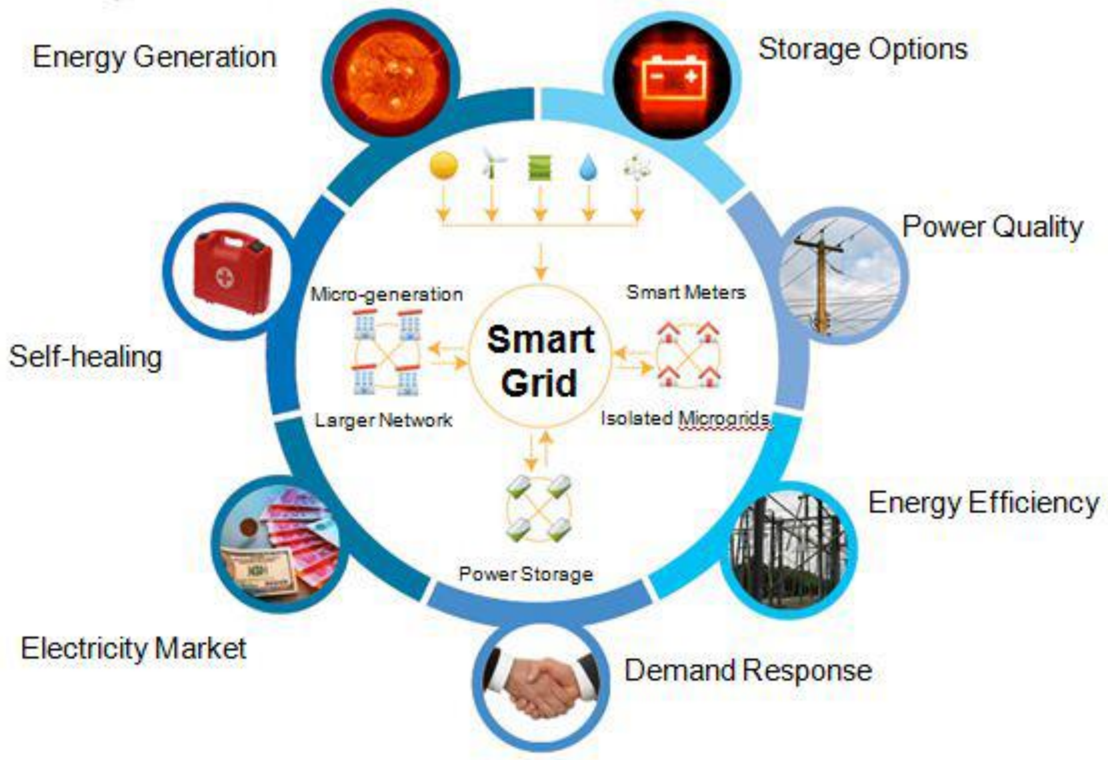
Wireless Monitoring of Electrical parameters for NUS Smart Grid

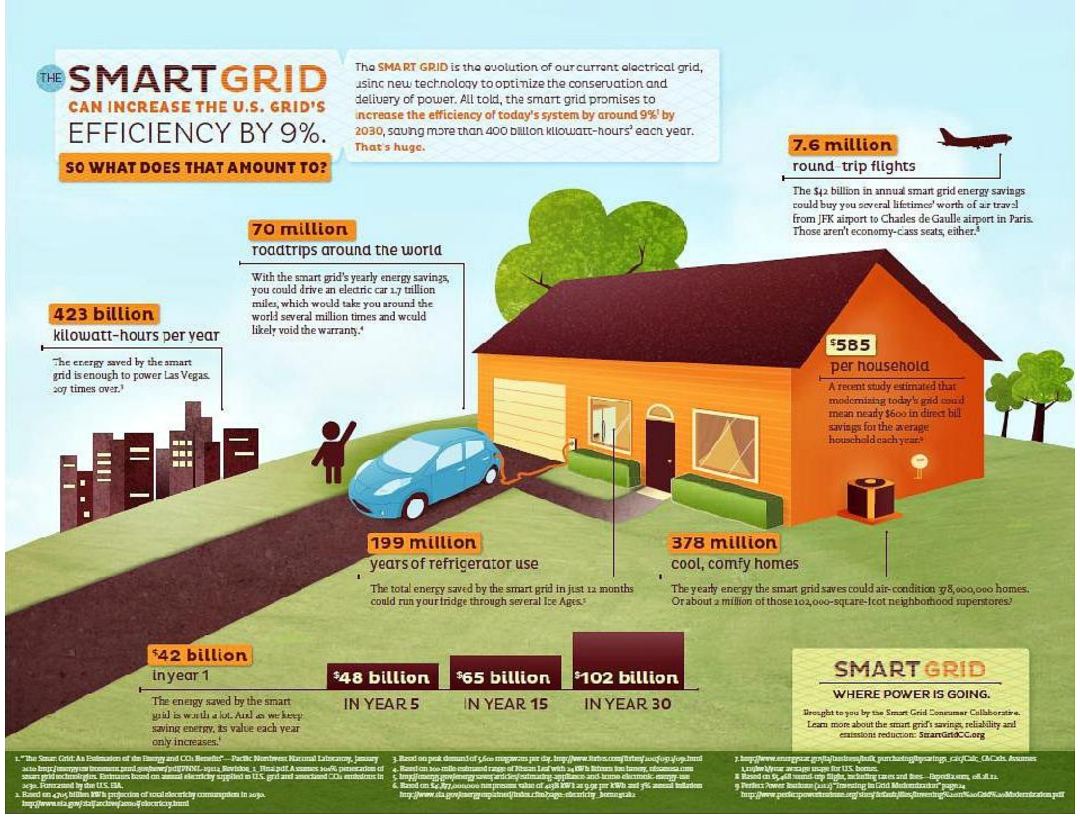
Summer Internship at NUS, Singapore

- By Subhajit Sahu

Smart Grid?


Energy Generation
Storage Options
Power Quality
Self-healing
Micro-generation
Smart Meters
Smart Grid
Larger Network
Isolated Microgrids
Energy Efficiency
Power Storage
Electricity Market
Demand Response

Why Smart Grid?

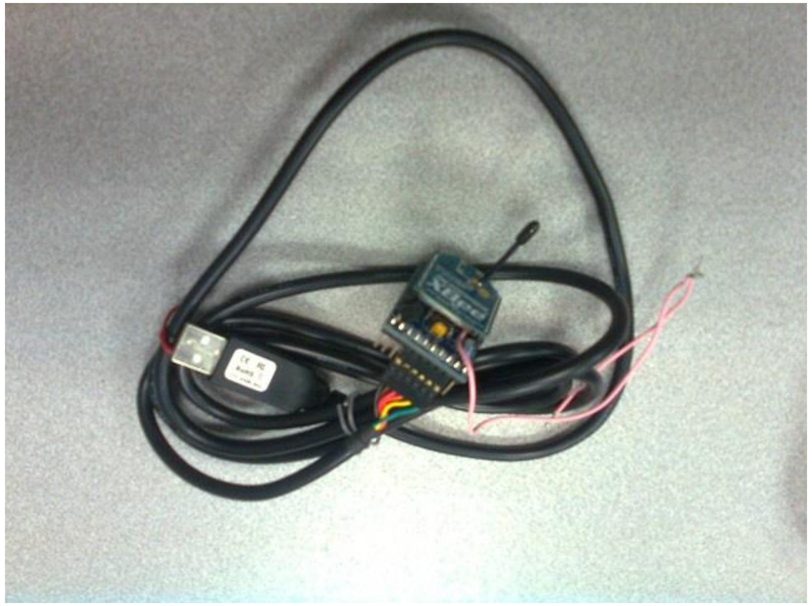


Xbee 802.15.4

- Pin details of the XBee module

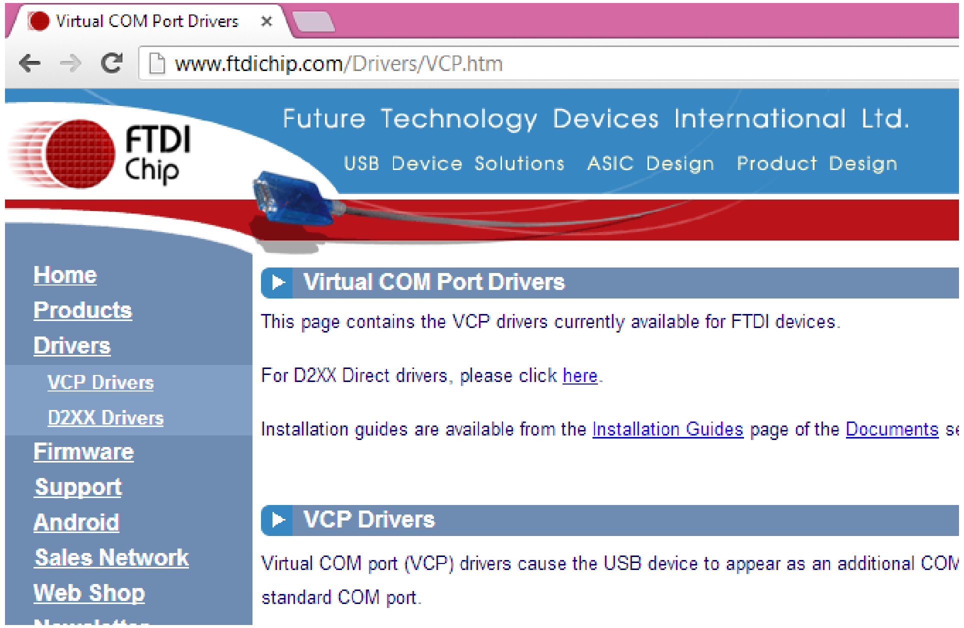
FTDI cable

- FTDI cable to connect XBee to PC



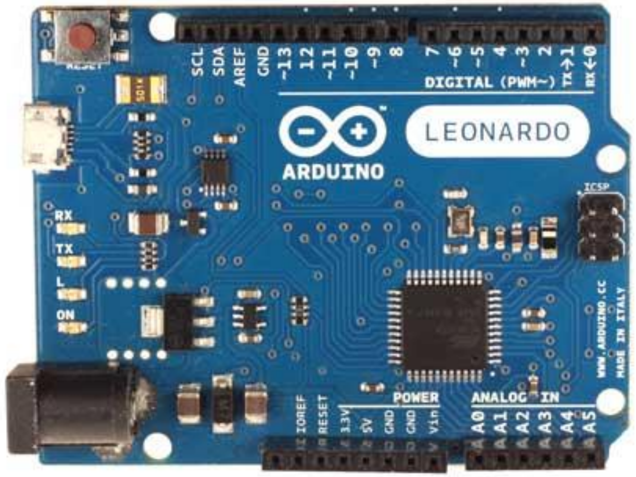
FTDI Driver

- Virtual COM port driver



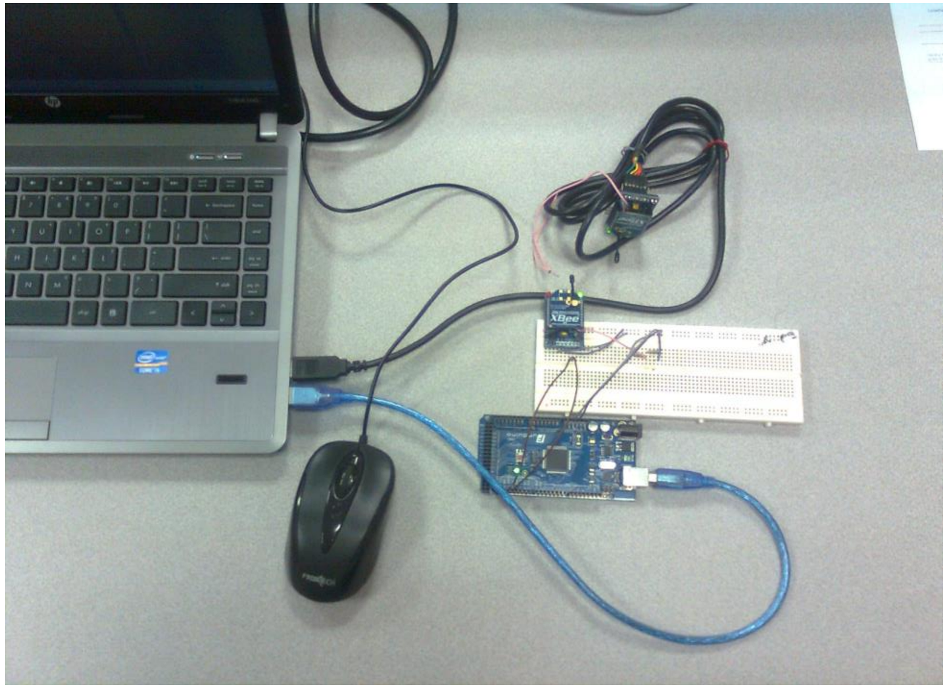
Arduino

- Arduino Leonardo board: (used this originally for testing the Xbee module)



