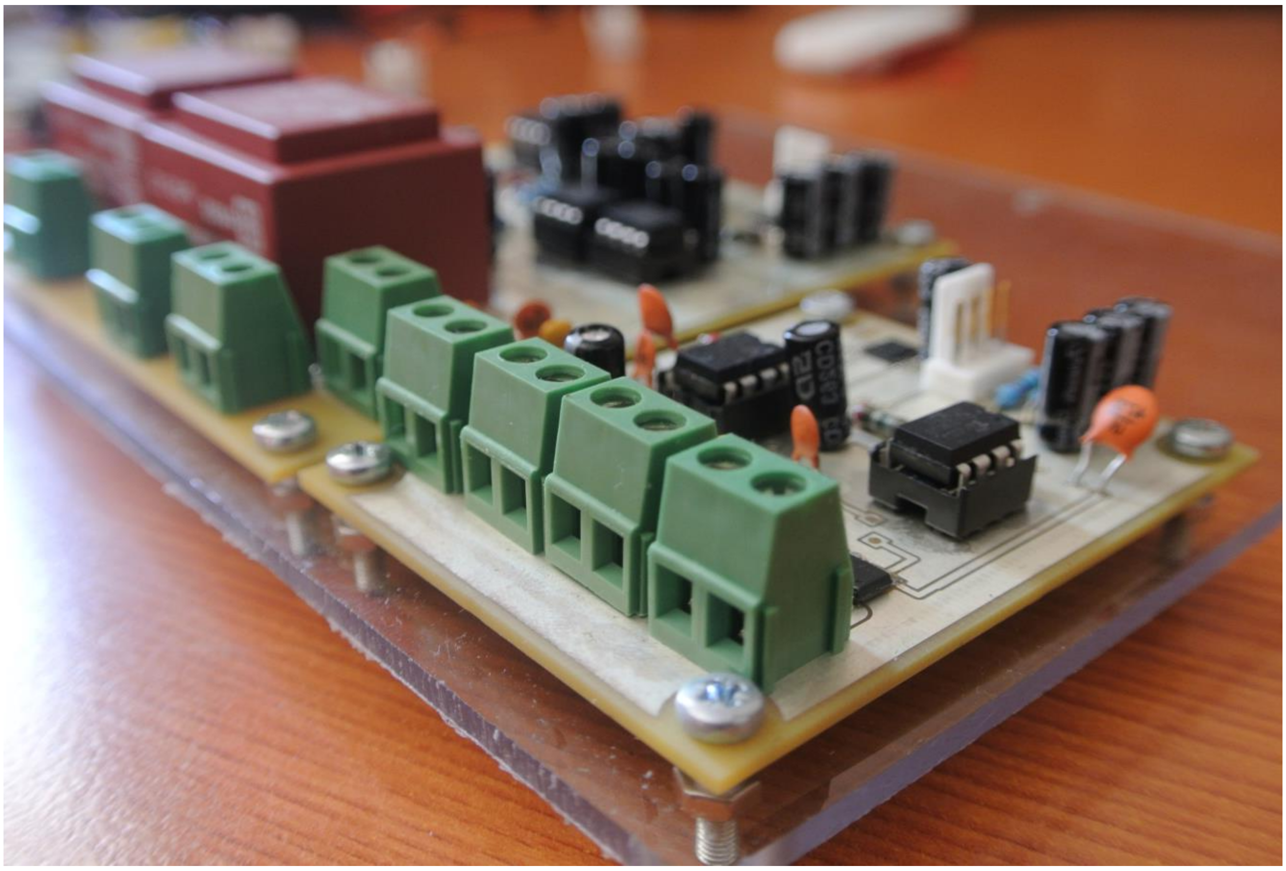
Datalogging 2012/2013

TECHNICAL DOCUMENTATION



Kishan Amratia
Alex Stott
Saurab Chhachhi
David Fofana

1. INTRODUCTION

This is a technical documentation highlighting the design, manufacture, testing and finance of a custom datalogging system for e.quinox's datalogging project. The project involves designing and building a system which remotely monitors the equipment in an energy kiosk in Rwanda and uploads the data collected to an online server.

2. TARGET OBJECTIVES

Considering the time, budget and resources available to us, we devised a set of manageable target objectives from which a minimum must be met for which we can consider the project a success. These are:

Minimum Target Objectives:

1. Design, Test and Build long-lasting AC and DC sensors or both Current and Voltage
2. Build a system which is simple, modular and within a certain budget
3. Store the data in an organized manner so that meaningful information can be collected from it
4. Write robust software that can recover from any software malfunctions autonomously
5. Upload the data to an online server for remote data access
6. Ensure smooth running of the system

Other Target Objectives:

1. Design a form of status indication on the system to allow quicker debugging for errors
2. Keep a log file which logs information about the connectivity and any errors that occur
3. Create a web interface to display the data in real-time
4. Design, Test and Build other useful sensors
5. Install a two-way remote communication or control of the system

Note: the minimum target objectives have been prioritized for this project and if time prevails then the other target objectives may be met as well.

3. SYSTEM CONCEPT

The concept behind the datalogging system (aka datalogger) is to have a microcontroller connected to several separate modules of the system. Each module will contain circuitry and components to fulfil its tasks. The Modules are:

1. Power Module to provide power the entire system
2. Sensor Module to sense relevant data from the equipment in the energy kiosk
3. Internet Module to allow communication with a server
4. Storage Module to store the data that we sample from the sensors

The above modules will be further discussed in the PCB design section of this documentation. For the internet module, an unlocked version of the Vodafone K3770 USB Modem was used since we are operating the device in Rwanda. Instructions on how to unlock the modem can be found on the e.quinox datalogging wiki or can be found online.

The microcontroller of choice for this project is the MBED NXP LPC1768 (Simply known as the "mbed"). This is due to the following factors:

- The mbed is an easy prototyping platform
- Most of the team is familiar with it
- There are several libraries available
- It is an open source platform and therefore fitting in with e.quinox's values
- It has networking capability with a USB modem which is a common internet connection method in Rwanda
- It is relatively low power

4. SENSOR DESIGN

There are 4 different types of sensors that are needed to fulfil the minimum target objectives. These are:

1. DC Voltage sensor
2. DC Current sensor
3. AC Voltage sensor
4. AC Current sensor

It is important to understand the energy kiosk model to know where we will be connecting the sensors and measuring the relevant data from.

4.1 The Energy Kiosk Model

Below is a simple diagram showing the energy kiosk model.

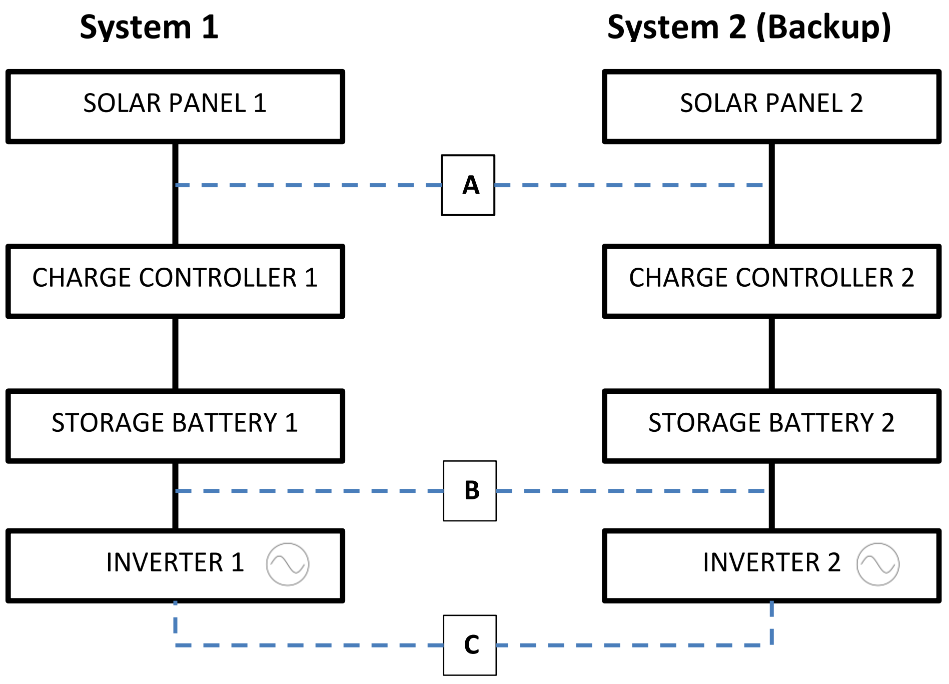


Figure 1: Energy Kiosk Model.

