|  |  |
| --- | --- |
| CURRENT TRASFORMER METER  CTM-3501 | Abstract  This report contains the design, implementation, and testing of a non-invasive near to real-time current metering protocol for alternating systems.  Michael Giorgas & Clinton Elliott  Electrical Engineering BA (Electronic) |

You had me at “Hello World”;

-Anonymous

TABLE OF CONTENTS

[1. INTRODUCTION 1](#_Toc497049116)

[2. OVERVIEW OF DESIGN 2](#_Toc497049117)

[3. HARDWARE DESIGN 3](#_Toc497049118)

[3.1. Schematic 3](#_Toc497049119)

[3.1.1. Power Supply 3](#_Toc497049120)

[3.1.2. Operational Input Shifters 3](#_Toc497049121)

[3.1.3. Multiplexer 4](#_Toc497049122)

[3.1.4. ZigBee 5](#_Toc497049123)

[3.1.5. M5 & H5 Processors 6](#_Toc497049124)

[3.1.6. Mini-USB 6](#_Toc497049125)

[3.1.7. JTAG and SDA Headers 7](#_Toc497049126)

[3.1.8. 3-Stage Contingency 7](#_Toc497049127)

[3.1.1. Power Saving 8](#_Toc497049128)

[3.1.2. Battery Selection 9](#_Toc497049129)

[3.2. Power Optimisation 10](#_Toc497049130)

[3.2.1. Operational Amplifiers 10](#_Toc497049131)

[3.2.2. Voltage Regulator 10](#_Toc497049132)

[3.2.3. XBee Pro 11](#_Toc497049133)

[3.2.4. MK20DX128VLH5 & MK20DX128VFM5 Processors 11](#_Toc497049134)

[3.2.5. SN74LVC125A Quadruple Bus Buffer Gate 12](#_Toc497049135)

[3.2.6. HEF4066B Quad Single-Pole Single-Throw Analog Switch(Multiplexer) 12](#_Toc497049136)

[3.2.7. Mini USB 12](#_Toc497049137)

[3.2.8. LED’s 12](#_Toc497049138)

[3.2.9. Battery Sizing 13](#_Toc497049139)

[3.3. PCB Design 15](#_Toc497049140)

[4. SOFTWARE DESIGN 17](#_Toc497049141)

[4.1. Kinetis Code 17](#_Toc497049142)

[4.1.1. Analog to Digital Converter 17](#_Toc497049143)

[4.1.2. Timing 17](#_Toc497049144)

[4.1.3. Data Percentage 17](#_Toc497049145)

[4.1.4. Data Calculation & Mapping 18](#_Toc497049146)

[4.1.5. ZigBee Transmission 19](#_Toc497049147)

[4.2. C++ Code 21](#_Toc497049148)

[4.2.1. ZigBee Data Acquisition 21](#_Toc497049149)

[4.2.2. WiPi Upload 23](#_Toc497049150)

[4.2.3. ThingSpeak Display 23](#_Toc497049151)

[4.3. Webpage Code 25](#_Toc497049152)

[4.3.1. Hosting 25](#_Toc497049153)

[4.3.2. Hyper Text Markup Language (HTML) 25](#_Toc497049154)

[4.3.3. Cascading Style Sheet (CSS) 25](#_Toc497049155)

[4.3.4. File Transfer Protocol (FTP) 26](#_Toc497049156)

[5. TESTING & RESULTS 27](#_Toc497049157)

[5.1. Real-World Set Back 27](#_Toc497049158)

[5.2. Final Product 28](#_Toc497049159)

[5.1. Webpage Display 29](#_Toc497049160)

[5.2. Game-Day Performance 30](#_Toc497049161)

[5.3. Results 31](#_Toc497049162)

[6. LIMITATIONS & RECOMMENDATIONS 32](#_Toc497049163)

[6.1. Limitations 32](#_Toc497049164)

[6.2. Recommendations 32](#_Toc497049165)

[7. CONCLUSION 33](#_Toc497049166)

[8. REFERENCES 34](#_Toc497049167)

[APPENDIX A – SCHEMATIC 1](#_Toc497049168)

[APPENDIX B – PCB LAYOUT 2](#_Toc497049169)

[APPENDIX C – BILL OF MATERIALS 3](#_Toc497049170)

[APPENDIX D – CODE 4](#_Toc497049171)

[APPENDIX E – FINAL PRODUCT 5](#_Toc497049172)

LIST OF FIGURES

[Figure 1 - Current Transformer [1] 1](https://d.docs.live.net/c07c9239cf8ef362/4th%20Year%20Sem2/CC3501/Github/CT-Monitoring-Assignment/CTM%20-%20Report%20%5e0%20Manual/CT-Metering-Assignment-R_4.docx#_Toc497053455)

[Figure 2 -Overview of Operation 2](#_Toc497053456)

[Figure 3 - Power Supply and Corresponding Protection 3](https://d.docs.live.net/c07c9239cf8ef362/4th%20Year%20Sem2/CC3501/Github/CT-Monitoring-Assignment/CTM%20-%20Report%20%5e0%20Manual/CT-Metering-Assignment-R_4.docx#_Toc497053457)