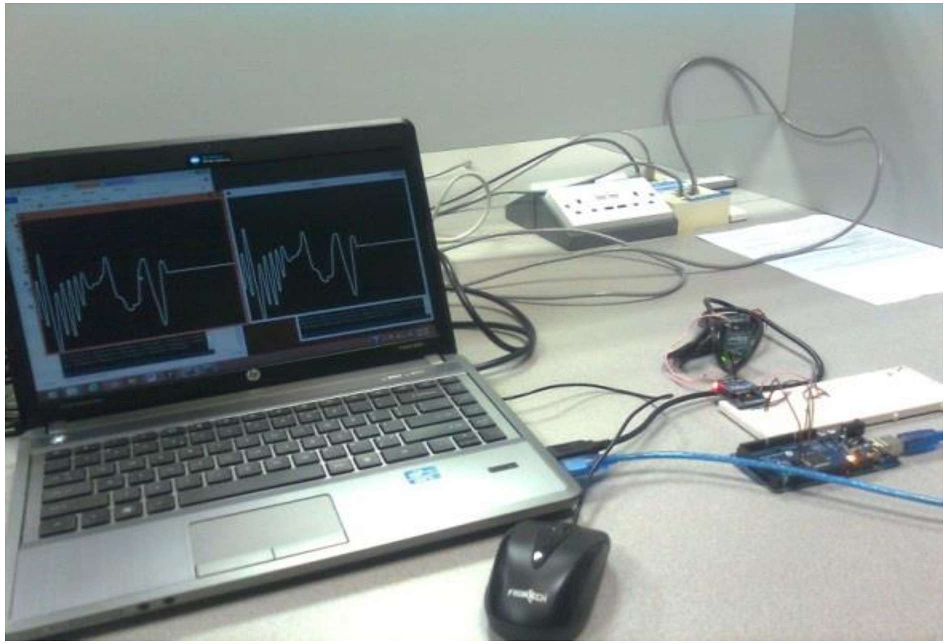
Setup for Xbee to Xbee Communication

- Experimental setup for basic XBee to XBee communication

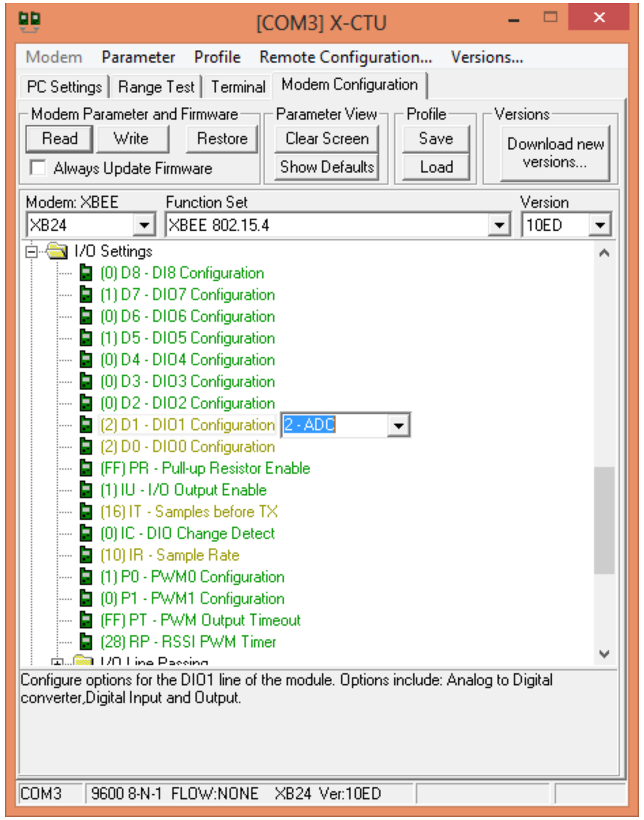


Xbee to Xbee Communication - Result

- Sent and received signals (signals were generated manually using the PC)



Configuring Xbee for Int. ADC Operation



XBee Packet Format (API)

- *Standard XBee packet example:*
- *7E 00 1C 83 56 78 22 00 05 06 00 00 00 03 FF 00 00 03 FF 00 00 03 FF 00 00 03 FF 00 00 03 FF 77*
- *Where the UART API data stream can be broken down as:*

XBee Packet Format (API)

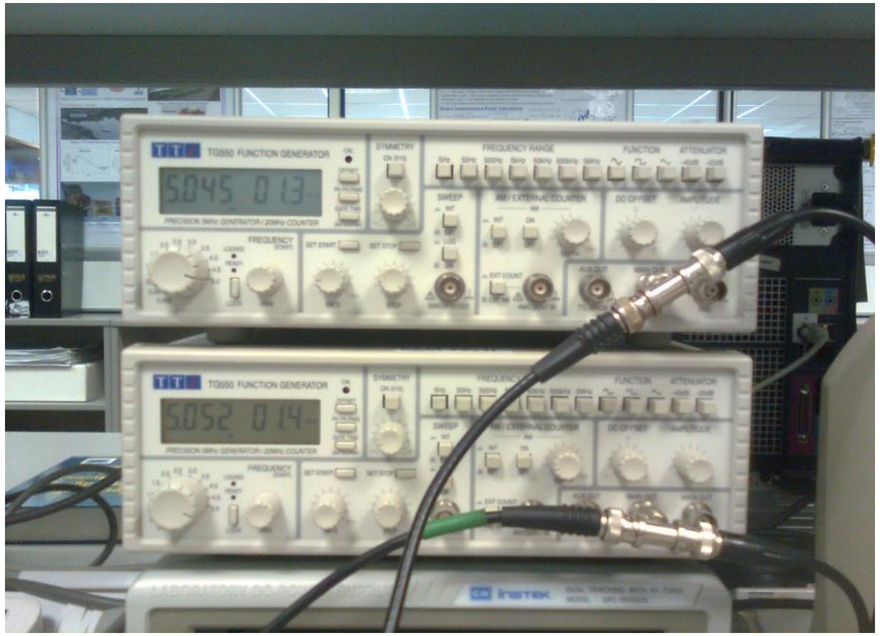
7E	**Start Delimiter**
00 1C	Length Bytes
83	API Identifier Byte for 16bit A/D data (82 is for 64bit A/D data)
56 78	Source Address Bytes
22	RSSI Value Bytes
00	Option Byte
05	Sample Quantity Byte
06 00	00000110 00000000 Channel Indicator *
00 00	Sample Data ADC0 (min value for A/D is 00 00)
03 FF	Sample Data ADC1 (max value for A/D is 03 FF, indicating the value of ADC1 is Vref)
77	Check Sum

XBee Packet Format (API)

MSB = 0x06		LSB = 0x00													
0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0
NA	A5	A4	A3	A2	A1	A0	D8	D7	D6	D5	D4	D3	D2	D1	D0

Setup for Signal Monitoring with XBee Internal ADC

- Two function generators being used to generate 2 unipolar signals to be measured

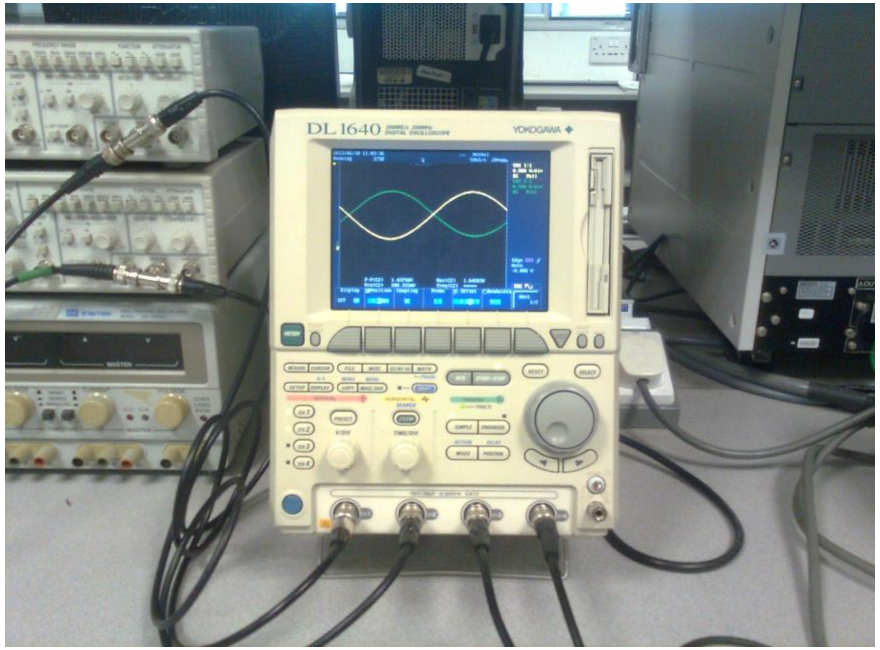


Setup for Signal Monitoring with XBee Internal ADC

- Experimental setup for capturing 2 unipolar analog signals using in-built ADC present in XBee

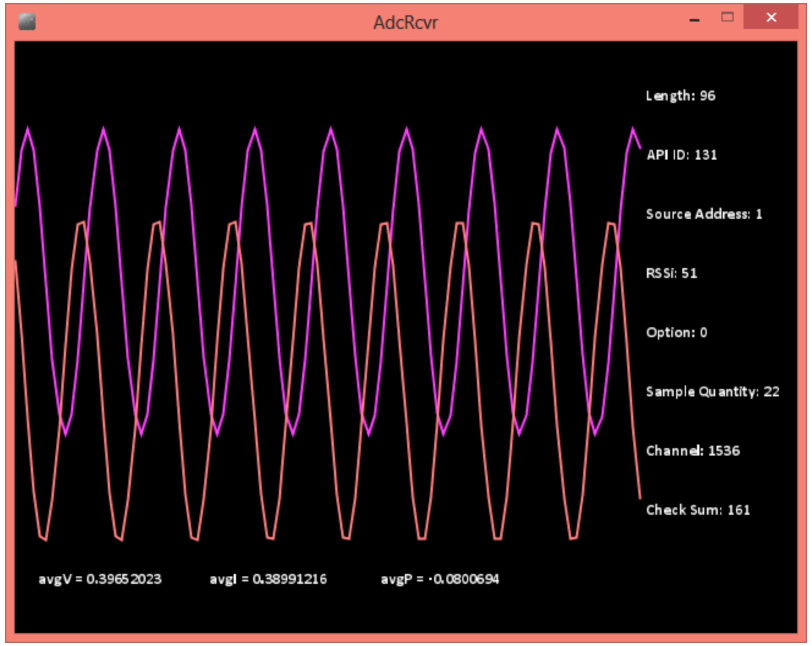
Setup for Signal Monitoring with XBee Internal ADC

- 2 unipolar analog signals being displayed on an oscilloscope

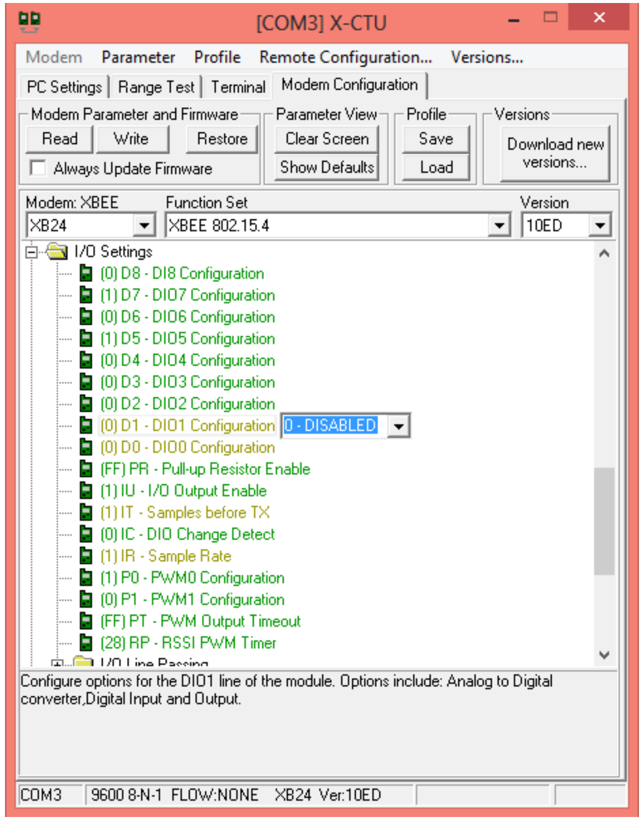


Signal Monitoring with XBee Internal ADC - Result

- Received and decoded analog signals received on PC using the developed ADC receiver program (using Processing) at around 160sps