Figure 1 shows the energy kiosk model. In each energy kiosk there are two separate systems in place. This is a precautionary measure in case one of them is not working. The letters A, B and C with the blue dashed lines indicate the points at which the datalogger sensors will be connected to measure the data.

The whole system is primarily DC until the Inverter stage which outputs single phase AC Voltage at 230V (50Hz) hence the symbol ∿ in figure 1. Below is a breakdown of the type of sensors needed at each point.

Point	System 1	System 2
A	DC Current, DC Voltage	DC Current, DC Voltage
B	DC Current, DC Voltage	DC Current, DC Voltage
C	AC Current, AC Voltage	AC Current, AC Voltage

According to the table above, the following sensors are needed to fulfil the minimum target objective:
- 4 DC current sensors
- 4 DC voltage sensors
- 2 AC Current sensors
- 2 AC voltage sensors

This makes a total of 12 sensors.

4.2 General Sensor Design Rules

To ensure safety of the datalogging equipment, the sensors have to be physically isolated from the kiosk's equipment while still being electrically connected. This is so that the microcontroller does not get damaged from any fluctuations in voltage/current. The sensors are basically circuits designed to step down the voltage/current to an acceptable level so as to allow the microcontroller to measure it. It is essentially important that the sensors have a linear relationship between the input and the output so that the values measured by the microcontroller can easily be mapped to the original value sensed.

The reference voltage we will be using on the MBEDs ADC is 5V so the stepped down version of the original measured value should be between 0 and 5V.

4.3 DC Voltage Sensor

This sensor will be used to measure the voltage at points A and B. The maximum value that the sensor should be able to measure is about 20V from point A i.e. the solar panel. For this we can use a basic potential divider circuit with an isolating chip (IL300) to provide the isolation. The IL300 is a basic optocoupler which requires two supply voltages for it to work.

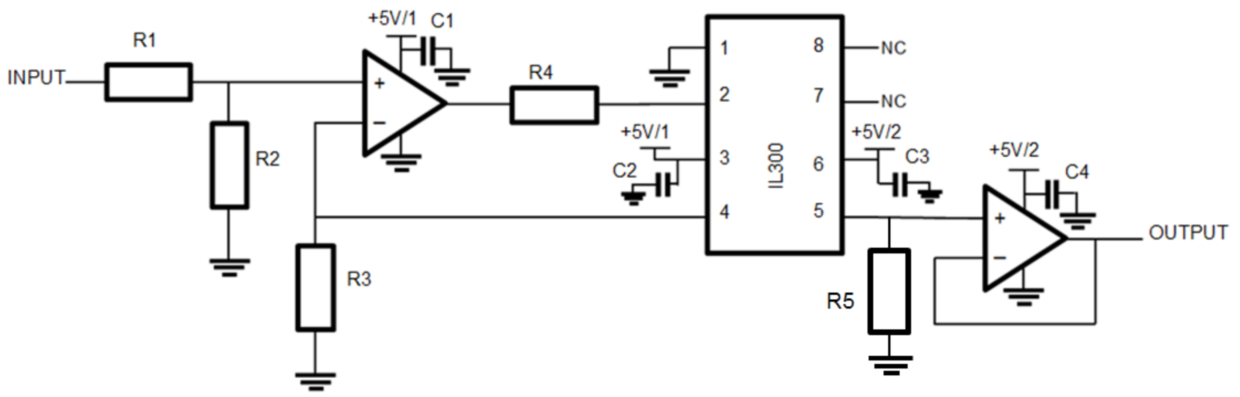


Figure 2: Circuit diagram of the DC Voltage sensor

The circuit shown in figure 2 is an application circuit taken directly from the IL300 datasheet. The potential divider at the start i.e. R1 and R2 were added at the input to scale the voltage down. The opamps used were MCP602 rail-rail single ended supply dual opamps.

Note that this circuit requires two different supplies for the isolation stage. The 5V/1 and 5V/2 are shown on the circuit diagram to differentiate between the two supplies.

Below is a table of the component values. These values were calculated using the datasheet and noting that the maximum voltage that we are expected to measure which is 20V.

Component	Value
R1	1kΩ
R2	4kΩ
R3	100Ω
R4	30kΩ
R5	30kΩ
C1	1µF
C2	1µF
C3	1µF
C4	1µF

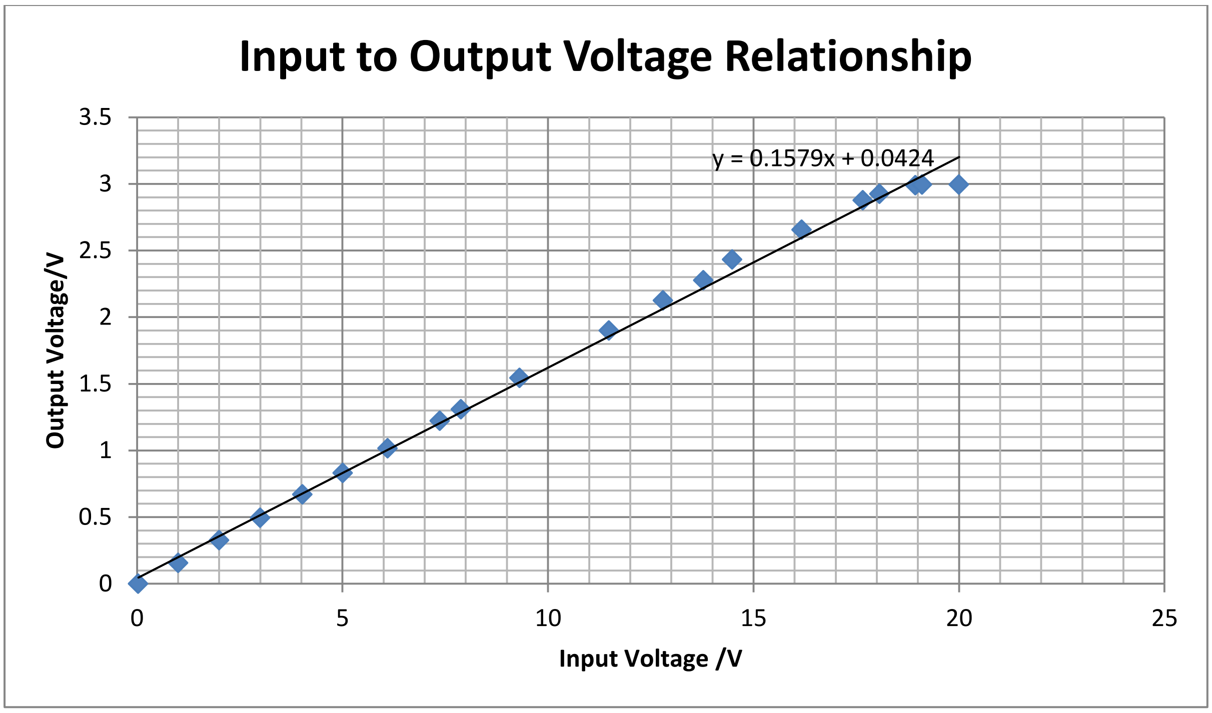


Figure 3: Variation of the input to the output voltage of the DC voltage sensor.

The graph has a linear relationship and therefore the sensor will work quite well. The initial potential divider can be scaled in order to accommodate a higher input voltage range if needed.

4.3 AC Voltage Sensor

This sensor needs to measure the AC voltage from point C i.e. the inverter. The circuitry for this sensor is fairly complicated since we need to ensure that the AC wave that we measure is between 0 and our reference voltage which is 5V.

To do this, a transformer is used to step down the voltage and then we add a DC bias so that the wave is within 0 and 5V rather than the wave oscillating around zero. The transformer also provides physical isolation as a transformer has no physical connections between the primary and secondary coils.