[Figure 4 - Operational Amplifiers and Input Shifting Circuit 4](https://d.docs.live.net/c07c9239cf8ef362/4th%20Year%20Sem2/CC3501/Github/CT-Monitoring-Assignment/CTM%20-%20Report%20%5e0%20Manual/CT-Metering-Assignment-R_4.docx#_Toc497053458)

[Figure 5 - Multiplexer 4](https://d.docs.live.net/c07c9239cf8ef362/4th%20Year%20Sem2/CC3501/Github/CT-Monitoring-Assignment/CTM%20-%20Report%20%5e0%20Manual/CT-Metering-Assignment-R_4.docx#_Toc497053459)

[Figure 6 - ZigBee Transmitting Device 5](https://d.docs.live.net/c07c9239cf8ef362/4th%20Year%20Sem2/CC3501/Github/CT-Monitoring-Assignment/CTM%20-%20Report%20%5e0%20Manual/CT-Metering-Assignment-R_4.docx#_Toc497053460)

[Figure 7 - Primary (left) and SDA (right) Processors and Oscillators (bottom right) 6](https://d.docs.live.net/c07c9239cf8ef362/4th%20Year%20Sem2/CC3501/Github/CT-Monitoring-Assignment/CTM%20-%20Report%20%5e0%20Manual/CT-Metering-Assignment-R_4.docx#_Toc497053461)

[Figure 8 - Mini USB Circuit 7](https://d.docs.live.net/c07c9239cf8ef362/4th%20Year%20Sem2/CC3501/Github/CT-Monitoring-Assignment/CTM%20-%20Report%20%5e0%20Manual/CT-Metering-Assignment-R_4.docx#_Toc497053462)

[Figure 9 – SDA (left) and JTAG (right) Headers 7](https://d.docs.live.net/c07c9239cf8ef362/4th%20Year%20Sem2/CC3501/Github/CT-Monitoring-Assignment/CTM%20-%20Report%20%5e0%20Manual/CT-Metering-Assignment-R_4.docx#_Toc497053463)

[Figure 10 – FRDM Board Headers 8](https://d.docs.live.net/c07c9239cf8ef362/4th%20Year%20Sem2/CC3501/Github/CT-Monitoring-Assignment/CTM%20-%20Report%20%5e0%20Manual/CT-Metering-Assignment-R_4.docx#_Toc497053464)

[Figure 11 - Coin Cell Battery and Isolator 8](https://d.docs.live.net/c07c9239cf8ef362/4th%20Year%20Sem2/CC3501/Github/CT-Monitoring-Assignment/CTM%20-%20Report%20%5e0%20Manual/CT-Metering-Assignment-R_4.docx#_Toc497053465)

[Figure 12 - Auto Routed Test of Placement 15](https://d.docs.live.net/c07c9239cf8ef362/4th%20Year%20Sem2/CC3501/Github/CT-Monitoring-Assignment/CTM%20-%20Report%20%5e0%20Manual/CT-Metering-Assignment-R_4.docx#_Toc497053466)

[Figure 13 - SDA Processor with Poor Routing 15](https://d.docs.live.net/c07c9239cf8ef362/4th%20Year%20Sem2/CC3501/Github/CT-Monitoring-Assignment/CTM%20-%20Report%20%5e0%20Manual/CT-Metering-Assignment-R_4.docx#_Toc497053467)

[Figure 14 - Finished PCB 16](https://d.docs.live.net/c07c9239cf8ef362/4th%20Year%20Sem2/CC3501/Github/CT-Monitoring-Assignment/CTM%20-%20Report%20%5e0%20Manual/CT-Metering-Assignment-R_4.docx#_Toc497053468)

[Figure 15 - Definition of Timing in Circuit (seconds) 17](https://d.docs.live.net/c07c9239cf8ef362/4th%20Year%20Sem2/CC3501/Github/CT-Monitoring-Assignment/CTM%20-%20Report%20%5e0%20Manual/CT-Metering-Assignment-R_4.docx#_Toc497053469)

[Figure 16 - Shifted Waveform from CT to ADC Input 18](https://d.docs.live.net/c07c9239cf8ef362/4th%20Year%20Sem2/CC3501/Github/CT-Monitoring-Assignment/CTM%20-%20Report%20%5e0%20Manual/CT-Metering-Assignment-R_4.docx#_Toc497053470)

[Figure 17 - Mapping Layout of Data 19](https://d.docs.live.net/c07c9239cf8ef362/4th%20Year%20Sem2/CC3501/Github/CT-Monitoring-Assignment/CTM%20-%20Report%20%5e0%20Manual/CT-Metering-Assignment-R_4.docx#_Toc497053471)

[Figure 18 - Frame Details 20](https://d.docs.live.net/c07c9239cf8ef362/4th%20Year%20Sem2/CC3501/Github/CT-Monitoring-Assignment/CTM%20-%20Report%20%5e0%20Manual/CT-Metering-Assignment-R_4.docx#_Toc497053472)

[Figure 19 - Xbee API Packet 20](https://d.docs.live.net/c07c9239cf8ef362/4th%20Year%20Sem2/CC3501/Github/CT-Monitoring-Assignment/CTM%20-%20Report%20%5e0%20Manual/CT-Metering-Assignment-R_4.docx#_Toc497053473)

