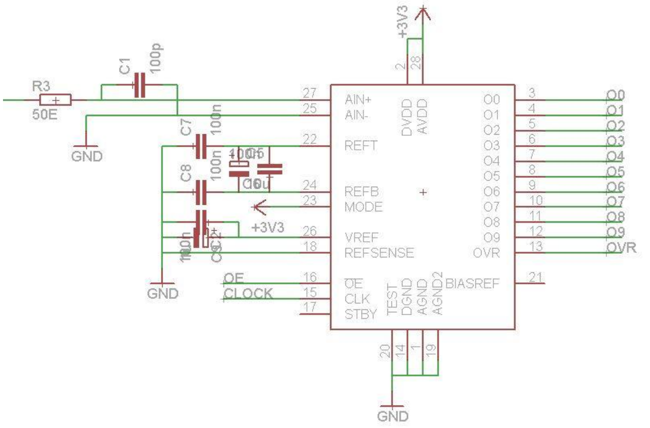


Fig. 4: ADC Module Schematic

Isolated Link:

There are 2 communication links in the device:

1. PC interface – Wireless Module Link
2. Wireless Module – Wireless Module Link

By virtue of the fact that the first link is a RF link isolation is automatically implemented for that link. The link between 2 modules is however electrical and hence isolation is necessary. In our case one module (triggering module) generates the ADC clock on which the ADC samples the data and notifies the other uC as soon as its channel triggers.

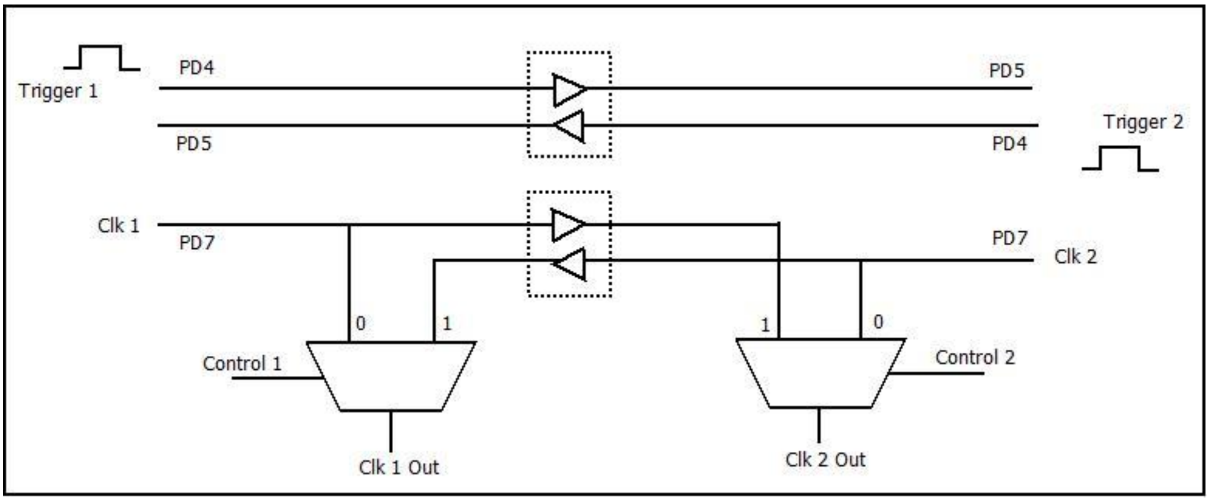


Fig. 5: Isolated Link Block Diagram

The link between 2 modules is necessary to communicate the following information:

1. Transmission of the triggering clock
2. Assertion of triggering so that the secondary module can start storing data.

TI ISO7221 is the device chosen to implement this isolated link.

The device selected offers up to 5Mbps data rates and Isolation up to 4000 KV rms.