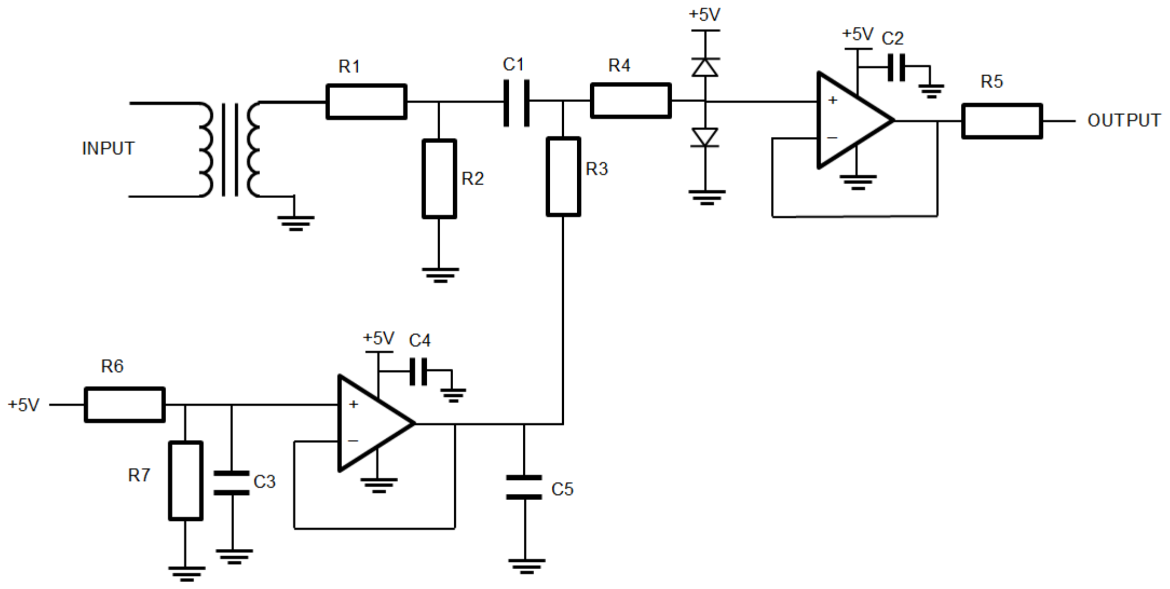


Figure 4: Circuit Diagram of the AC voltage sensor

In this circuit, there was a lot of experimentation of the transformer and resistor values involved to ensure that the waveform at the output was acceptable. The transformer we used was rated at 230:6 but due to the poor regulation of small transformers, the output voltage at 230V input was close to 8V (These quoted results are RMS).

The diodes we used were standard schottky diode's (SD103) to ensure that the signal does not become negative. The op-amps used are the same as in the voltage sensor i.e. MCP602. The Transformer used is a MYRRA 1VA 6V PCB mount transformer.

Below is the table of components and component values used. These values were carefully calculated to ensure that the signal is acceptable. R1 and R2 is a potential divider which scales the voltage down. R6 and R7 creates a 2.5V DC bias which is then added to the AC signal after the capacitor C1. The capacitor values chosen are to ensure clean signals and to reject high frequency noise. The value of R5 was calculated as a current limiting resistor for a maximum current of 10mA.

Component	**Value**
R1	7.8kΩ
R2	1.1kΩ
R3	5.1kΩ
R4	5.1kΩ
R5	500Ω
R6	5.1kΩ
R7	5.1kΩ
C1	47μF
C2	1μF
C3	1μF
C4	1μF
C5	1μF

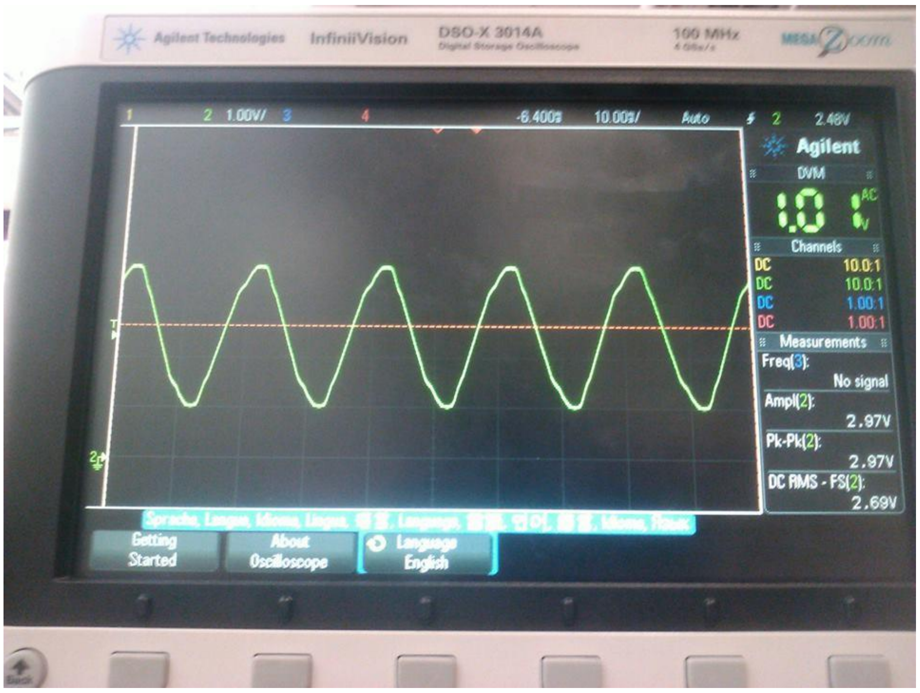


Figure 5: The output waveform when 230V is input into the circuit.

Figure 5 shows that the output is well within the specified range i.e. it is about 2.97V pk-pk and within 0 and 5V.

4.4 Current Sensor

The current sensor is designed to work for both AC and DC. We will be using a Hall-Effect Current Sensing chip (ACS712). It uses the Hall Effect to convert the current into a proportional voltage. This provides isolation with the microcontroller.

The reason that this chip works for both DC and AC is because the output of the chip is offset in the middle of the supply voltage. If you have an oscillating current, the output voltage from the chip will oscillate around the offset voltage which is generally around half the supply voltage.

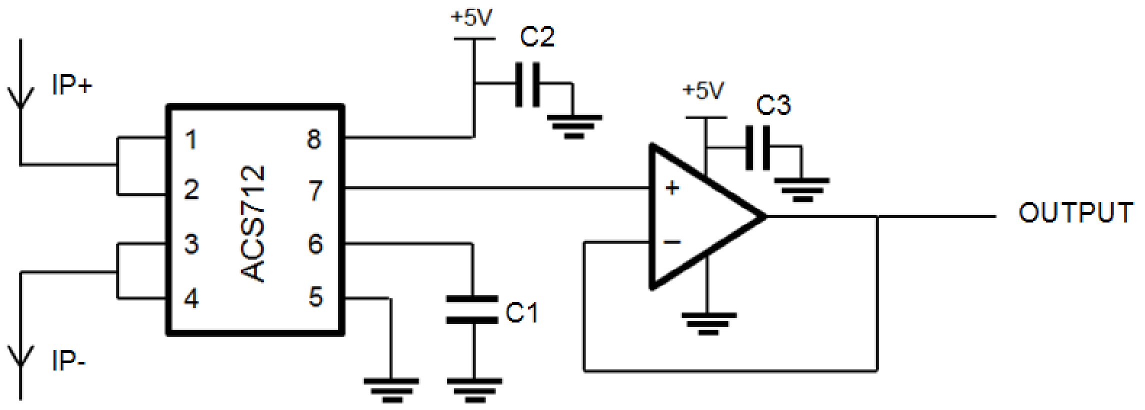


Figure 6: Circuit Diagram of the Current sensor. The IP+ and IP- denotes the current that will flow through the chip in a certain direction.

The circuit used is a typical application circuit taken from the datasheet. Below is a table of the component values used.