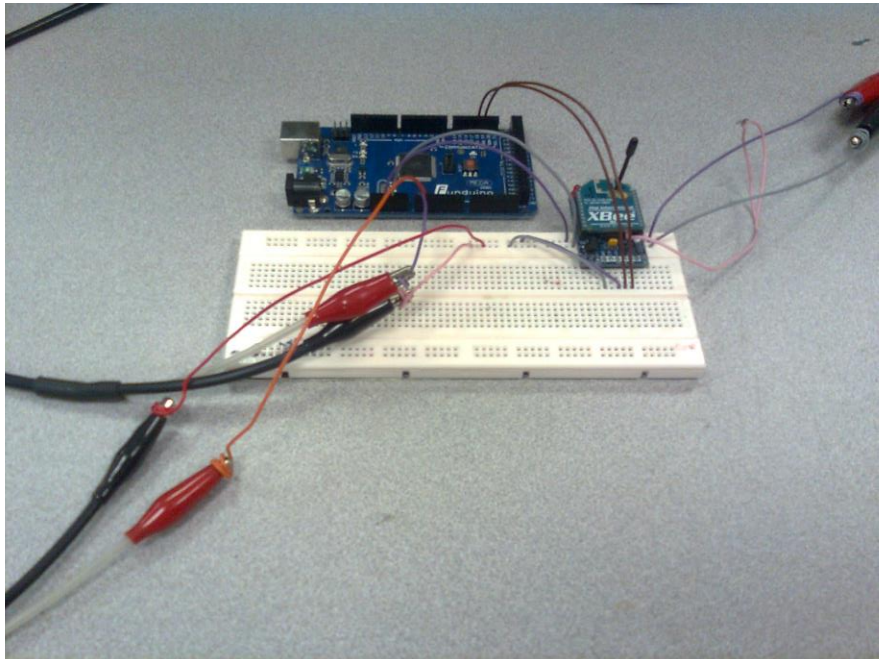


Configuring Xbee for Wireless Tx



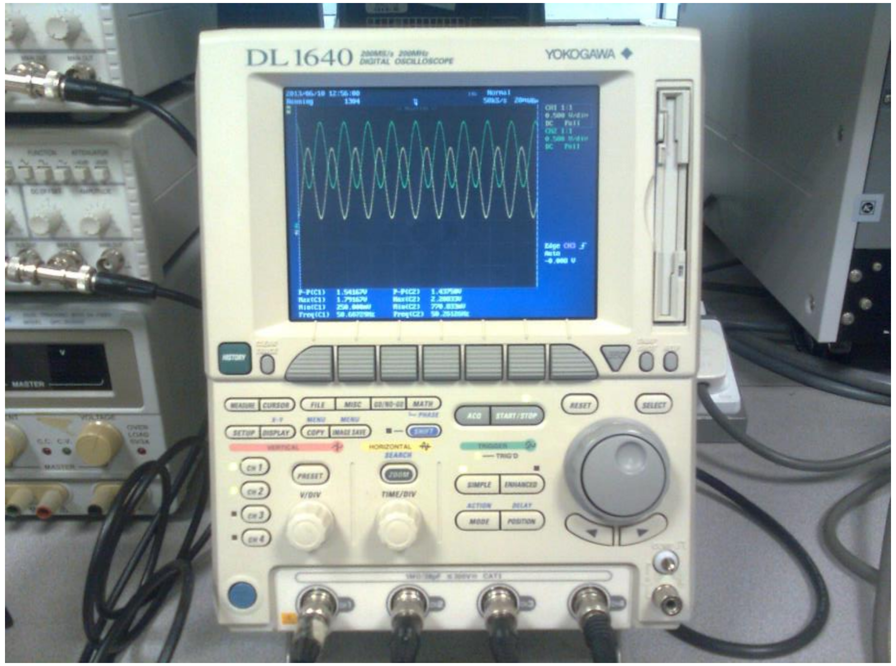
Setup for Measuring Unipolar Signals

- Experimental setup for sending unipolar analog signals to PC using Arduino



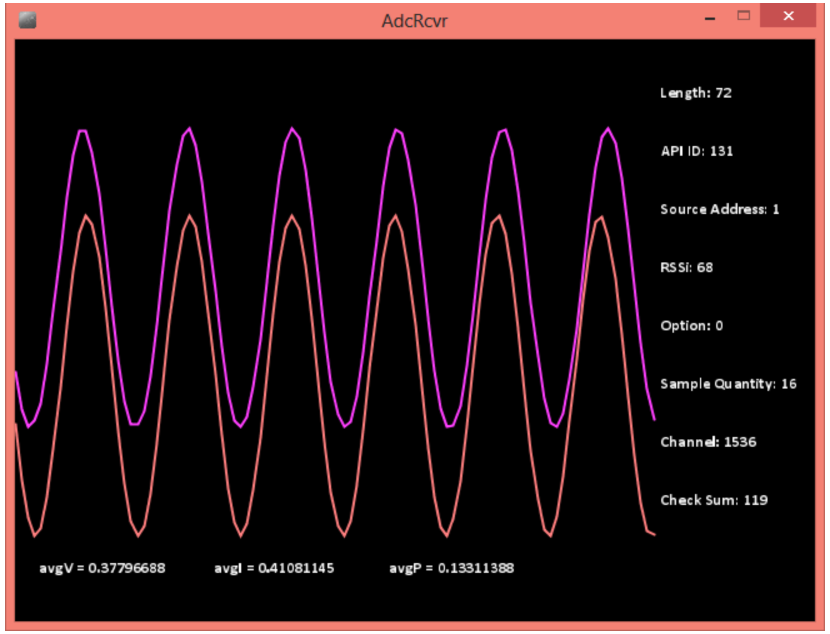
Setup for Measuring Unipolar Signals

- Original analog signals being sent by Arduino on oscilloscope



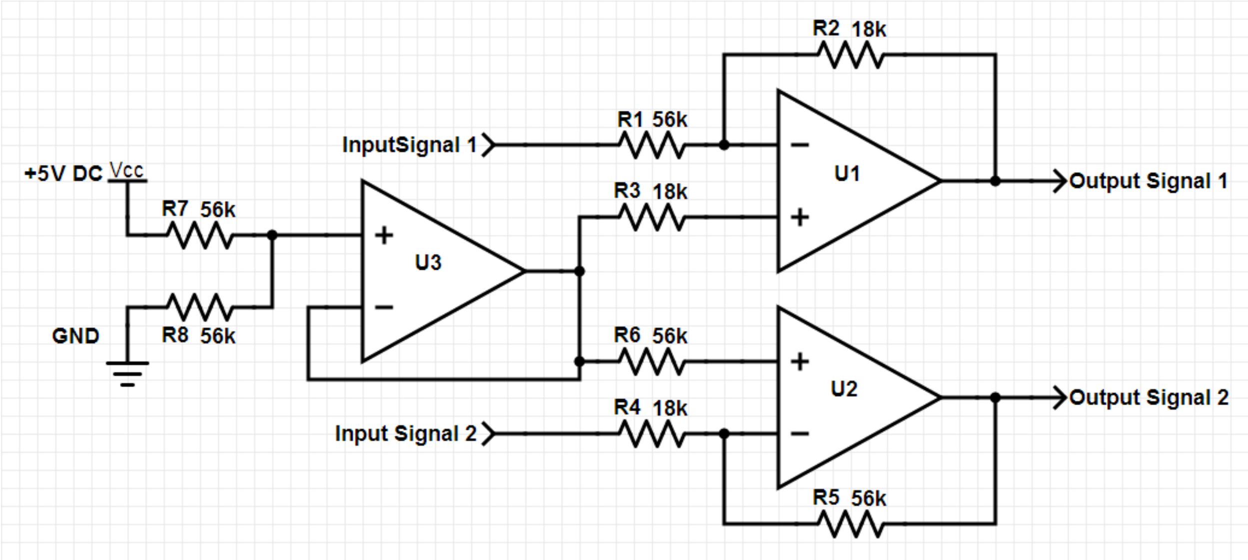
Measuring Unipolar Signals - Result

- Received and decoded analog signals on PC with a sampling rate of around 1ksps

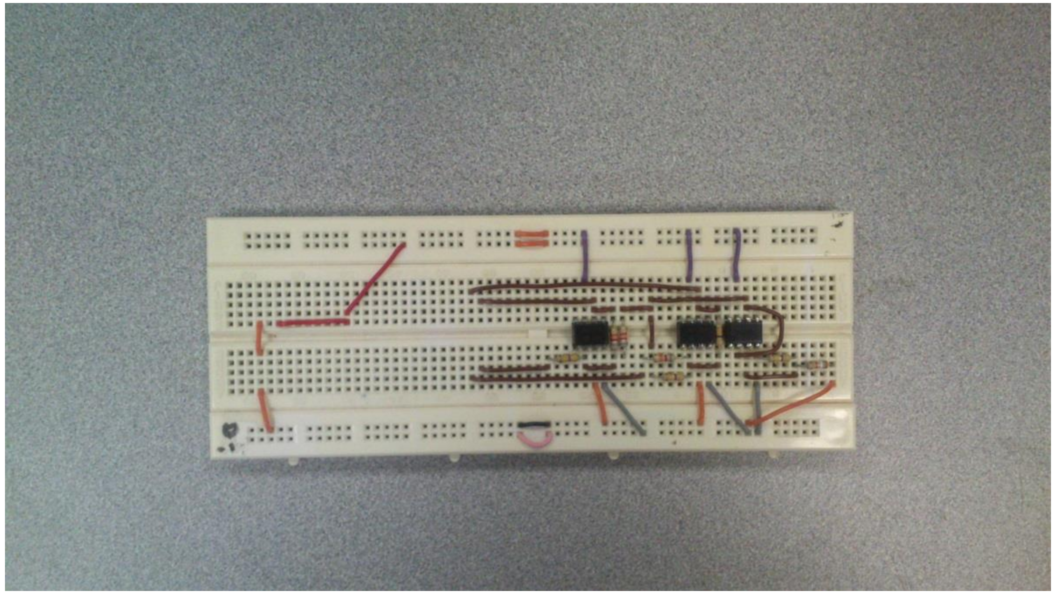


Signal Conditioning Circuit for Measuring Bipolar Signals

- Signal conditioning circuit designed to convert bipolar signals to unipolar signal centered at 2.5V and with scaling factors of 18/56 and 56/18 for signals 1 and 2 respectively