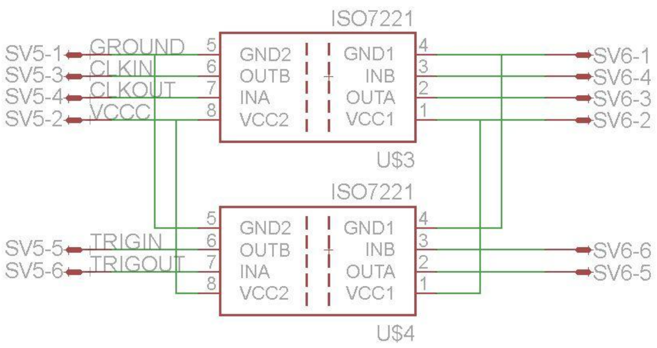


Fig. 6: Isolation Module Schematic

Microcontroller:

The micro-controller selected was primarily due to ease of use. The team has had had prior experience in using AVR micro-controllers and hence we decided to settle on the same. It was seen that due to voltage level issues a Vdd of 3.3V was essential for the micro-controller. This is because the ADC can operate only at 3.3V As such a low power variant of an AVR device needed to be selected.

Keeping these facts in mind an Atmega16L was selected. The Atmega16L is being externally clocked at 16Mhz. Some additional utilities such as a debugging switch interfaced on an external interrupt and debugging LEDs have been provided. The micro-controller has 2 8-bit timers out of which one is being used to generate a clock for the ADC. The other micro-controller peripheral that is being used is the inbuilt UART module.

More details regarding the micro-controller operation can be found in the section on embedded software.

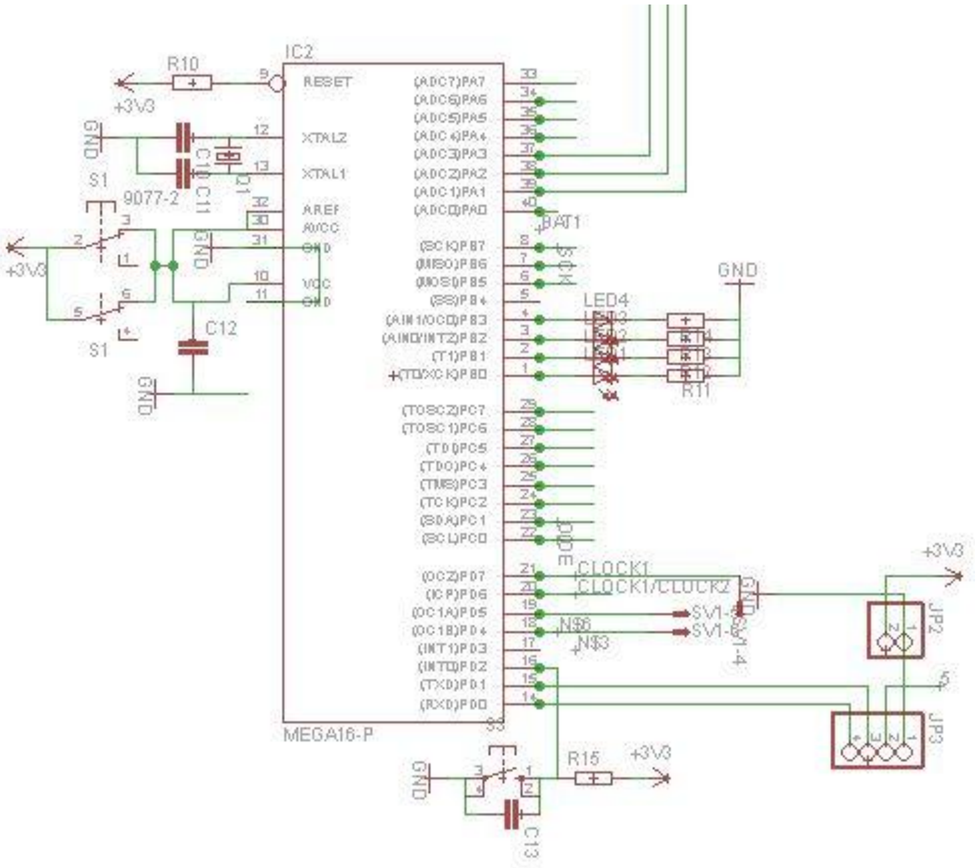


Fig. 7: Microcontroller Module Schematic

RF module:

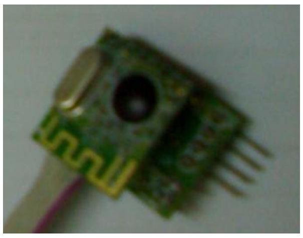


Fig. 8: RF Module

The RF modules used for implementing the wireless module to PC interface link are custom made modules based around the CC2500 chip. These modules operate at 5V V_{cc} and communicate at a 3.3V logic level. There is a micro-controller on these modules which performs basic tasks like buffering etc. The modules offer a maximum data rate of 2 Mbps and a range of around 10-15m.