Component	Value
C1	1nF
C2	100nF
C3	1µF

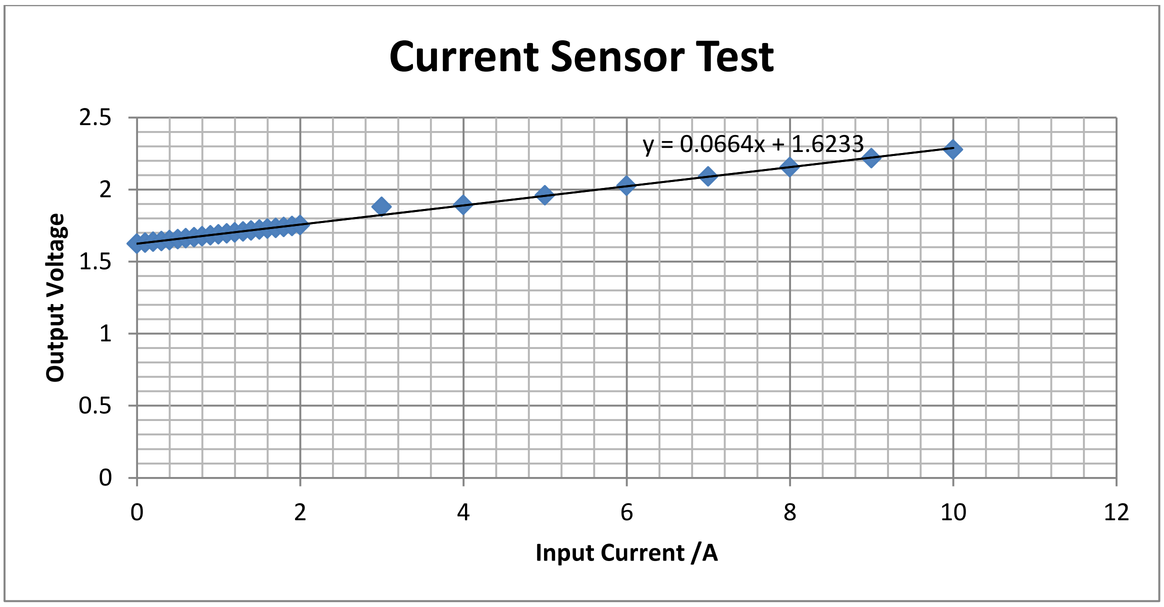


Figure 7: Graph showing the input/output relationship of the current sensor circuit. Note: the graph shows results of a similar circuit as in figure 6 but had a potential divider at the end. However, the main idea was to show that the graph is linear.

5. PCB DESIGN

The PCBs that are needed for the system depend on the number of sensors and the how we want to connect the different modules together. We have split the PCBs in the following way:

1. Main Board – contains the microcontroller, internet module and storage module
2. Power Board – basically the circuitry for the power module
3. AC Sensor Board (2 current sensors and 2 AC voltage sensors)
4. DC Current Board (4 current sensors)
5. DC Voltage Board (4 DC sensors)

Because of the vast number of sensors and the limited amount of pins available on the MBED to connect the output of the sensors to, the I2C protocol was needed to send the values into the mbed. An extra ADC I2C enabled chip is needed to allow communication over the I2C Bus of the mbed. This will also keep the system modular since many different sensors can be added onto the bus in the future. The ADC is the AD7993/4.

5.1 Connection Diagram

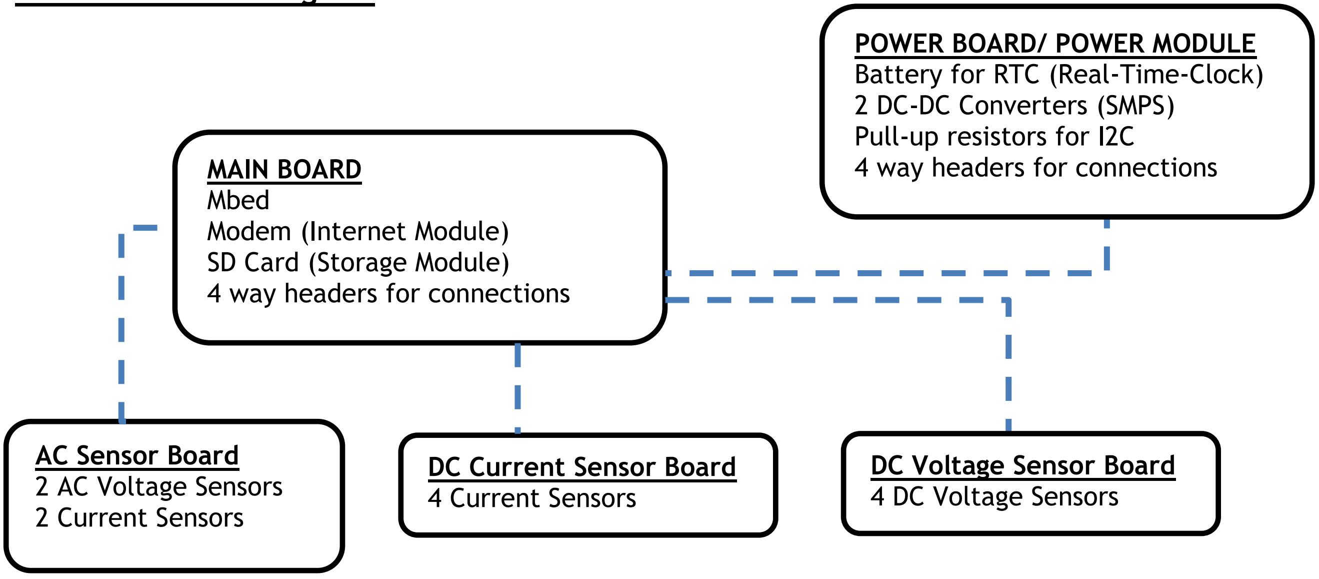


Figure 8: A diagram showing the main connections of the datalogger between the different modules and boards.

5.2 Main Board

To keep the system simple, it was decided to use a readily available MBED development board which had features such as an SD card slot and USB input for the modem. This was ideal as it had all the features needed on the main board.

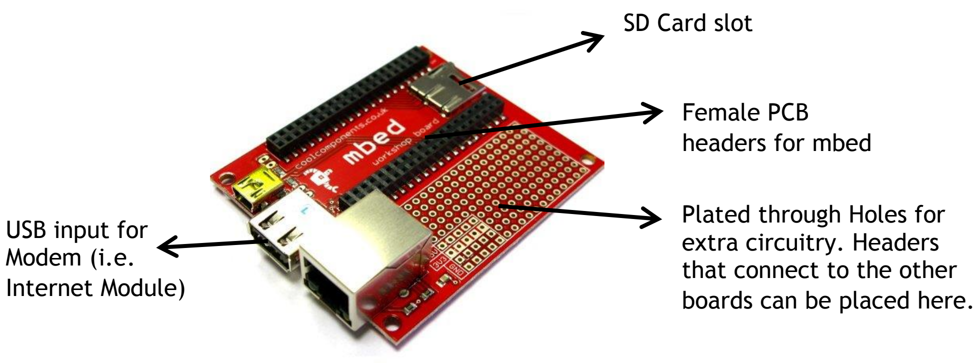

SD Card slot
Female PCB headers for mbed
Plated through Holes for extra circuitry. Headers that connect to the other boards can be placed here.
USB input for Modem (i.e. Internet Module)

Figure 9: Image of the mbed development board (image taken from the cool components website)

5.3 Power Board

This board powers the entire datalogger components. It will have the switch mode power supplies (R-78E5) on it which will use the 2 storage batteries in the energy kiosk to give 2 5V supplies. The significance for 2 of them comes from the fact that the DC Voltage sensor needs 2 separate supplies.

The power board also has the external battery for the real-time clock. This is a CR2032 3V Phillips coin battery which should last a very long time considering the RTC only requires a couple of micro Amps to work when there is no power in the MBED.