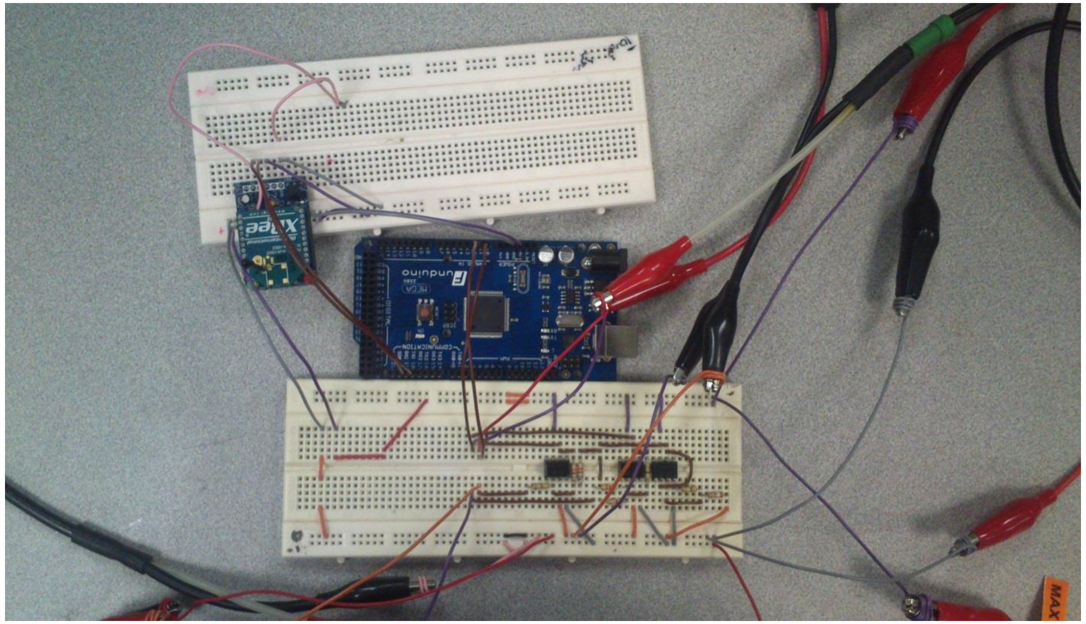


Signal Conditioning Circuit for Measuring Bipolar Signals



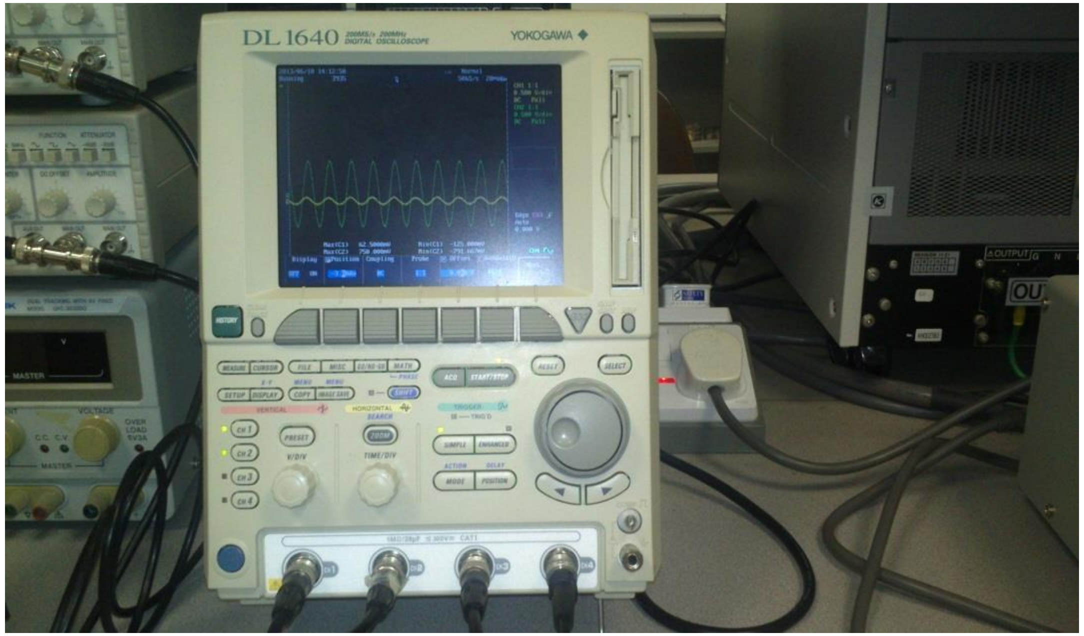
Setup for Measuring Bipolar Signals

- Experimental setup for capturing 2 bipolar analog signals using signal conditioning circuit and Arduino



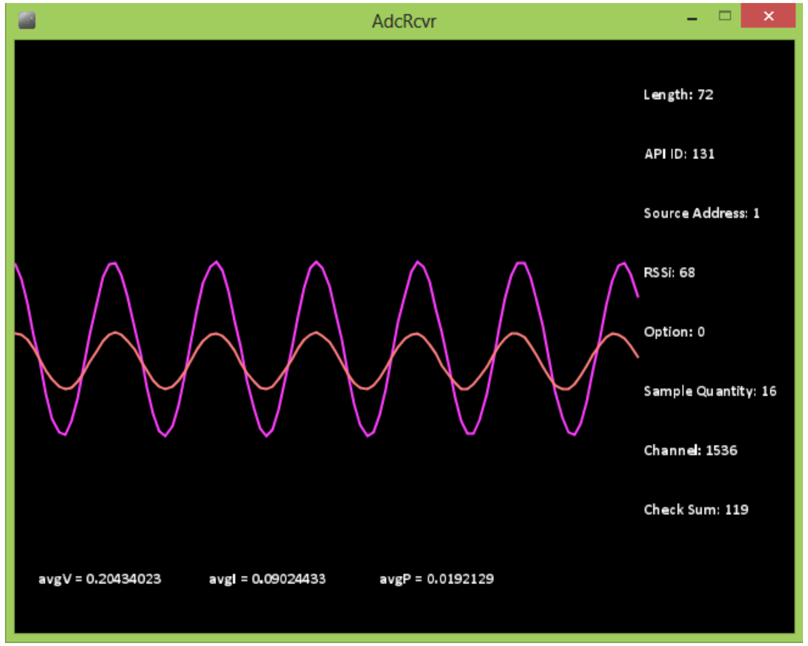
Setup for Measuring Bipolar Signals

- 2 bipolar analog signals being displayed on an oscilloscope

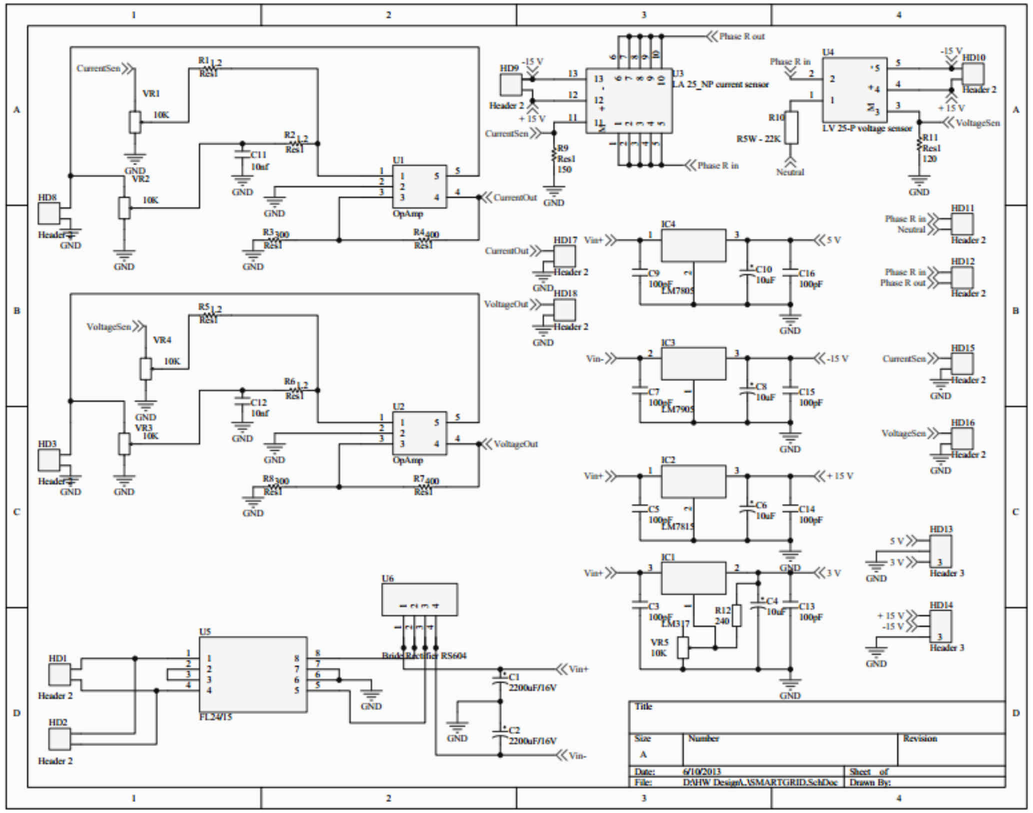


Measuring Bipolar Signals - Result

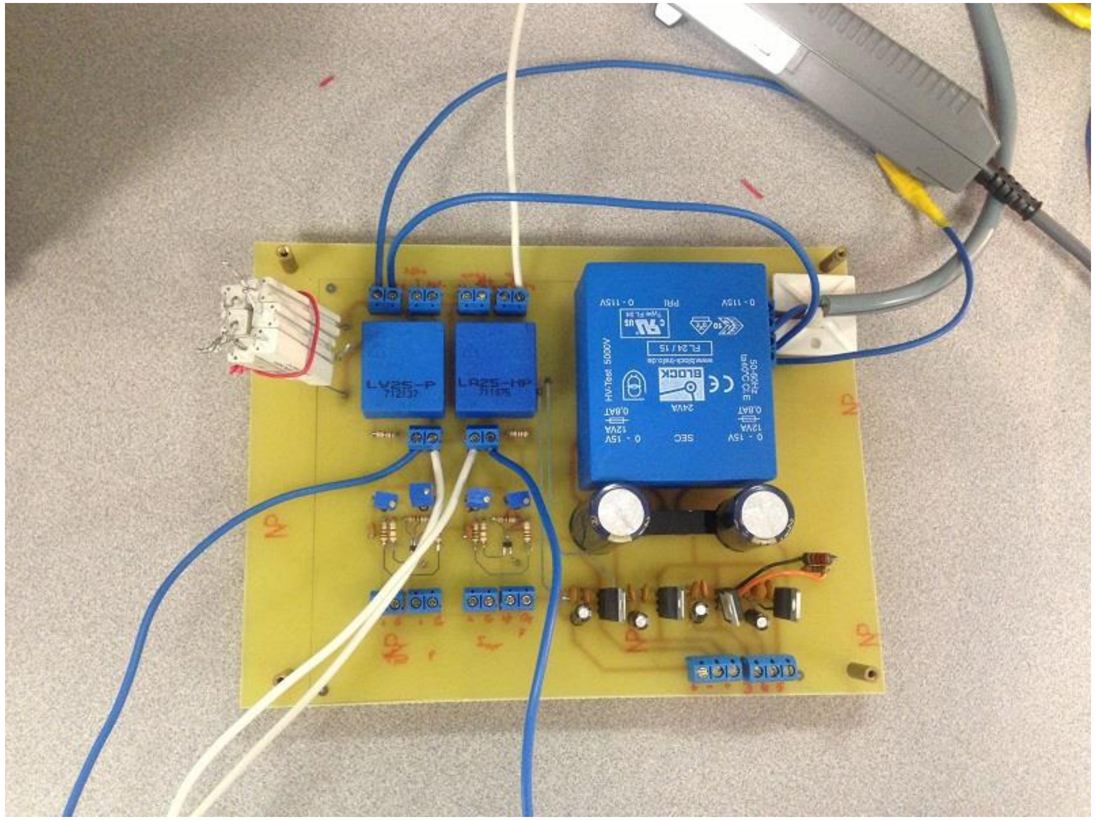
- Received and decoded analog signals received on PC using the developed ADC receiver program (using Processing) at around 1ksps