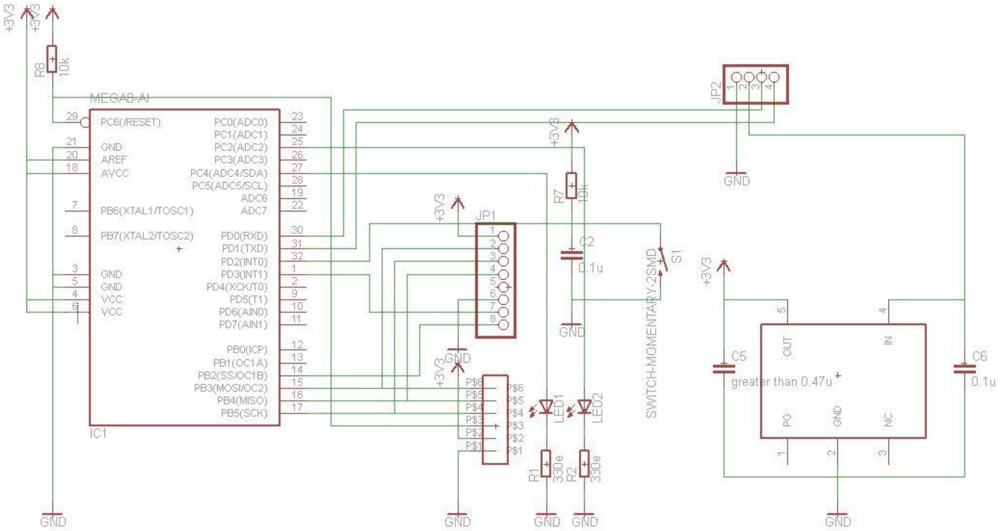


Fig. 9: RF Module Schematic

Power Supply:

The power supply requirements of the device have a moderate level of complexity. The device consists of several components that work only at 3.3V while 5V must be provided due to power supply constraints of the RF modules that have been designed. Finally the PGA designed requires a dual ended supply and this is set to +3.3V and -3.3V. Below is a list of the various ICs and their operating voltages:

Devices	Possible Range	Operating Voltage
PGA – OPA725	4V – 12V	6.6V (-3.3V to +3.3V)
ADC – THS1040	3 – 3.6V	3.3V
Isolator – ISO7221C	3 – 5.5V	3.3V
uC – Atmega16L	2.7 – 5.5V	3.3V
RF Module	4.0 – 7V	5V

The above set of requirements confirms the need for three operating voltages of 5V, 3.3V and -3.3V. To produce these voltages various alternative including buck convertors were analysed. Final decision was taken based on the simplest design alternative available.

The power supply is derived from 2 separate battery packs. One of 2 Li-ion cells and one of one single 1 single Li-ion cell. The 2-cell pack has a nominal voltage of 8.4V and is connected to a linear regulator REG103-5. The 5V output of this regulator is then connected to a 3.3V LDO regulator TPS73633. To derive the negative voltage an innovative scheme is used. The single cell Li-ion battery is connected to a TPS73633. Thus a +3.3V is available between the output of the LDO and its ground. Now the output of the LDO is connected to the circuit ground to derive a -3.3V supply at the LDO ground terminal. The table below provides a brief overview of the regulators used on the device:

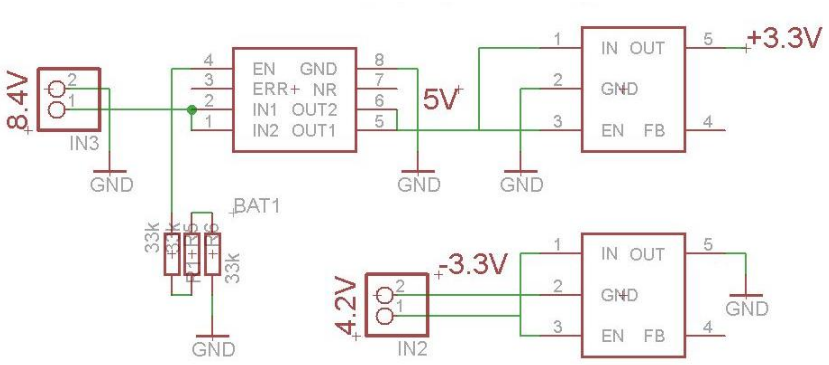


Fig. 10: Power Supply Stage

Device	Output Voltage	Output Current (Max)
REG103-5	5V	500mA
TPS73633	3.3V	400mA

In addition the power supply stage also provides a second feature of monitoring the battery voltage. This is done by connecting the battery to a resistance divider. This divided voltage is then provided to the ADC of the onboard uC operating at 3.3V.