[Figure 20 - Creation of Frame Packet 20](https://d.docs.live.net/c07c9239cf8ef362/4th%20Year%20Sem2/CC3501/Github/CT-Monitoring-Assignment/CTM%20-%20Report%20%5e0%20Manual/CT-Metering-Assignment-R_4.docx#_Toc497053474)

[Figure 21 - Checksum and AS Transmission 21](https://d.docs.live.net/c07c9239cf8ef362/4th%20Year%20Sem2/CC3501/Github/CT-Monitoring-Assignment/CTM%20-%20Report%20%5e0%20Manual/CT-Metering-Assignment-R_4.docx#_Toc497053475)

[Figure 22 – Receiving Code from Zigbee 22](https://d.docs.live.net/c07c9239cf8ef362/4th%20Year%20Sem2/CC3501/Github/CT-Monitoring-Assignment/CTM%20-%20Report%20%5e0%20Manual/CT-Metering-Assignment-R_4.docx#_Toc497053476)

[Figure 23 - Libcurl Code 23](https://d.docs.live.net/c07c9239cf8ef362/4th%20Year%20Sem2/CC3501/Github/CT-Monitoring-Assignment/CTM%20-%20Report%20%5e0%20Manual/CT-Metering-Assignment-R_4.docx#_Toc497053477)

[Figure 24 - Transmission with API Key to ThingSpeak 24](https://d.docs.live.net/c07c9239cf8ef362/4th%20Year%20Sem2/CC3501/Github/CT-Monitoring-Assignment/CTM%20-%20Report%20%5e0%20Manual/CT-Metering-Assignment-R_4.docx#_Toc497053478)

[Figure 25 Hosting Website 25](https://d.docs.live.net/c07c9239cf8ef362/4th%20Year%20Sem2/CC3501/Github/CT-Monitoring-Assignment/CTM%20-%20Report%20%5e0%20Manual/CT-Metering-Assignment-R_4.docx#_Toc497053479)

[Figure 26 - HTML Initial Setup 25](https://d.docs.live.net/c07c9239cf8ef362/4th%20Year%20Sem2/CC3501/Github/CT-Monitoring-Assignment/CTM%20-%20Report%20%5e0%20Manual/CT-Metering-Assignment-R_4.docx#_Toc497053480)

[Figure 27 - Background Style Sheet 26](https://d.docs.live.net/c07c9239cf8ef362/4th%20Year%20Sem2/CC3501/Github/CT-Monitoring-Assignment/CTM%20-%20Report%20%5e0%20Manual/CT-Metering-Assignment-R_4.docx#_Toc497053481)

[Figure 28 - Transferring Files to the Host 26](https://d.docs.live.net/c07c9239cf8ef362/4th%20Year%20Sem2/CC3501/Github/CT-Monitoring-Assignment/CTM%20-%20Report%20%5e0%20Manual/CT-Metering-Assignment-R_4.docx#_Toc497053482)

[Figure 29 - Burnt PCB 27](https://d.docs.live.net/c07c9239cf8ef362/4th%20Year%20Sem2/CC3501/Github/CT-Monitoring-Assignment/CTM%20-%20Report%20%5e0%20Manual/CT-Metering-Assignment-R_4.docx#_Toc497053483)

[Figure 30 - Finished PCB and One CT Connected 27](https://d.docs.live.net/c07c9239cf8ef362/4th%20Year%20Sem2/CC3501/Github/CT-Monitoring-Assignment/CTM%20-%20Report%20%5e0%20Manual/CT-Metering-Assignment-R_4.docx#_Toc497053484)

[Figure 31 - Final Product with Case and One CT Connected 28](https://d.docs.live.net/c07c9239cf8ef362/4th%20Year%20Sem2/CC3501/Github/CT-Monitoring-Assignment/CTM%20-%20Report%20%5e0%20Manual/CT-Metering-Assignment-R_4.docx#_Toc497053485)

[Figure 32 - The Checking Of Data Being Sent By CTM-3501 And Being Displayed On The Pi And Internet Webpage 29](https://d.docs.live.net/c07c9239cf8ef362/4th%20Year%20Sem2/CC3501/Github/CT-Monitoring-Assignment/CTM%20-%20Report%20%5e0%20Manual/CT-Metering-Assignment-R_4.docx#_Toc497053486)

[Figure 33 - The CTM Setup 30](https://d.docs.live.net/c07c9239cf8ef362/4th%20Year%20Sem2/CC3501/Github/CT-Monitoring-Assignment/CTM%20-%20Report%20%5e0%20Manual/CT-Metering-Assignment-R_4.docx#_Toc497053487)

[Figure 34 - Current Draw of Vacuum Cleaner 31](https://d.docs.live.net/c07c9239cf8ef362/4th%20Year%20Sem2/CC3501/Github/CT-Monitoring-Assignment/CTM%20-%20Report%20%5e0%20Manual/CT-Metering-Assignment-R_4.docx#_Toc497053488)

CURRENT TRASFORMER METER

# INTRODUCTION

The aim of this project was to design and implement the CTM-3501. This report outlines the design of the circuit with calculations and diagrams. The implementation in Code Warrior using Free Scale will be discussed and explained. The final product and testing will be evaluated, commented upon, and recommendations are given.