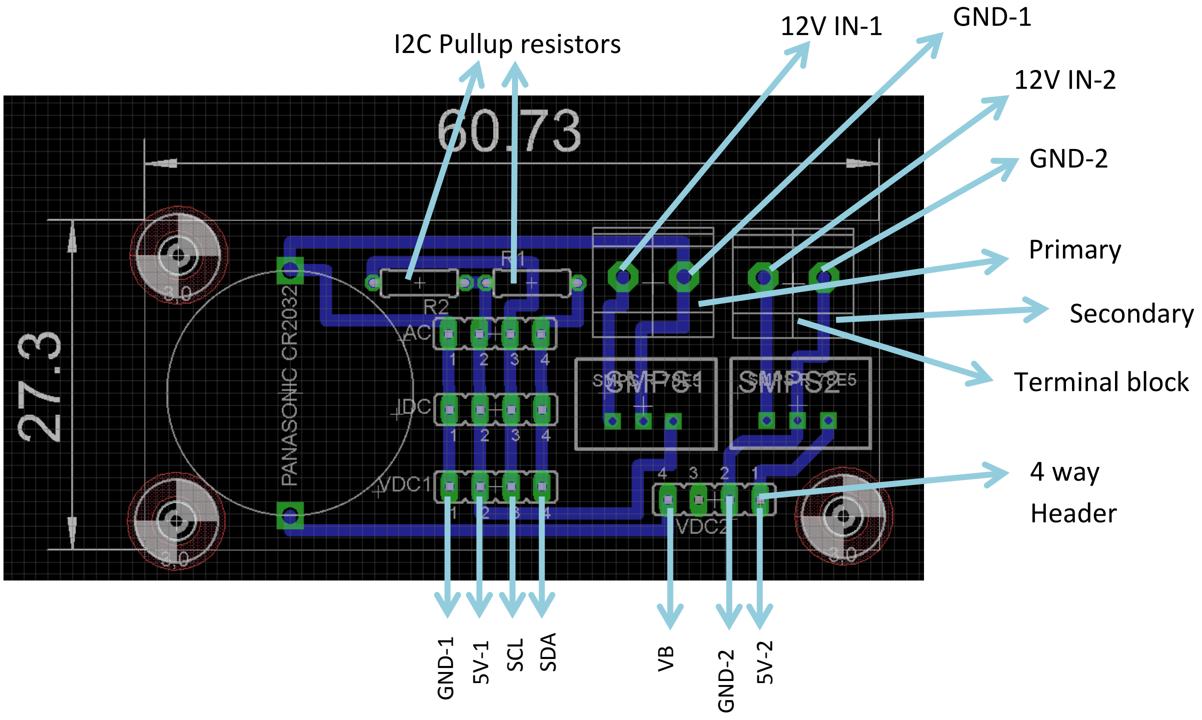


Figure 10: The Power Board layout.

The primary terminal block will power everything except for the input side of the IL300 in the DC Voltage sensors whereas the secondary terminal block will power the input side of the IL300. *The labelling of the headers will be consistent with the rest of the PCB layouts.*

5.4 DC Voltage Board

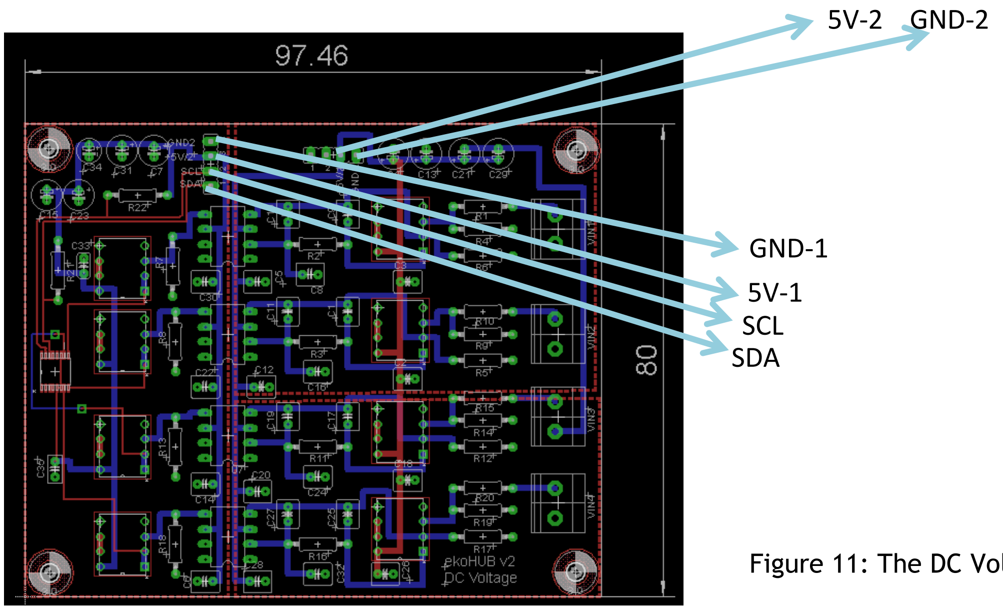


Figure 11: The DC Voltage Board layout.

5.4 DC Current Board

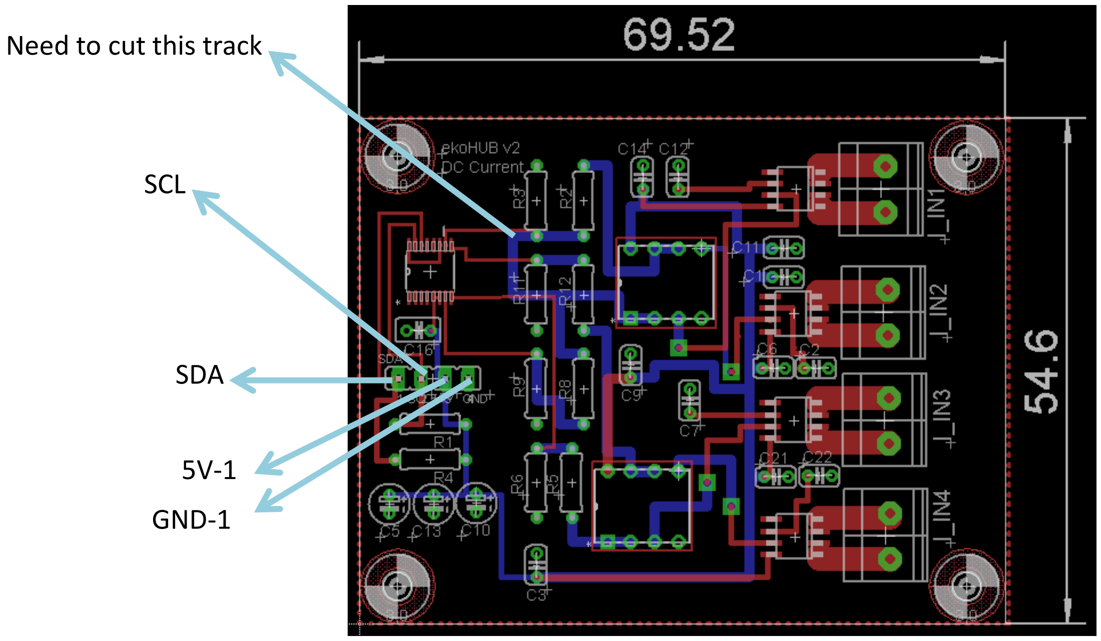


Figure 12: The DC Current Board layout.

There is a mistake on the board above and this was noticed after it came back from manufacturing. One of the tracks needed to be cut but luckily this wasn't too difficult and the PCB worked well in the end. The track that needed cutting is shown in the diagram above. Something to note on this PCB is that since there are possibilities of high currents at the input, the track width needs to be large in order to accommodate this.