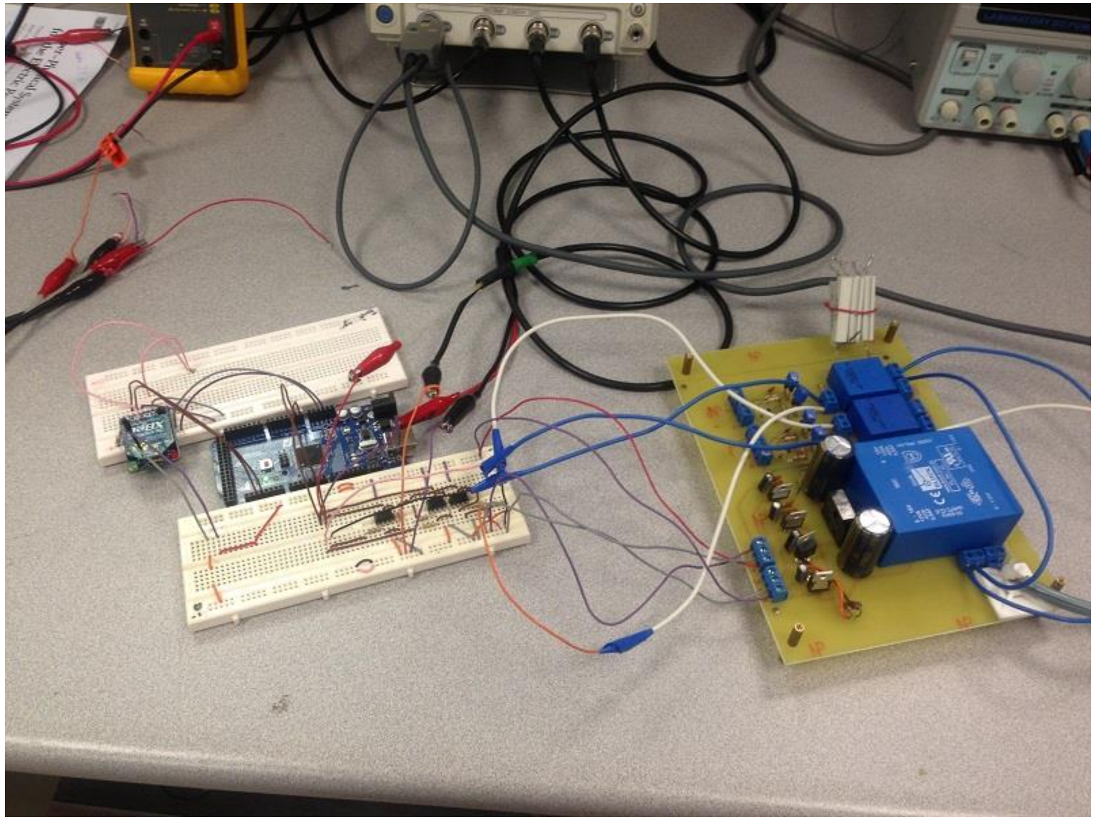
Voltage and Current Sensing Circuit



Voltage and Current Sensing Circuit

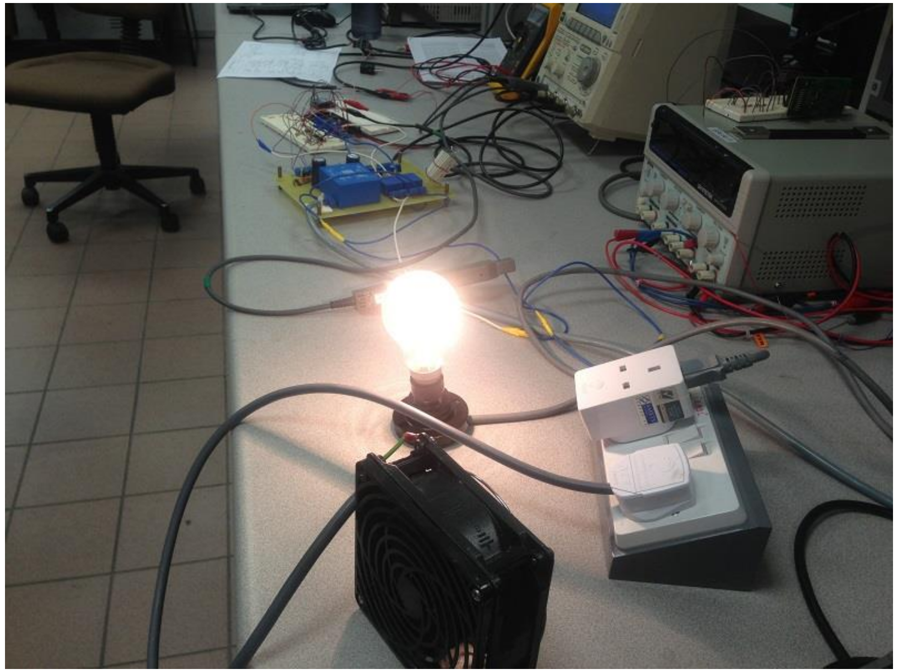


Setup for Voltage and Current Sensing



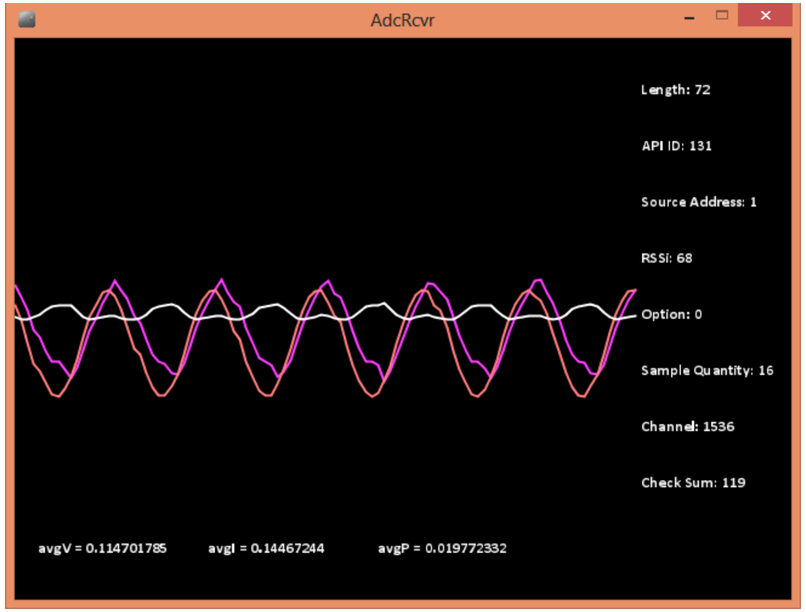
Setup for Voltage and Current Sensing

- Loads connected to the socket of the Voltage and Current sensing circuit



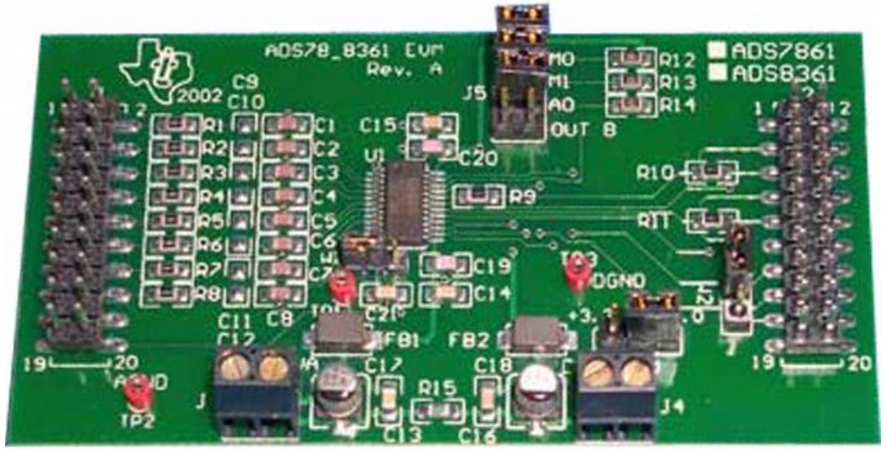
Voltage and Current Sensing Result

- Received and decoded voltage and current sensing output received on PC using the developed ADC receiver program (using Processing) at around 1ksps



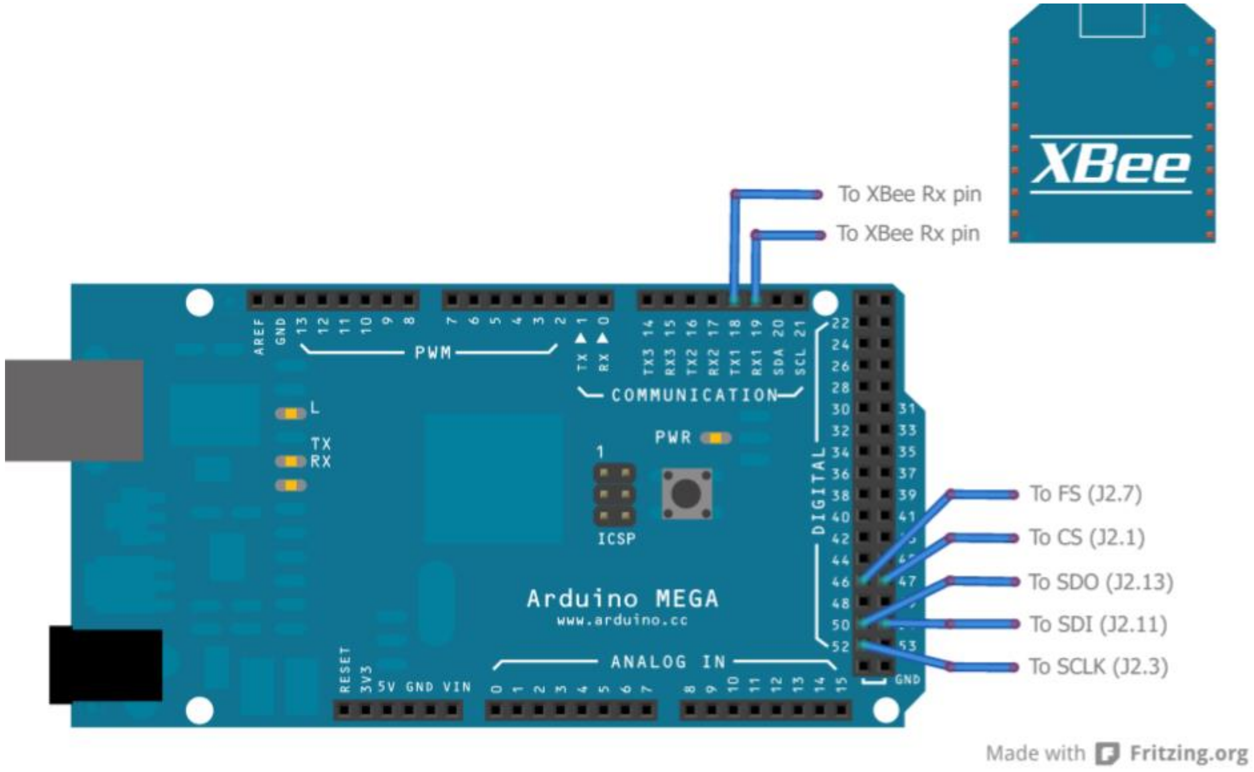
Measurement from a Simultaneous sampling External ADC

- ADS8361 EVM board is a high-speed simultaneous sampling ADC board that supports sampling rate up to 500ksps, and can be interface with SPI protocol