The CTM-3501 (CTM for short) stands for Current Transformer Meter and followed by the model number i.e. CTM-xxxx. The 35xx series are the top of the range on the market. The CTM was initially designed to meet the minimum requirements of having an embedded system with a sensor which acquires data. The features to be implemented were given flexible conditions, while the team behind the CTM were driven to achieve a glorious result.

A current transformer (CT) is and electrical device used to step or scale current down to a safe working current to be measured by meters and relay for protection and monitoring. The CT gives a proportional current in its secondary winding to the current in the primary winding based on the turns ratio, and should have approximately a negligible load. Owing to the construction of CTs, they provide electrical isolation from the high voltage circuit through the windings, consequently increasing safety. Generally, these transformers have a small number of primary turns and a larger number of secondary turns which defines the turns ratio [1]. Figure 1 shows a VT comprising of the primary and secondary windings, and earthed secondary winding.

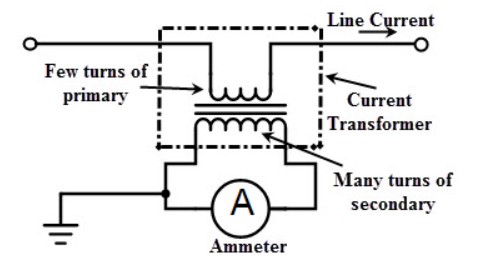


Figure 1 - Current Transformer [1]

There are several types of CTs including wound type, bar type and window type transformers. The CT behaves similar to a normal transformer in no load conditions (small burden) and therefore the secondary normally has low currents and is earthed for safety precautions. The current on the secondary winding is governed by the equation:

Where is the primary winding current, is the secondary winding current, and are the number of turns for the primary and secondary windings respectively. If the secondary winding is not shorted (by a burden resistor) than high dangerous voltages can occur governed by equation:

To demonstrate this effect, we assume a and are 1 and 100 with a primary voltage of 230. Substituting into the equation:

Therefore, a CT with a turns ratio of 100:1 can develop 23 kV with nominal phase voltage. Most insulation is rated to 1000 V and this high voltage can degrade the insulation creating hazards. Current transformers are used in many aspects of a power system. They are utilized in power distribution, generating stations, substations and at domestic, commercial and industrial levels [1].

The essence of the design was to monitor an alternating current using a current transformer (CT). The CT’s were to be part of a sensor node which sent information via wireless communication to a base station. The base station was a Raspberry Pi which could forward near to real-time data upload to the internet. The complete design process of the CTM is explained in this document and a corresponding user manual is attached.

# OVERVIEW OF DESIGN

The fundamental operation of the CTM is displayed in Figure 2 and depicts the start of the data acquisition from the current transformers. The voltage is shifted using the wave shaping and converted to a digital signal. This signal is then computed to get a desired output based on the calibration phase and type of current transformer being used. The data is then sent over radio frequency to a base station Raspberry Pi and uploaded via WiPi to ThingSpeak for webpage display.

Figure 2 -Overview of Operation

# HARDWARE DESIGN

## Schematic

The CTM was based on the embedded system design of the Kinetis FRDM K20DX128M5 board for the processing circuit. The schematic was compartmentalized into several parts during the design process and each part is discussed as per its design in the following sections.

### Power Supply

The two processors each required 3.3 V power supply, so during the design this was the primary reason for selecting 3.3 V. The power design is shown in Figure 3.