5.5 AC Voltage Board

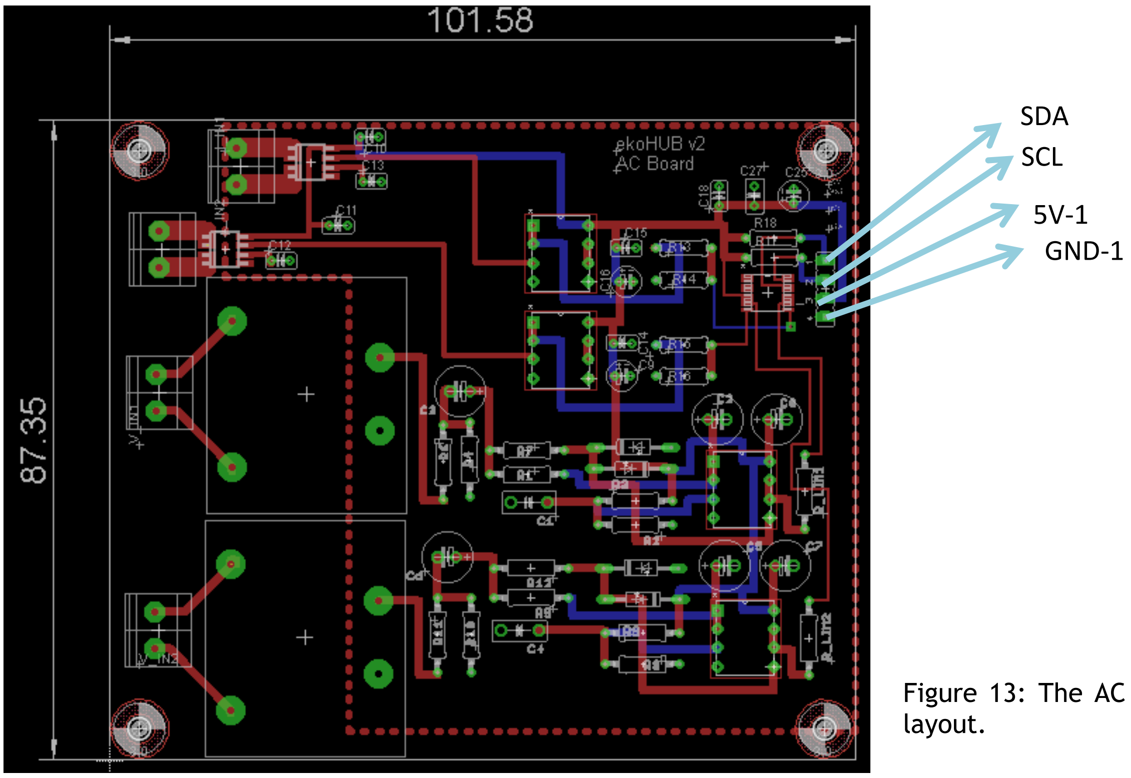


Figure 13: The AC Voltage Board layout.

An important thing to note about the AC board is that it is a high voltage board and so there must be a minimum clearance between the high voltage tracks of atleast 3.5mm to prevent arcing. Also, there shouldn't be any tracks that pass underneath the transformer because this could cause induction within the transformer and give you wrong values.

Note: All of the PCBs were made using Eagle. The designs can be found on the e.quinox wiki or on the e.quinox datalogging Github repository.

6. DATALOGGER CASING /BOX

The datalogger circuitry needs to be encased in a box which has some switches and some minor control of the device. The battery box team at e.quinox had some sample aluminium extruded IP65 rated electrical encasings which the datalogging team were able to use. Holes were drilled on the box to fit the switches, LED holders, power jack inputs and rubber grommets.

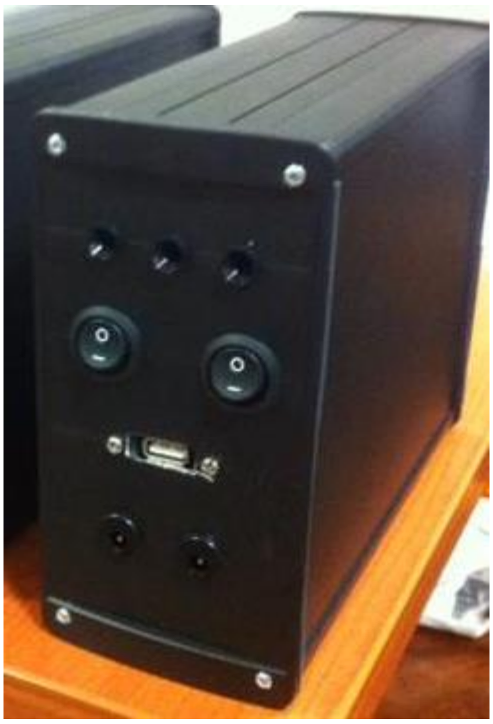


Figure 14: Picture of the front of the datalogger box/casing. The back has three rubber grommets to allow cables to go in.

7. SOFTWARE

The datalogger needs some well-written software that runs smoothly and can recover automatically from a failure. The software is written in C/C++ in the mbed online compiler. This section of the documentation simply explains the program and code in a general sense and gives reasons for certain decisions made.

The program has to fulfil the following functions as a minimum:

- Make sure that the time is synchronized with the actual time
- Sample and store sensor data into SD card during certain times of the day
- Upload the data
- Stay in an idle state during certain times of the day/night

The program will use various pre-built libraries and functions which are easily available on the mbed online compiler:

1. Vodafone USB Library
2. SD File System Library
3. HTTPClient Library
4. NTPClient Library
5. PowerControl Library

Note that this section of the documentation does not document the development of the code but only describes the final version. In reality, the program went through a number of iterations and revisions to ensure optimum performance.

Notes on the software:

- Although it is rare, when an internet connection is setup with the modem, the code can get stuck in an infinite loop. The reason for this could not be determined. Therefore watchdog timers have been placed in the code to ensure that the code can recover from any infinite loops which are bound to happen but are rare when the modem tries to connect to the internet.

- The initial plan was to upload the data at night when we are not gathering any data. However it was decided to send the data and log files immediately after sampling the sensors. This is because there is a risk of the data not being uploaded at night due to the large chunk of data. Therefore, to ensure that the data gets uploaded fully, it is less risky to upload a smaller chunk but upload more frequently. This way, it is guaranteed to receive upwards of 80% of the data on the server. However, if the upload fails, the data is still stored on the SD card and a note is made that the upload failed.

- Status indication has been implemented in the system. There are 3 LEDs on the front panel of the box. There is a red one and two green ones. The indication happens as follows:
    - Red LED stays on to indicate that there is power connected to the datalogger
    - The green LED1 turns on when the modem tries to connect to the internet
    - The green LED2 turns on when the datalogger is sampling the data from the sensors
    - When an upload fails, both green LEDs will stay on for about 4-5 seconds
    - When an SD card is not detected, the Red and Greed LED1 will flash continuously until an SD card is detected

- Status indication for the scenario when the modem is not physically connected did not work reliably. It is not possible to do reliably with software and therefore it cannot be implemented without some hardware changes especially to the PCB which is not an option but can be in the future.

- To try and conserve power, the mbed was put into a low power state. There is a possibility to put it into a deep power down when we are not using it and consume extremely little amount of power. It was not possible to wake the chip up from that state to resume normal operation. However, some power control features have been implemented to turn off unused peripherals like the I2C Clock so that there is little power wasted.

- The apn must be changed to the network that you are working on to allow an internet connection to be made.

7.1 Software Flow Diagram

Refer to appendix (Section 12.1) for the diagram.

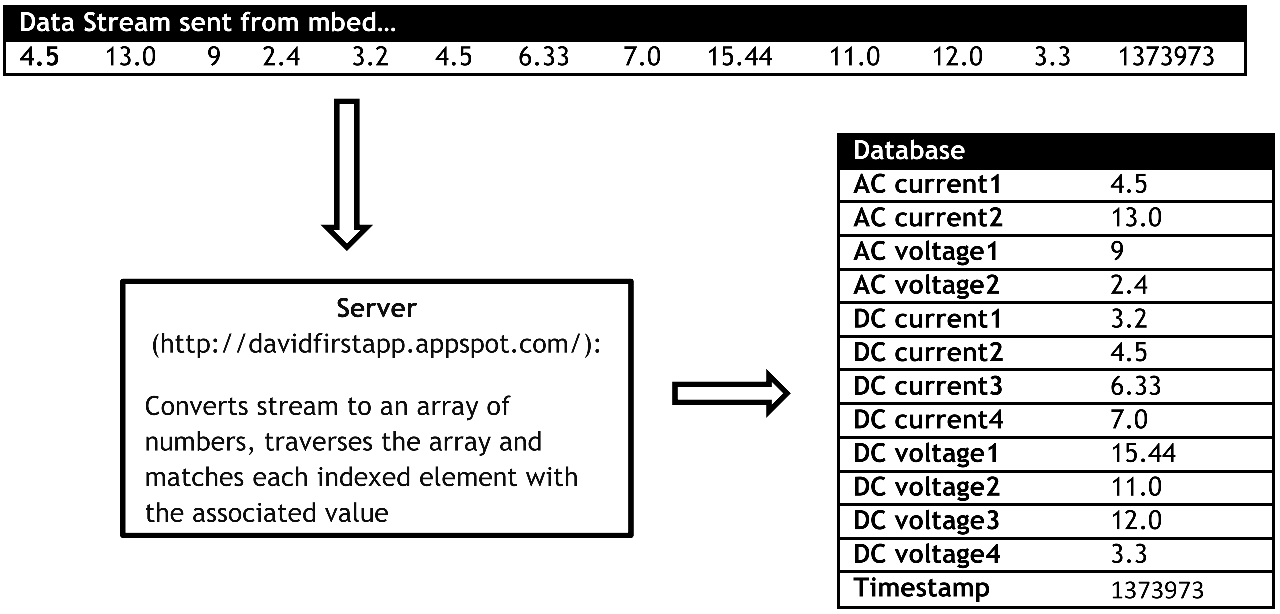
Commonly used abbreviations:
RTC – Real Time Clock
NTP – Network Time Protocol