The selection of the divider was done as follows:

$$V_{ADC} = V_{batt} \times \left(\frac{R_1}{R_1 + R_2}\right)$$

For the chosen resistance values this reduces to:

$$V_{ADC} = \frac{V_{batt}}{3}$$

Since V_{batt} varies between 7V and 8.4V, the V_{adc} will vary between 2.33V and 2.8V. This can be measured by the ADC.

Software Description

Embedded Software:

The embedded software is essentially responds to the commands sent to it via the base station. As such the entire embedded software is heavily interrupt driven. The main loop periodically checks to see whether a data-packet is requested and if so sends the packet. In addition it monitors the battery voltage and checks for user input via the touch switch which has already been debounced in hardware. The main DSO application is built around the two interrupts:

1. Timer 0 Overflow interrupt
2. UART Receive Complete interrupt

The timer 0 overflow interrupt implements the data logging and triggering component of the embedded software.

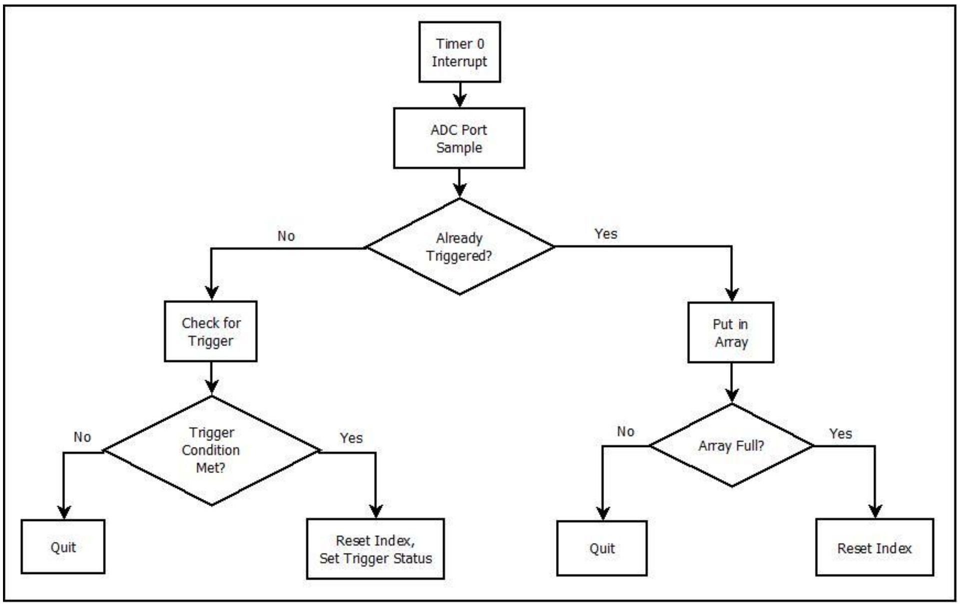


Fig. 11: Timer 0 ISR Flowchart

The ISR proceeds as follows:

1. ADC port is sampled to check for new data
2. If the module is in the triggered state, the data is logged till the array is full
3. If the module is not in the triggered state, it checks whether the triggering condition is satisfied by the current data sample.
4. If the data buffer is full the index is reset and the triggered state is disabled.
5. If a new trigger condition succeeds than the array starts filling up from the beginning overwriting the original values similar to a normal oscilloscope.

The triggering conditions used above are as follows:

1. The triggering module uses a level triggering on a positive slope.
2. The slave module triggers as soon as the master module triggers

The state machine and other setting of the module are set in the UART receive interrupt in direct response to commands sent from the PC.