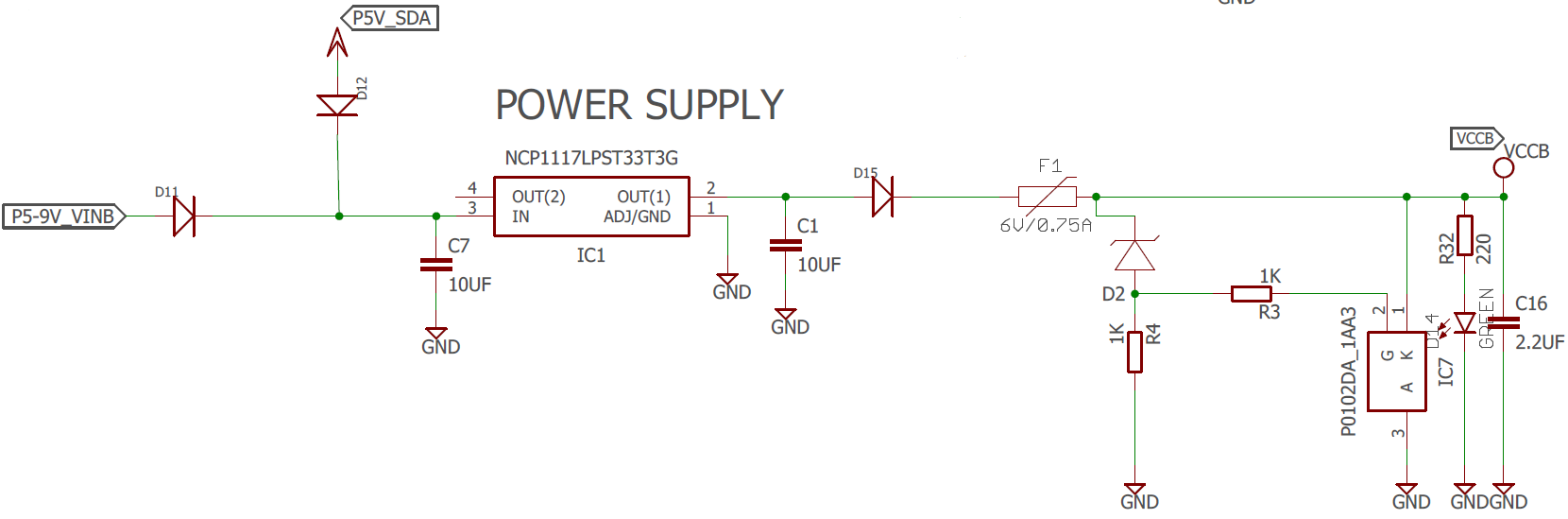


Figure 3 - Power Supply and Corresponding Protection

The supply to the circuit was from a 9 V coin-cell battery or from a 5 V USB with each passing through the diodes D11 and D12 to prevent incorrect connection. Each supply was fed into a power supply with the capacitor sizes selected from the data sheet for the NCP1117LPST33TCG. It was noted that at 3.3 V the regulator could output current in excess of 1.0A. The poly fuse F1 was selected at 0.75 A as this would protect the regulator and was above the calculated maximum current draw of the circuit (See Battery Sizing & Maximum Current). The SCR and D2 and resistor provide Crowbar protection in the event of voltage spikes. D2 was selected at 3.5 V to prevent transients entering the processors and causing damage. The poly fuse provides over-current protection and the crowbar protection provides over-voltage protection.

### Operational Input Shifters

The CTs transduce the current into a voltage which is fed into the LM358 op-amp. The output voltage is level shifted from the input bias voltage on the non-inverting pin from the voltage division over the potentiometer VPOT as shown in Figure 4.

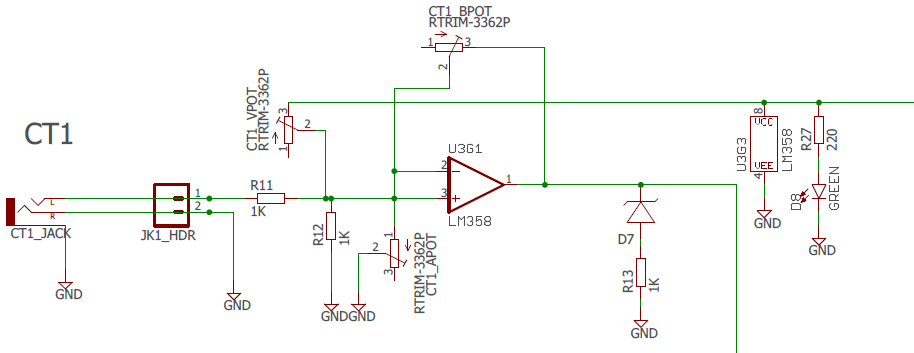


Figure 4 - Operational Amplifiers and Input Shifting Circuit

Figure 4 shows the output is clipped by the diode D7 which was set at 3.3 V. If the output voltage was above this value then it was clipped to earth protecting the input pins to the processor against transients. The original values for the POTs were calculated but it was found that each need to be adjusted due to manufacturer tolerances. The wave form need to be shifted positively half the maximum input voltage of 3.3 V. This meant 1.65 V input bias but this was affected during prototyping due to power supply voltages. Note the headers which were installed near the headphone jack for testing purposes.

### Multiplexer

The multiplexer (MUX) was incorporated to reduce the power consumed by the op-amps and POTs. The MUX was supplied power by the microprocessor and switched between the op-amp circuits. This allowed the op-amp circuits to be only powered while data was being recorded. The MUX is shown in Figure 5 and the timing is shown in Figure 15.

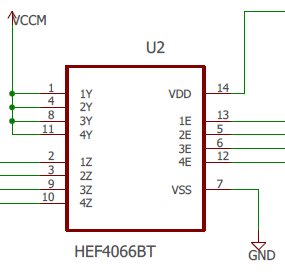


Figure 5 - Multiplexer

The MUX is powered during the cycle (see Figure 15) and switches between CT1, CT2, CT3 and CT4. This allows only one CT circuit to be powered at a time reducing the power consumed by four. The upshot of switching from the microprocessor is that the MUX is only power during the cycle which reduces the amount of power consumed by the MUX by 15.

### ZigBee

The Xbee module brand a Zigbee compliant radios made by Digi International wwas chosen for the wireless communication due to its low power, low data rate which is ideal for battery applications. The Zigbee communication standard supports mesh networks and is used in applications with sensor networks which require machine-to-machine communications.

The transmitter used was the ZigBee. This was selected due to familiarity and the ability to place the device into sleep mode by switching the sleep pin from the microprocessor and is shown in Figure 6. This meant that only once a phase the ZigBee had to be powered which decreased the power consumption.