The diagram does not use conventional flow chart shapes. This is because the shape for a conditional step (diamond) restricts the amount of text that can be written within it. Therefore, a colour code is used instead. The diagram ignores some of the processes like the status indication and setting the watchdog timers for simplicity.

7.2 Server Side

The web server was developed under the Google App Engine platform, using the python language. Its main purpose is to receive the data sent from the mbed, parse it accordingly and store it in an online database, within the App Engine cloud domain. The diagram below explains in a simple example how the data is saved. The order in which the server saves the data corresponds to the order in which they are sent from the mbed, which sends 13 values, as shown in the newly updated database.

Data Transfer between mbed and web server:



Data Stream sent from mbed...												
4.5	13.0	9	2.4	3.2	4.5	6.33	7.0	15.44	11.0	12.0	3.3	1373973

Server
(http://davidfirstapp.appspot.com/):

Converts stream to an array of numbers, traverses the array and matches each indexed element with the associated value

Database	
AC current1	4.5
AC current2	13.0
AC voltage1	9
AC voltage2	2.4
DC current1	3.2
DC current2	4.5
DC current3	6.33
DC current4	7.0
DC voltage1	15.44
DC voltage2	11.0
DC voltage3	12.0
DC voltage4	3.3
Timestamp	1373973

The database is an efficient and scalable datastore, a big table where at least 1GB worth of data can be saved across multiple entries. The data storage is more than sufficient for the total size of data being sent.

In addition to saving the sensor values, the server also saves the log information sent from the mbed. It also provides other functionalities such as a timeserver, from which clients can fetch timestamps of specific time zones. The end product of the web application will have a user interface, which will display according to the user's selection and filtering options the values measured from the different equipment and systems being monitored. This will hopefully be accompanied with graphs and other useful plots to give users a complete and useful description of how the datalogging system and its associated modules are behaving

8. TESTING

8.1 Sensor Testing

The sensors were all tested so that an exact equation that governs the input/output relationship can be found. This allows conversion of the decimal value of the sensor ADC output back to the original input value. This was done by sweeping the input and recording the output value. The equations of the linear graphs are shown in the table below.

	<u>DC Current Board</u> (y is a decimal value beween 0 and 1023) (x is in mA)	<u>DC Voltage Board</u> (y is a decimal value beween 0 and 1023) (x is in V)
Sensor 1	$y = 0.0207x + 510.64$	$y = 32.339x + 8.6802$
Sensor 2	$y = 0.0199x + 512.8$	$y = 32.3858x + 8.504$
Sensor 3	$y = 0.0206x + 511.65$	$y = 32.337x + 10.64$
Sensor 4	$y = 0.0205x + 508.81$	$y = 33.006x + 10.14$

For the AC board, it isn't necessary to measure the input/output relationship simply because it is an expected AC wave being measured at the kiosk i.e. 230V RMS at 50Hz. In terms of the AC current, this will fluctuate according to the load on the inverters in the kiosk. Therefore it is not entirely possible to fully test the AC current sensors until installing the system, testing it and adjusting the software accordingly to reflect the correct waveform.

8.2 Power Consumption Tests

Some power consumption tests were performed to see how much power on average the datalogger is consuming. The power consumed at each state was measured and the results were as follows:

Datalogger State	Power Consumed (W)
Getting Internet time	2.52
Sampling the sensors	1.44
Data upload	2.52
Log Upload	2.52
Idle	1.26

Using this, an estimate of the average power consumption of the datalogger of a typical day can be made. A typical day for example is where the datalogger starts sampling and uploading data between 7AM and 7PM. This value is estimated at approximately **1.32W.**

In comparison to a standard energy saving light bulb which has a wattage at around 10-15W, having 7 dataloggers in one kiosk is equivalent to having one 10W energy saving light bulb on for 24Hrs. In other words, the datalogger consumes 13% of energy on average of that of a standard 10W energy saving light bulb.

9. FINANCE

The target was to spend under £1000 on the development of the project. The target cost for a finished datalogger was around £200. In total the cost was £900 to research, develop, test and build 2 dataloggers. Below is a breakdown of the costs of components used within a datalogger.

Component	Quantity	Unit Cost (£)	Total (£)
MBED LPC1768	1	41.38	41.38
Vodafone K3770 USB Modem	1	9.99	9.99
MBED Development Board	1	21.95	21.95
PCB Manufacture PCB Train			
- AC Board	1	28.065	28.065
- DC Voltage Board	1	25.44	25.44
- DC Current Board	1	18.13	18.13
Terminal Blocks	14	0.28	3.92
MCP602 OPAMP	14	0.46	6.44
MYRAA 1.5VA Transformer	2	2.83	5.66
IL300	4	3.30	13.20
AD7993	4	2.75	11.00
PCB Headers	14	0.49	6.86
Crimps	112	0.058	6.42
Crimp Housing	28	0.026	0.73
ACS712	6	3.23	19.38
SD103 Diodes	4	0.03	0.12
R78E5 DC-DC Converter	2	2.31	4.62
Panasonic CR2032	1	1.10	1.10
		Grand Total	**224.44**

The table above shows only the costs incurred by a single datalogger. However, some of the items listed on the table had to be bought in excess therefore excess components were left over and these are not accounted for in the table but are accounted in the total figure which amounted to **£887.64**.

Also, some of the unit cost listed is as per the bulk discount we received.

Most of the items were ordered from Farnell, RS and Cool Components.

10. EVALUATION AND FUTURE CONSIDERATION

10.1 Evaluation

To evaluate the system completely, we must let it run in Rwanda for a few months and then make a full evaluation according its performance. However, this documentation has been made prior to the installation of the datalogger and the full evaluation will be done on the expedition report.

Measuring the system against our target objectives, it seems that most of the minimum target objectives have been met. There are certain objectives which cannot be measured qualitatively such as longevity of the sensors and smooth running of the system due to time. However, the datalogger was tested for about 10 days leaving it to run on its own and it worked fine without any major errors.

Some of the other target objectives were also fulfilled such as designing the status indication and log files which give important information of the internet connection properties.

In terms of the finances, the team could have done slightly better in reducing the costs. The PCB manufacture was slightly too expensive. Some other manufacturers (Seeed Studio from China) were much cheaper but their delivery time was unreliable for the time that was left to finish the project.