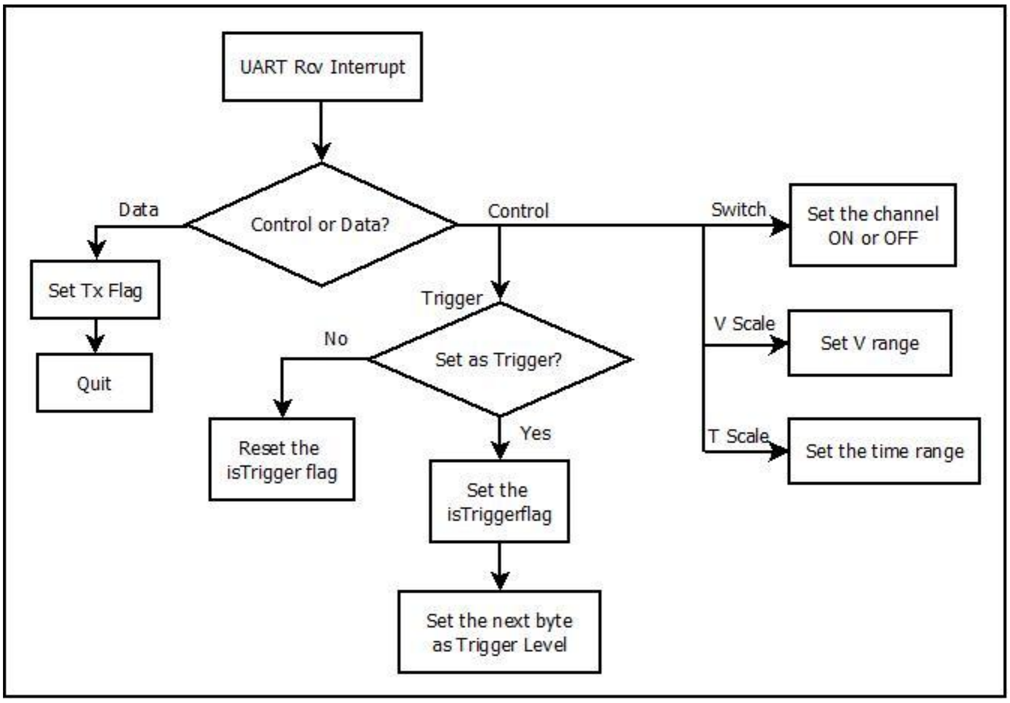


Fig. 12: UART ISR Flowchart

The computer interface communicates to the module using a specific one byte long format. At the beginning of a communication the device checks whether it is the device being addressed. There are essentially two types of request made by the PC interface:

1. Data Request
2. Control Command Request

In response to a data request the module sets a transmit flag which enables transmitting of data via the UART transmit interrupt.

The response to the control command request depends on the exact control command.

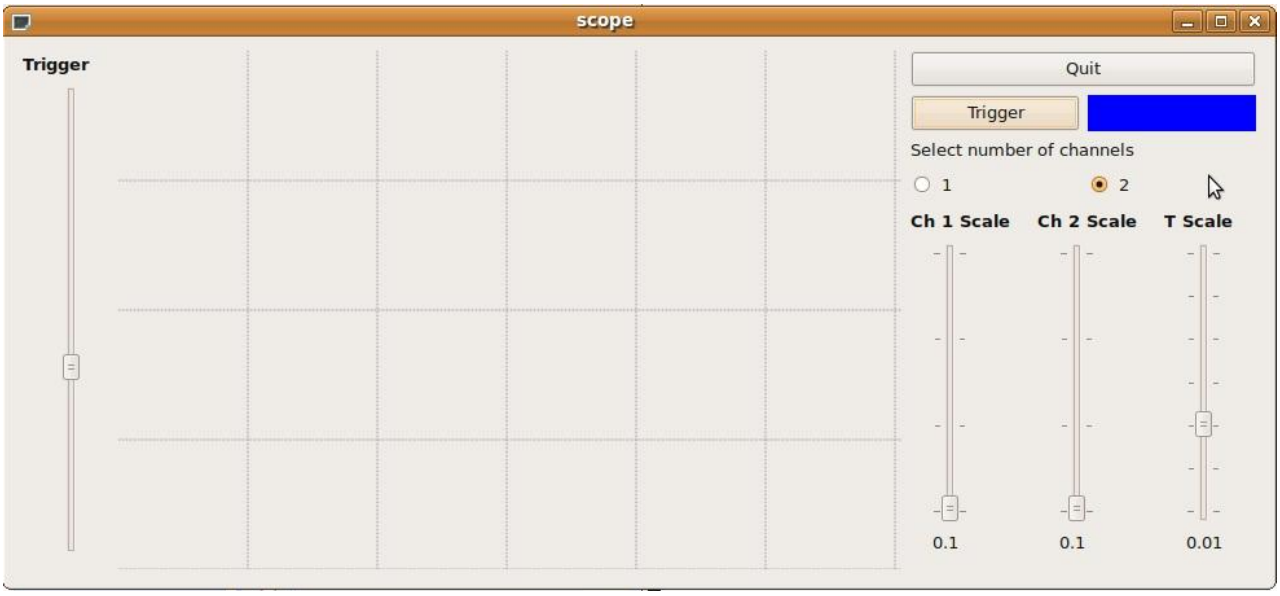
1. Trigger Set/Unset: The response to the unset command results in the module going into the slave mode. In response to the set command the module goes into the master mode and awaits the exact trigger level as the next byte
2. Channel Disable: The response to this mode is that the device does not send any data to the PC interface until it is re-enabled. In future this mode may also be used to commit the device to a power down state.
3. Time Scale Select: The response to this command is that the module waits for the next byte so as to set the appropriate time scale for the device.
4. Voltage Scale Select: The response to this command is that the device sets the PGA gain according to the voltage scale selected by the user