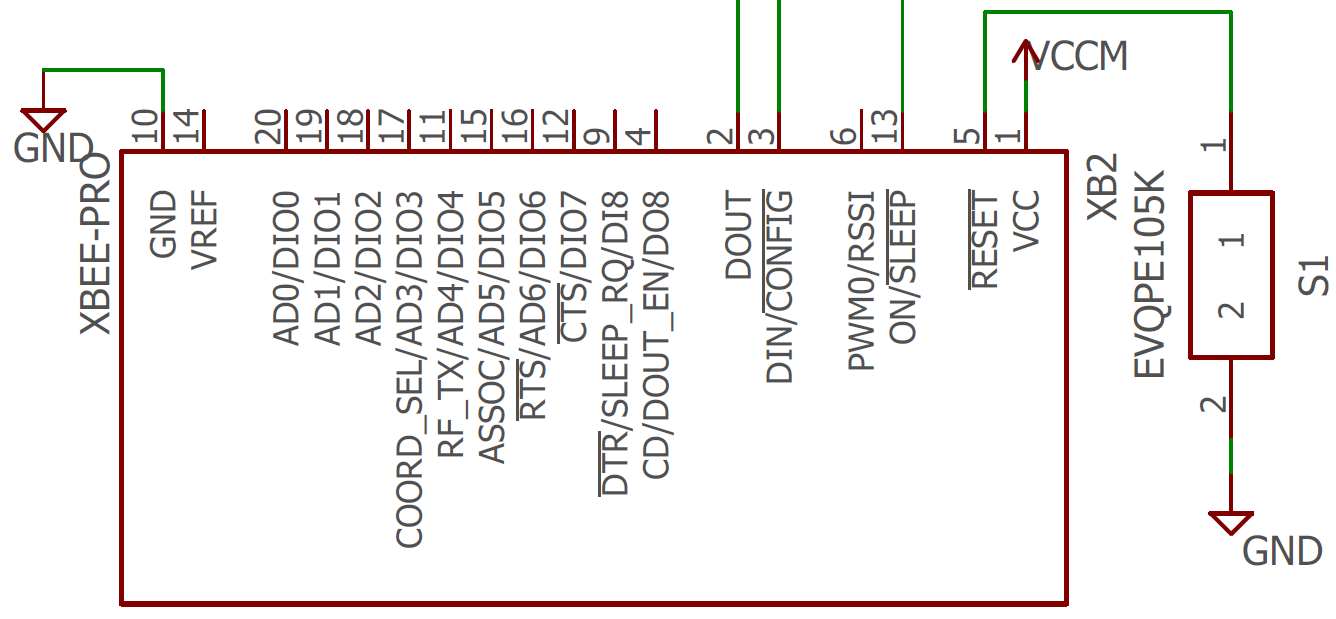


Figure 6 - ZigBee Transmitting Device

### M5 & H5 Processors

There were two microcontrollers used in the design: the MK20DX128VLH5 and the MK20DX128VFM5 which will be referred to here after as the Primary and the SDA processors. As mentioned previously these two processors were selected as they were based off the FRDM board and are shown in Figure 7. These two processors required three oscillators shown in Figure 7 at frequencies of 32 kHz and 8 MHz.

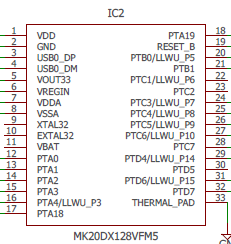
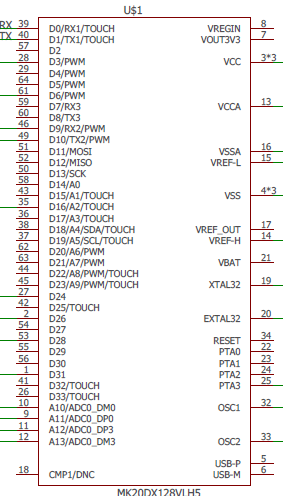
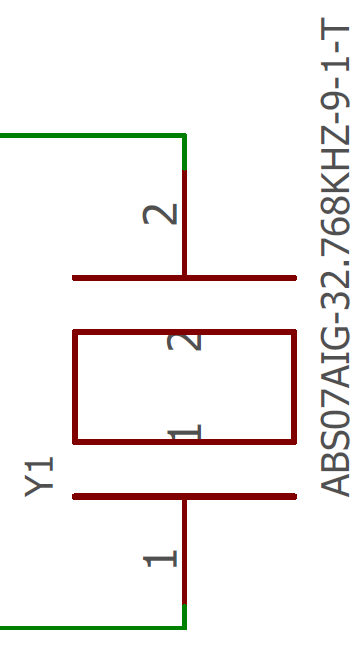
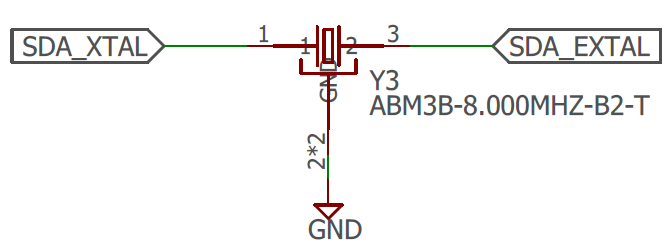


Figure 7 - Primary (left) and SDA (right) Processors and Oscillators (bottom right)



### Mini-USB

The mini USB (Figure 8) was selected to allow connection to the SDA processor to convert the signals to serial to program the primary processor. The BZT52C15S Zener (D1) was installed to protect the board from high input transients and static. The L1 and L2 inductors or ferrite beads are installed to further decrease these effects.