10.2 Future Consideration

A few observations were made during the project where improvement is needed in the next design of the datalogging system. These are listed below:

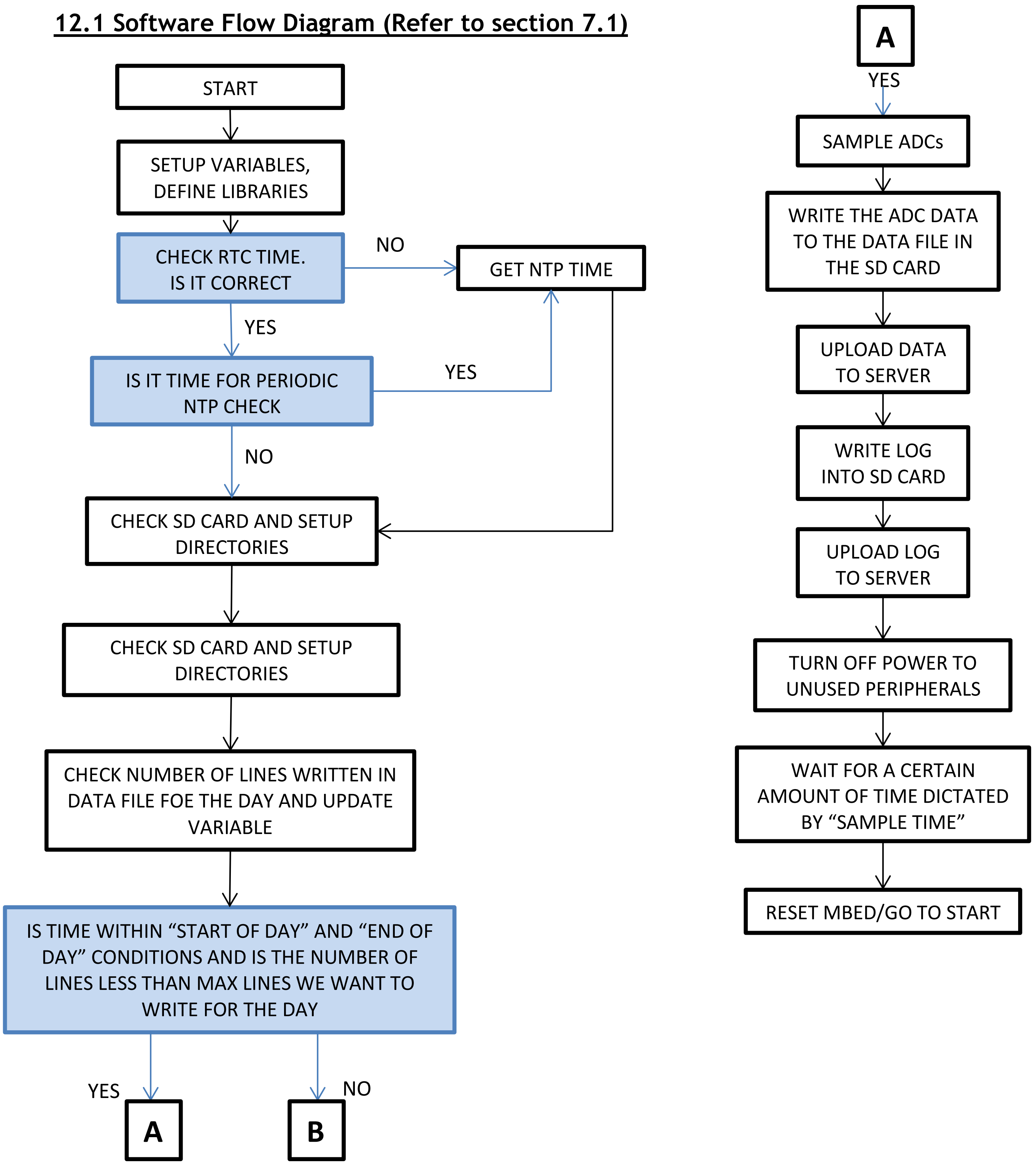
- The current sensor needs to be more sensitive to be able to be more accurate. It simply needs a minor change in the circuit.
- Design of a voltage sensor that does not require 2 supplies as this is slightly wasteful.
- The DC-DC Convertors used on the power board need to be isolating ones just for safety.
- Reduce the cost of the datalogger by designing our own main PCB instead of ordering the mbed development board which costs about £20 which is slightly high for what it is. Also, since this board was used, there was a lot of wiring that needed to be done between the PCB headers and the actual pins of the mbed. This is not good for replication of the datalogger for mass production so it will be much better to make a custom PCB instead. Also currently, it increases the debugging time needed if any errors occur.
- Another modification is that we should use a faceplate PCB design for the main PCB like the latest battery box design. This will also reduce the connections on the front plate of the datalogger box and therefore reduce the debugging time if and when errors occur. It will also make the datalogger tend towards looking like a more finished product.
- The modem's 5V pin needs to be connected to one of the mbed's digital out so that its power consumption can be controlled as the modem consumes the most power even if it is not in use.
- The arduino can be used instead of the MBED as an experiment. This is because, it is cheap, replicable, lower power and a GSM module is readily available. However, note that the mbed has a much better performance in general, but further testing is required.
- There is some documentation on the mbed website about converting the mbed into a custom PCB. This would reduce the cost a lot since the LPC1768 chip costs around £8 which when compared to the cost of the mbed which is £41 is a big reduction.
- Having an LCD display on the box could be useful as well instead of the LED status indication.

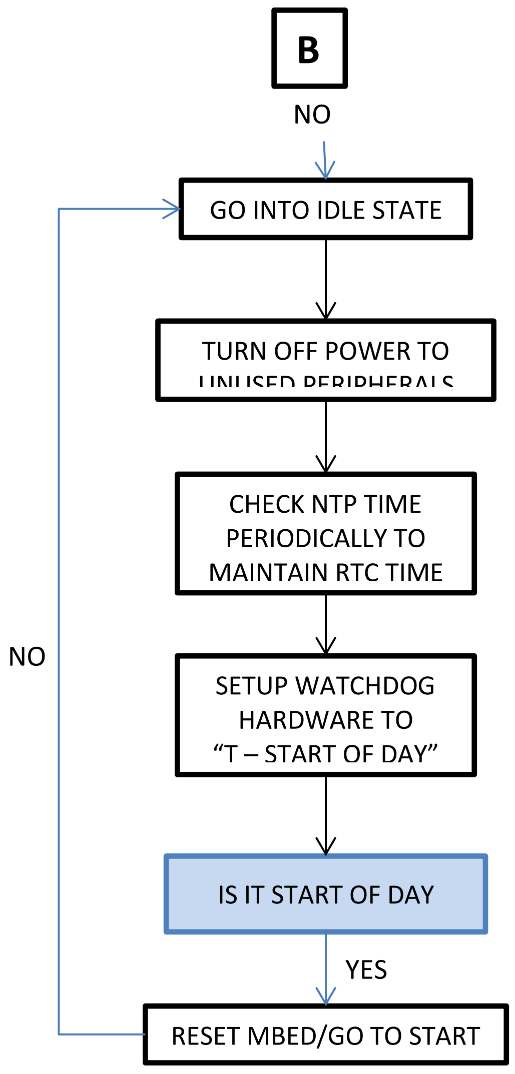
11. ACKNOWLEDGEMENTS

The datalogging team at e.quinox would like to thank Tom Berman for his help with the server side of the system.

12. APPENDIX

12.1 Software Flow Diagram (Refer to section 7.1)


START
SETUP VARIABLES, DEFINE LIBRARIES
CHECK RTC TIME. IS IT CORRECT
NO
GET NTP TIME
YES
IS IT TIME FOR PERIODIC NTP CHECK
YES
NO
CHECK SD CARD AND SETUP DIRECTORIES
CHECK SD CARD AND SETUP DIRECTORIES
CHECK NUMBER OF LINES WRITTEN IN DATA FILE FOE THE DAY AND UPDATE VARIABLE
IS TIME WITHIN "START OF DAY" AND "END OF DAY" CONDITIONS AND IS THE NUMBER OF LINES LESS THAN MAX LINES WE WANT TO WRITE FOR THE DAY
YES
A
NO
B
A
YES
SAMPLE ADCs
WRITE THE ADC DATA TO THE DATA FILE IN THE SD CARD
UPLOAD DATA TO SERVER
WRITE LOG INTO SD CARD
UPLOAD LOG TO SERVER
TURN OFF POWER TO UNUSED PERIPHERALS
WAIT FOR A CERTAIN AMOUNT OF TIME DICTATED BY "SAMPLE TIME"
RESET MBED/GO TO START


B
NO
GO INTO IDLE STATE
TURN OFF POWER TO UNUSED PERIPHERALS
CHECK NTP TIME PERIODICALLY TO MAINTAIN RTC TIME
SETUP WATCHDOG HARDWARE TO "T – START OF DAY"
IS IT START OF DAY
YES
RESET MBED/GO TO START
NO