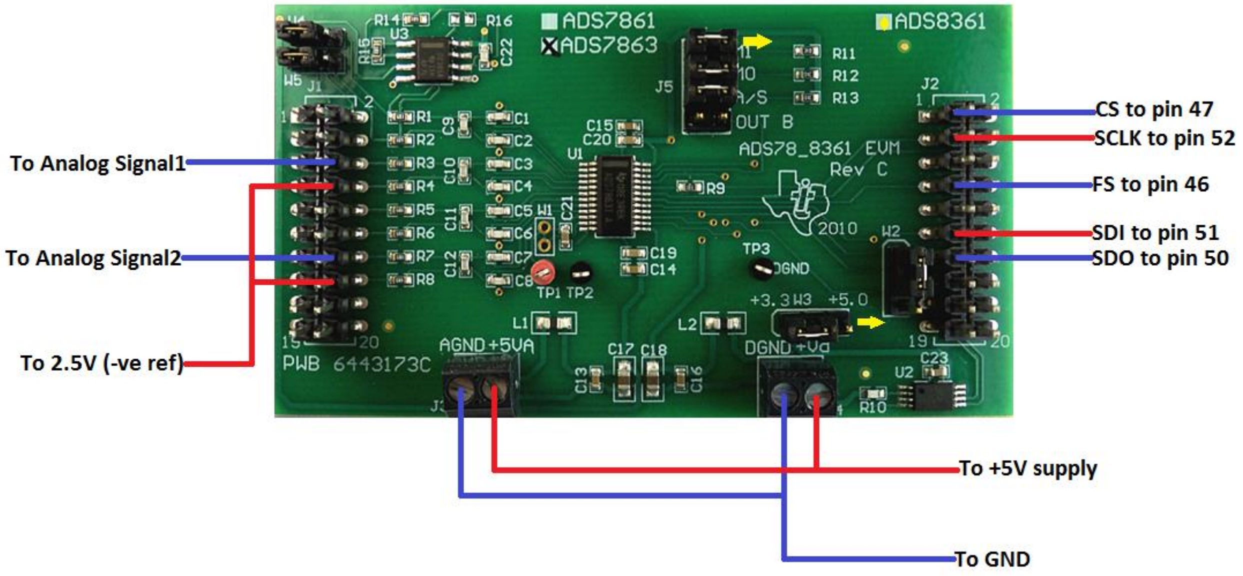
Connections from Arduino

- Connections to be made between Arduino Mega and the ADS8361 EVM board

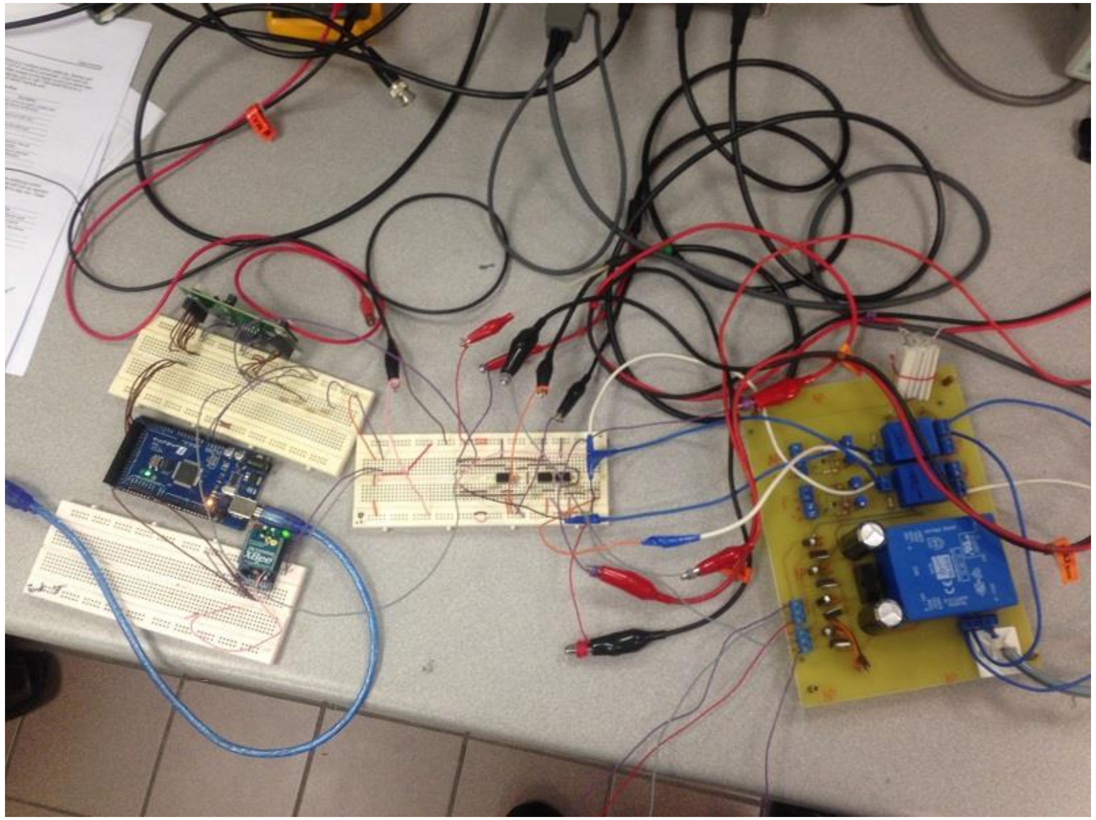


Connections from Ext. ADC

- Connections to be made to ADS8361 EVM board

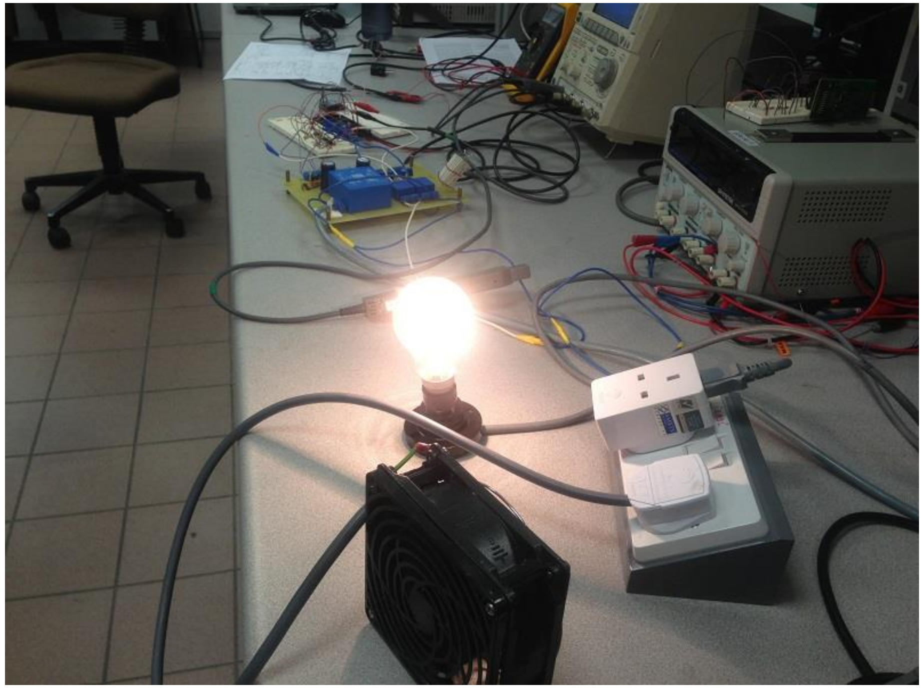


Ext. ADC Measurement Setup



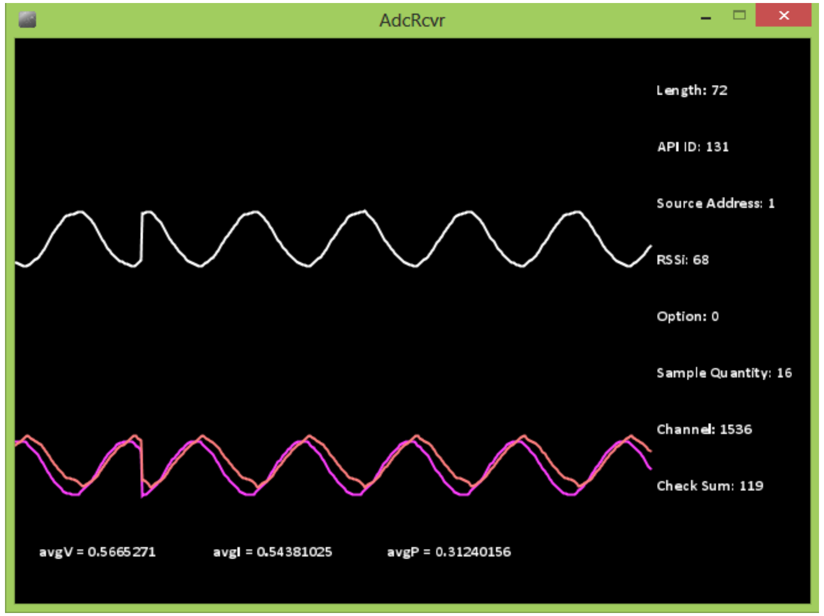
Ext. ADC Measurement Setup

- Loads connected to the socket of the Voltage and Current sensing circuit



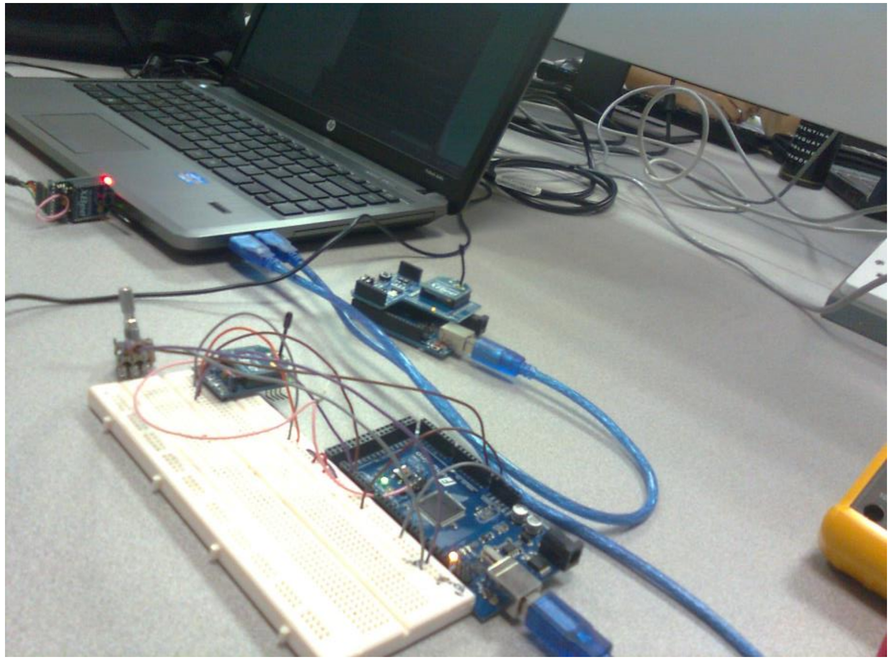
Ext. ADC Measurement Result

- Received and decoded voltage and current sensing output received on PC using the developed ADC receiver program (using Processing) at around 12.5ksps



Multi-Node Test Setup

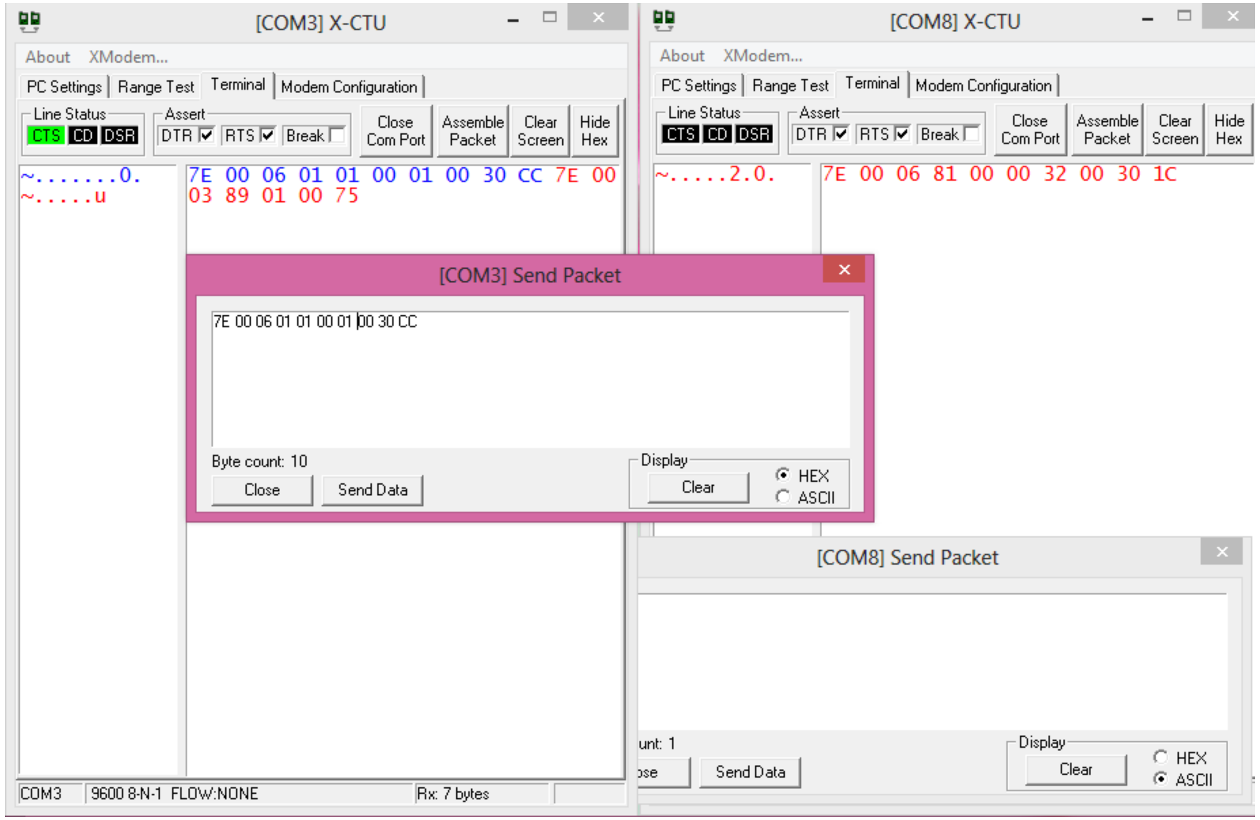
- Experimental setup for capturing analog signals from 2 different nodes and display it on the PC



Xbee API Test

- XBee API mode is being used to send data from one XBee module to another.
- This mode allows data to be transmitted in form of RF packets, as well as provides additional information such as packet reception status.
- The reception status can be seen on the COM3 X-CTU window (left) which is shown in red color.