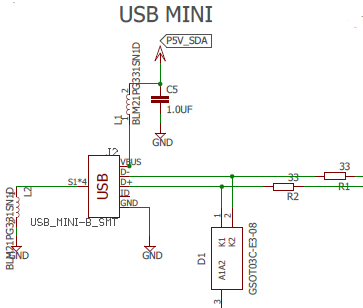


Figure 8 - Mini USB Circuit

### JTAG and SDA Headers

The JTAG and SDA headers were installed to allow serial communication directly to both processors and are shown in Figure 9. The SDA header allowed communication to the primary processor and the JTAG header allowed communication to the SDA processor.

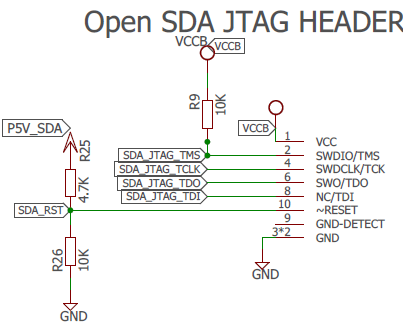
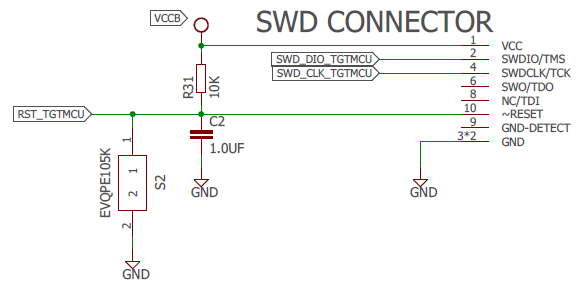


Figure 9 – SDA (left) and JTAG (right) Headers

### 3-Stage Contingency

During the development of the CTM system it was decided to include stages on contingency due to the frontiers which were being pushed.

* Stage 1 Contingency was to essentially build the board in reference to headers which could be connected to a FRDM board. This meant the CTM could be inserted into a FRDM board and operate. This was achieved by connecting to FRDM headers (see Figure 11) all the input and outputs from the CTs. The CT circuits would receive power from the FRDM board and this supply rails was called the Backup (VCCB)

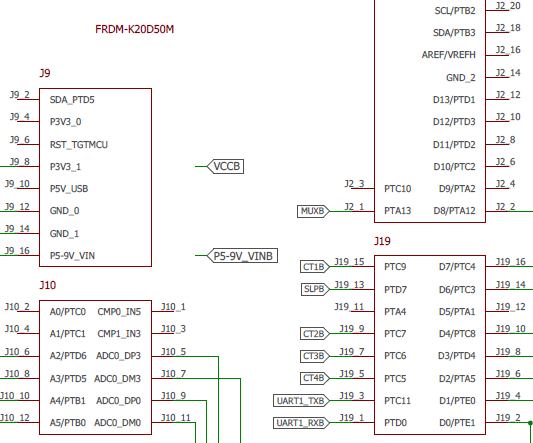


Figure 10 – FRDM Board Headers

* Stage 2 Contingency was having the primary processor drive the circuit connected across the FRDM headers by bridges. This circuit was powered by the CTM power supply and labelled the Main power supply (VCCM).
* Stage 3 Contingency was including the SDA chip and allowing the CTM to be programmed by mini USB interface.

### Battery Selection

The battery was isolated from the board by a 2-position dip switch which disconnected the positive and negative of the battery.

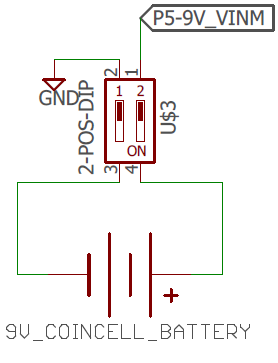


Figure 11 - Coin Cell Battery and Isolator

All electronic systems require power supplies to operate, Power management broadly refers to the generation and control of regulated voltages required to operate an electronic system. It encompasses much more than just power supply design. Today's systems require power supply design be integrated with the system design to maintain high efficiency

Integrated circuit components such as switching regulators, linear regulators, switched capacitor voltage converters, and voltage references are typical elements of power management.

The battery life of the circuit is determined by the battery capacity, which is the amount of current in total that the battery can supply during the lifetime of the battery and the load current of the circuit, the amount of current being consumed by the circuit. The first factor can be found by the manufacturer details of the battery which gives the capacity in amp-hours or milliamp- hours, the second factor requires to calculate the current drawn from the active devices in the circuit. The following formula is used to determine the battery life.