The GUI

The GUI is developed using the QT4 framework.

The GUI performs following two main functions:

1. Displaying the incoming data in a graphical format
2. Handling all control operations by sending appropriate instructions to the wireless modules.



Select number of channels

○ 1 ● 2

Ch 1 Scale 0.1

Ch 2 Scale 0.1

T Scale 0.01

Fig. 13: GUI Screenshot

The important features of the GUI are as follows:

1. **plotArea:** The plot area located in the centre is the widget on which the signal waveform is plotted. It has a permanent grid in the background.
   When the application is started, the PaintEvent related to the Grid is called. PlotArea is a child of the Grid widget. The PaintEvent of the plotArea contains the code section which plots the points from the data buffer according to whether it is running in single channel or dual channel mode. This PaintEvent is called whenever the timer interrupt is generated.
2. **Quit:** The "Quit" button calls a function cleanup(); which closes all the related processes and closes the application.
3. **Trigger Channel Select:** The "Trigger" push button allows the user to switch between the channel on which triggering is being done. The line colour of the selected channel is displayed in the box next to this button as an indicator. When no channel is selected or when mode of operation is changed, it turns gray. The default colour for channel 1 is red and channel 2 is blue.

When this push button is pressed it calls a function triggerChannelChanged() which switched between the trigger channels. If channel 1 is selected, this function sends the corresponding module control instructions to start triggering and the current triggerLevel value for reference. It also disables the triggering on channel 2 by sending an appropriate control instruction.

4. **Number of Channels Select:** Two radio buttons have been provided under the "Quit" button to toggle between the single and dual channel modes. Each of them calls the function numChannelChanged(int id) where "id" is the button ID for this purpose. This function activates or deactivates one of the channels depending on the mode selected, by sending a switch control signal to the corresponding module. In switching from dual channel to single channel mode, the triggering channel is preserved by default.
5. **Voltage Scale Sliders:** The 'Ch 1 Scale' and 'Ch 2 scale' sliders can be used to change the scale. In single channel mode, the scale corresponding to the deactivated channel is disabled. The default value for both is set to the minimum.(the 'safest' value). Whenever the slider value is changed, it calls a function vscaleChanged1(int value) (or vscaleChanged2(int value) corresponding to that slider). This function transmits the new value of the vscale to the corresponding module. The current value of the vscale is displayed below each slider.
6. **Time Scale Slider:** This slider can be used to change the time scale on the display. Whenever the value of this slider is changed, it calls tscaleChanged(int value) which instructs both the modules about the same. The current value of tscale is displayed below the tscale slider.
7. **Trigger Level Slider:** Internally, the trigger level slider can take values from 0 to 255. These correspond to the min to max value of voltage allowed. Thus, depending on the vscale the voltage corresponding to the trigger level is determined. On the user level, the trigger slider can be used to stabilise the waveform. It calls the function triggerLevelChanged(int value) which transmits the new value of the trigger level to the current triggering channel.