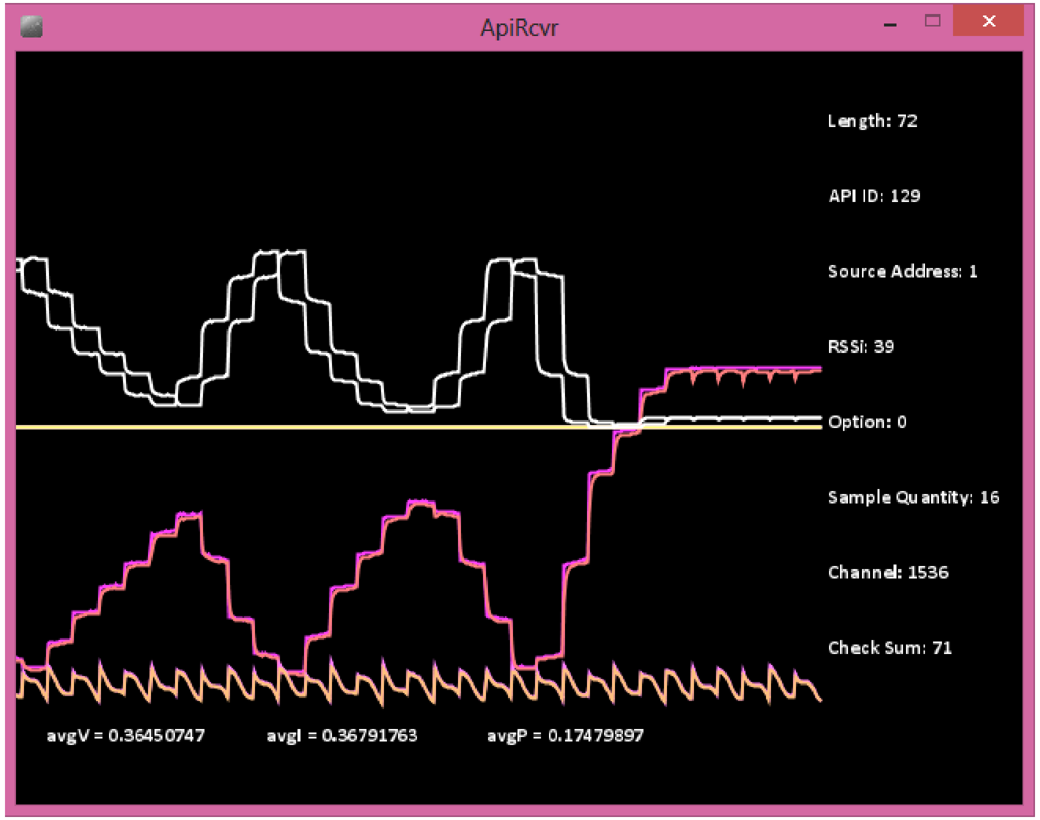
Xbee API Test



Multi-Node Data Reception - Result

- Four signals being received from 2 nodes and being displayed on the PC.
- The PC is connected to one XBee module and receives data from the nodes in alternating fashion.

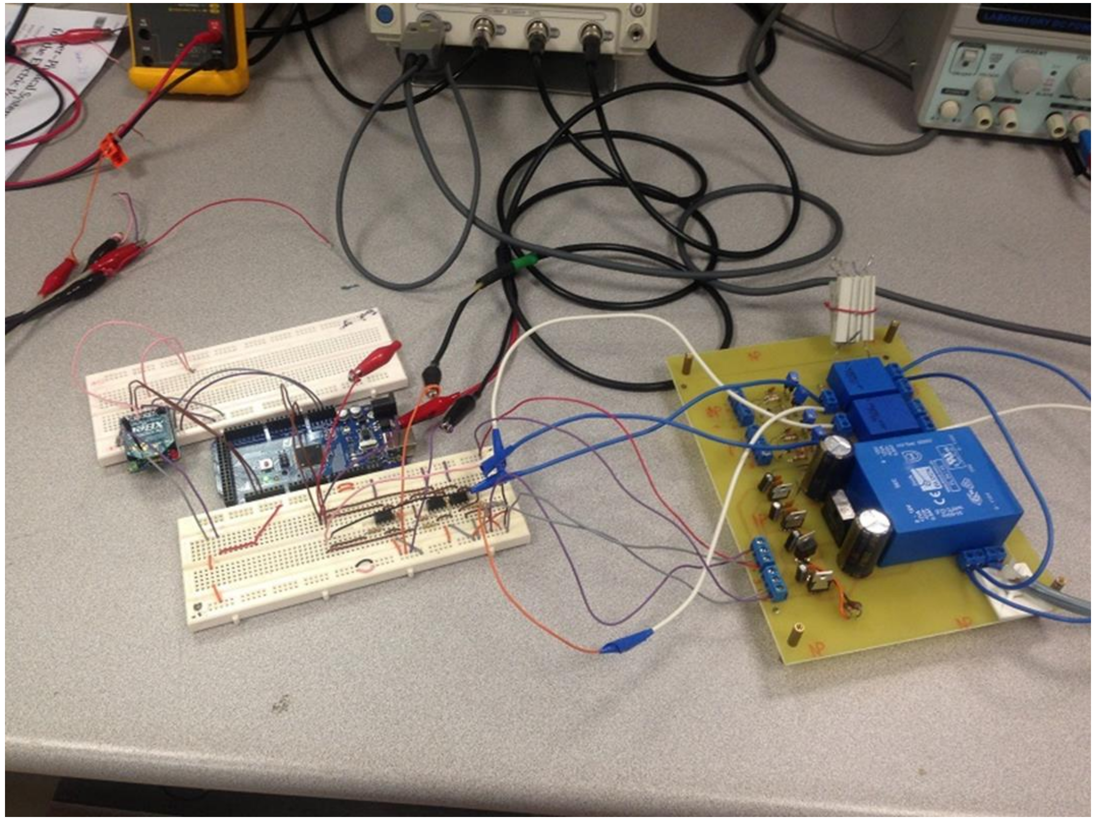
Multi-Node Data Reception - Result



Spectrum Test Setup

- Experimental setup for sensing voltage and current used on a socket using the voltage and current sensing board along with the signal conditioning circuit and Arduino.
- The captured samples were sent to through Xbee to a base station connected to Xbee which received the samples and stored it in a text file.
- The samples stored in the text file was then used in MATLAB to obtain the spectrum of voltage and current samples captured.

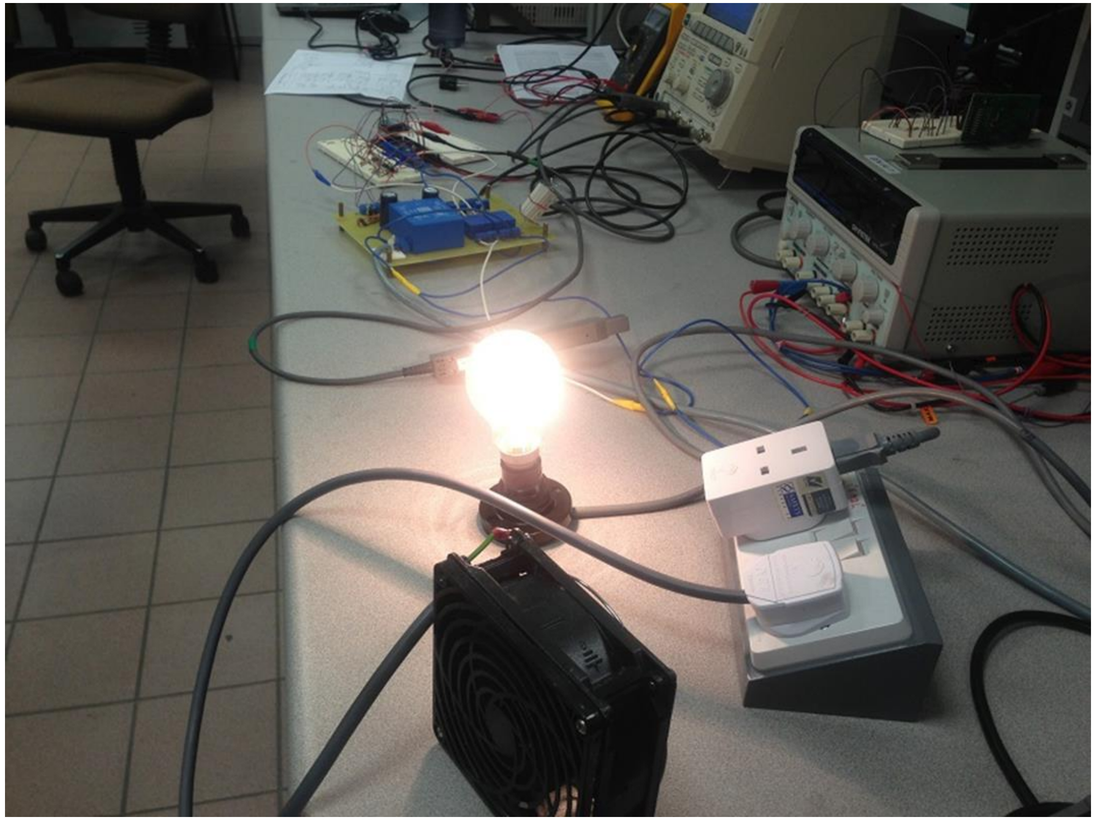
Spectrum Test Setup



Spectrum Test Setup

- Loads connected to the socket (which was connected to voltage and current transducer) were changed (turned on and off) in certain amount of time.
- A bulb and a CPU fan (shown in this picture) were used as loads, which were turned on and off).
- This was done in order to see the effect of harmonics introduced in the supply when loads were changed.
- The voltage and current transducer board which captures the signals is on the top of the picture.

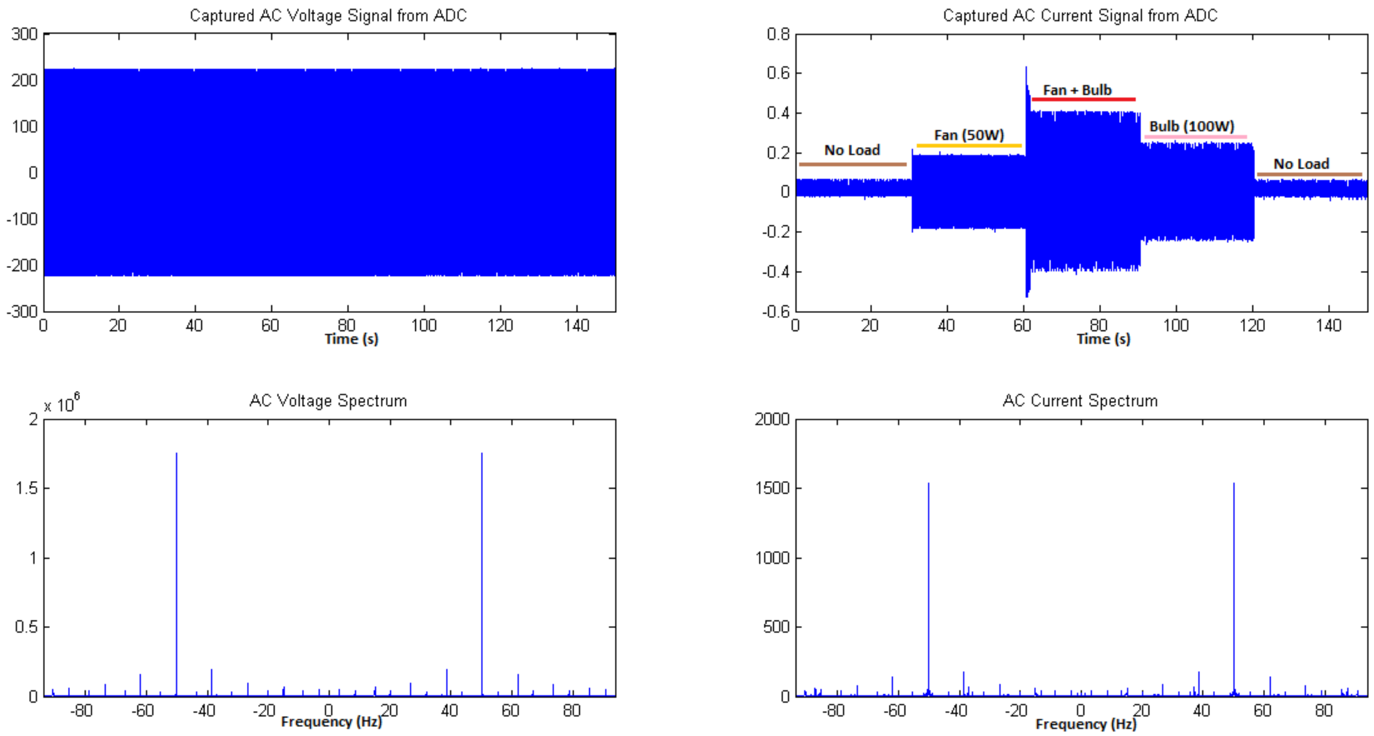
Spectrum Test Setup



MATLAB Recording

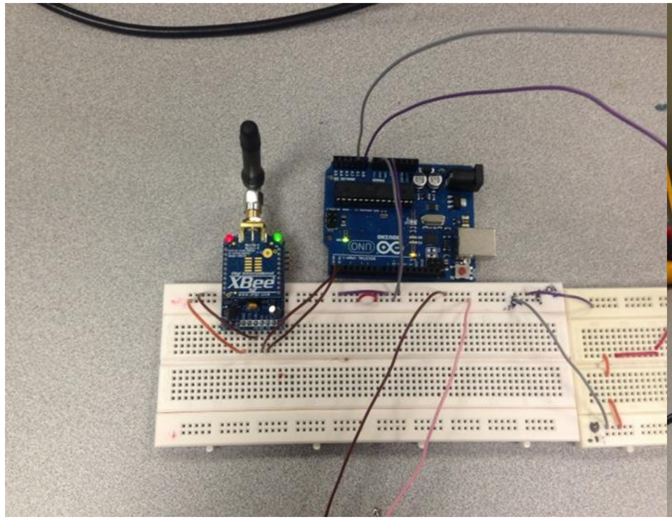
- The AC voltage and current samples that were captured by the microcontroller plotted in this figure, using MATLAB.
- DC Offset in the signals were removed and they were scaled to appropriate values.
- The spectrum of both the signals was then plotted to observe the frequencies of noise introduced in the waveforms due to load switching.

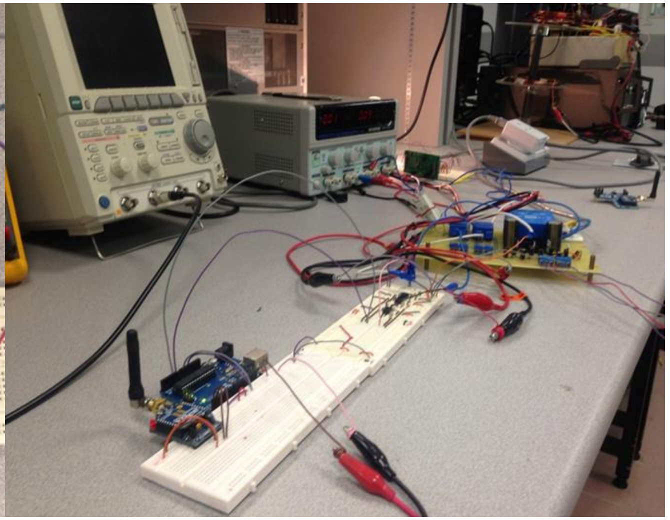
MATLAB Recording



Node 1

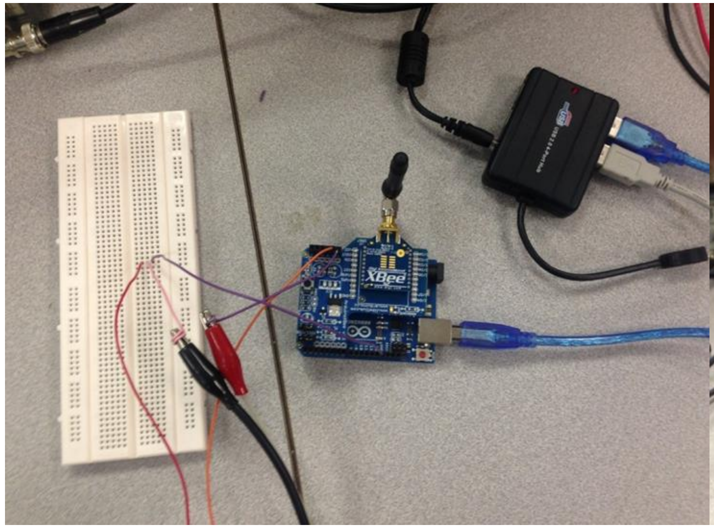
- Node 1 using Arduino Uno and Xbee S2 wireless board is capturing supply current and voltage from voltage and current transducers connected to a socket which powers a bulb and a CPU fan (not seen in picture)

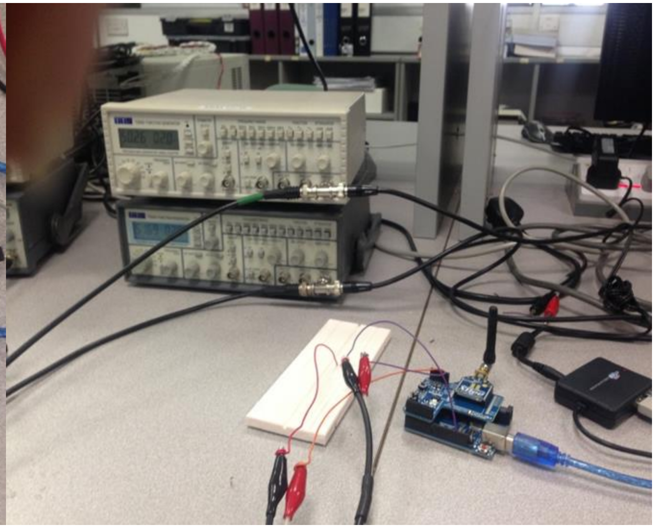




Node 2

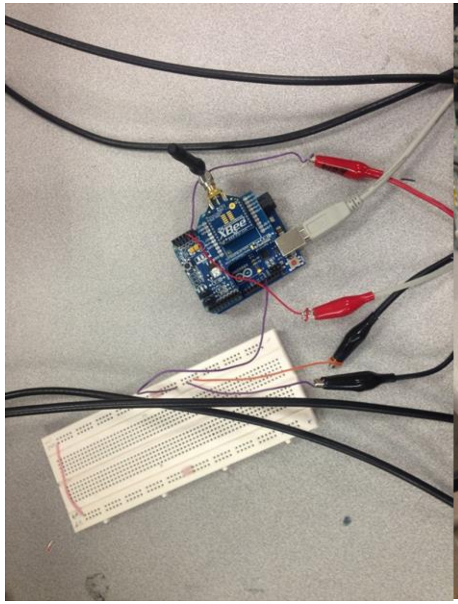
- Node 2 with the same configuration, but is capturing signals from 2 function generators instead of voltage and current transducer

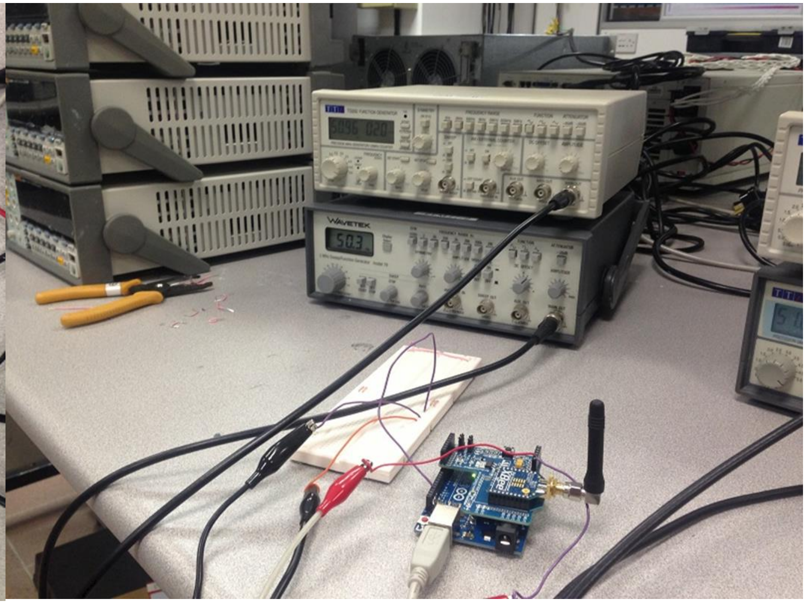




Node 3

- Node 3 with the same configuration, and is capturing signals from 2 other function generators instead of voltage and current transducer

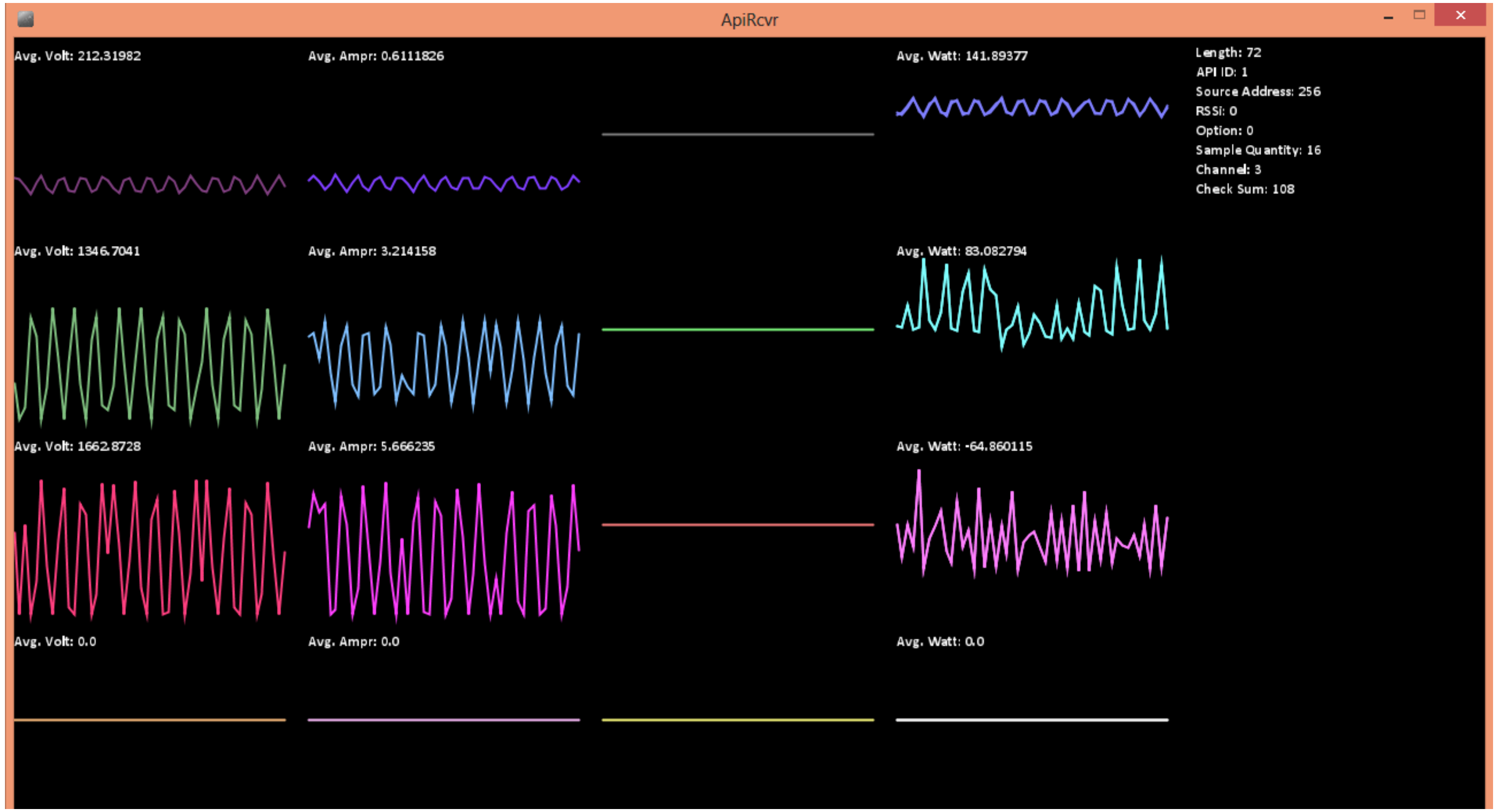




End Result

- Captured voltage and current signals captured from Node 1, 2 and 3 (at the base station connected to another Xbee S2 module) respectively from top to bottom.
- The graphs starting from the left to right denote received voltage, current and calculated power signals. In each case, the calculated average of each quantity is shown on the top of the graph.
- The voltage and current samples were received at a rate of 188 samples per second. This plot was displayed by capturing the received samples (which was sent in Xbee ADC packet format) and then displaying it using Processing. Each node sent its own identity information along with the samples so that they could be distinguished from samples from other nodes.

End Result


ApiRcvr
Avg. Volt: 212.31982
Avg. Ampr: 0.6111826
Avg. Watt: 141.89377
Length: 72
API ID: 1
Source Address: 256
RSSI: 0
Option: 0
Sample Quantity: 16
Channel: 3
Check Sum: 108
Avg. Volt: 1346.7041
Avg. Ampr: 3.214158
Avg. Watt: 83.082794
Avg. Volt: 1662.8728
Avg. Ampr: 5.666235
Avg. Watt: -64.860115
Avg. Volt: 0.0
Avg. Ampr: 0.0
Avg. Watt: 0.0