Approach towards the Project

1. At the beginning of the project, our main focus was setting our targets. We studied the commercial DSOs and their features and decided which of these were feasible to implement and what specifications are we targeting in terms of bandwidth and voltage ratings.
2. Once the project goals were finalised, the work was divided in 2 major parts, developing the GUI and making the hardware. One of us handled the GUI part while the other two focussed on hardware design.
3. A week or so was spent in making rough schematics. Once the schematics were verified, individual modules were fabricated and tested.
4. Most of the experiments with these individual modules were successful, except for the experiments with a PGA280. This module was later discarded, to be replaced by the PGA designed using an OP AMP.

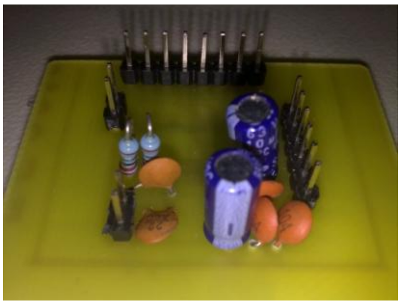


Fig. 14: ADC Testing Module

5. The GUI was built in 2 stages. The first stage was a very simple version which plotted the received data. Additional functionalities were gradually added to the GUI.
6. For testing the GUI a separate hardware module, which consisted of an Atmega16 Application board. Hard coded data samples were used to test the GUI.

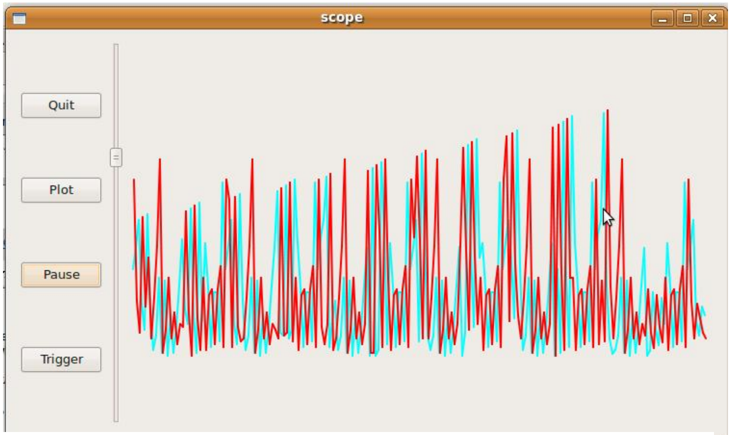


Fig.15: GUI Early Version

7. While the hardware modules were finalised and the final board was being fabricated, the codes for embedded software were written.

Future Work and Applications

Future Work:

Following are some of the ways in which the Wireless DSO modules can be made more functional:

1. A commercial DSO has many more functionalities such as autoset, cursor for accurate measurement, spectral analysis mode, automatic measurements etc. These can be implemented by further processing the signal obtained.
2. Bandwidth can be increased by using faster ADCs, processors and wireless links.
3. More sophisticated interpolation techniques can be use for plotting the signal waveforms.
4. The number of channels can be increased by implementing a more complicated protocol between channels.

Applications:

The wireless DSO modules can be used in numerous industrial and general purpose applications. To name a few,

1. The wireless DSO modules can be used in those applications where multiple signals, which are away from each other, need to be monitored and compared.
2. The isolation between the two channels makes it useful for power electronic applications where independent signals, which have their references separated by a large amount, need to be measured.

Appendix I: References

1. Datasheets: ATMega16, 74151, OPA725, THS1040, ISO7221
2. Tektronix DSO User’s manual.
3. Kartik Mohta’s webpage, www.kartikmohta.com
4. QT4 reference documentation and examples.

Appendix II: Acknowledgements

We sincerely thank our project guide Prof. Dipankar for giving direction at every stage of the project and mentoring us through it. We thank Prof. J. Mukherjee, Prof. M. Baghini and Prof. P. C. Pandey for guiding us at different stages of the project. We thank the WEL Lab staff and course TAs for helping us with everything.

Last but not the least, we would like to thank Mr. Mehul Tikekar for his immense support and *fundae*.