(1)

The Energizer 522 Alkaline 9-volt battery capacity is rated at 230 milliamp-hours at a 1000 milliamp discharge rate and was initially selected for its footprint but was pending the battery sizing calculations.

## Power Optimisation

### Operational Amplifiers

To determine the power draw of the op amp the quiescent current was obtained from the data sheet and the following formula used:

The data sheet provided a typical value and a maximum value for the quiescent current, each value was used to obtain the power of the op amp with the max value used as a worst-case scenario.

per Op-Amp

per Op-Amp

The component is a dual Op-Amp package therefore each package will draw and the board has 4 in total resulting in

### Voltage Regulator

The maximum package power dissipation of the NCP1117LP voltage regulator is given by the formula:

* is the maximum junction temperature range
* is the operating ambient temperature range
* is the thermal Resistance, Junction−to−Ambient

The maximum power :

The maximum current output of the regulator is where V is the voltage drop of the regulator.

To calculate a typical power rating of the voltage regulator it is somewhat hidden within the datasheet. It can be calculated by looking at the following related specifications.

* is the operating junction temperature range
* is the thermal resistance junction-to-case
* is the thermal resistance, junction−to−ambient

specifies how hot the “junction”, the active part of the regulator can get before it goes into thermal shutdown. specifies how much temperature difference to be expected between the junction and the outside of the package. This is relevant if you can quickly remove heat from the package. A perfect coupled heat sink hooked to your package, for each Watt the junction temperature would rise only above the temp of the heat sink. is how hot the junction gets when the regulator is dissipating a given amount of power and the regulator is sitting at a given ambient temperature. We design our regulator to work under modest commercial conditions such that it will not exceed and the junction temperature needs to stay below then the maximum temperature rise allowable is The power dissipation is given by:

The current is :

### XBee Pro

The XBee Pro attached to the board will draw current during transmission and when in its sleep mode. The maximum current draw will be transmission mode with the data sheet quoting a transmit current of The power down current of the XBee is

### MK20DX128VLH5 & MK20DX128VFM5 Processors

The absolute maximum ratings for the MK20DX128VFM5 and MK20DX128VLH5 are obtained from the device datasheet. The maximum power supply current , includes all current being sourced by the microcontroller pins in addition to the current used to operate the CPU and peripherals. For the MK20DX128VLH5 and MK20DX128VFM5 the

### SN74LVC125A Quadruple Bus Buffer Gate

The absolute maximum current draw of the SN74LVC125A was calculated from the manufacturers datasheet by adding the continuous input current and the output current.

### HEF4066B Quad Single-Pole Single-Throw Analog Switch(Multiplexer)

The max supply current at worst case scenario for the Multiplexer was when the device is operating at an ambient temperature of .

### Mini USB

The specifications of USB 2.0 states that the maximum current draw is .

### LED’s

The current drawn from the LED’s can be found using the formula:

* Supply voltage
* LED forward voltage drop (found in datasheet)
* Resistor value

The board has a total of 6 LED’s taken the worst case that all LED’s are on at the same time the total current draw

### Battery Sizing

Several factors affect the battery life of the design and with the primary considerations including: devices operating, sizes, and duty cycle. Table 1 lists the primary power consuming components and their maximum “worst case” current draw. These values are a worst-case scenario in which the absolute maximum current would be supplied to each component.

Table 1 - Current Draw of Design

|  |  |
| --- | --- |
| Component | Maximum Current |
| Op Amps |  |
| Voltage Regulator |  |
| XBee Pro |  |
| MK20DX128VLH5 |  |
| MK20DX128VFM5 |  |
| NAND Gates |  |
| Multiplexer |  |
| LED’s |  |
| Mini USB |  |
|  |  |

The battery life calculation will be done using the worst-case scenario for each component, assuming that all components are at maximum draw at the same time and operating all the time. The total is , which is approximately 1000mA.

Using the battery life equation (1):

The worst-case scenario indicates that the board could stand alone power itself for approximately 14 minutes. Therefore, power saving measures were undertaken.

### Power-Saving Strategies

To save power the following designs were implemented:

* Reduction in clock speed.
* Switched power to Multiplexor.
* Switched power to CT channels.
* If a low value was read, then turned off a CT channel off for 30 minutes.
* Sleep mode utilised with Zigbee.
* Waiting for interrupts and putting the CPU into low power mode.

## PCB Design

The selected method of layout approach for the PCB was trial and error. This approach allowed many designs to be made, consuming copious amounts and time, and delivering maximum returns in the form of stress. The size of the board was limited to 80x100 mm as this is the maximum allowable board size in the free version of eagle. This proved challenging in the creation of the PCB.

All the components were placed on the board and it was auto-routed to give an indication of placement pattern. The result yielded 84.2 % routed (see Figure 12), which was not high and indicated poor placement. It was decided to place all the components and route manually.

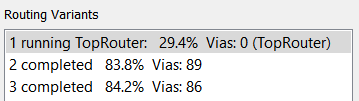


Figure 12 - Auto Routed Test of Placement

All of the components were successfully placed but the SDA processor which had 32 pins and proved difficult routing which originally could not be routed as indicated by the three air wires in Figure 13.

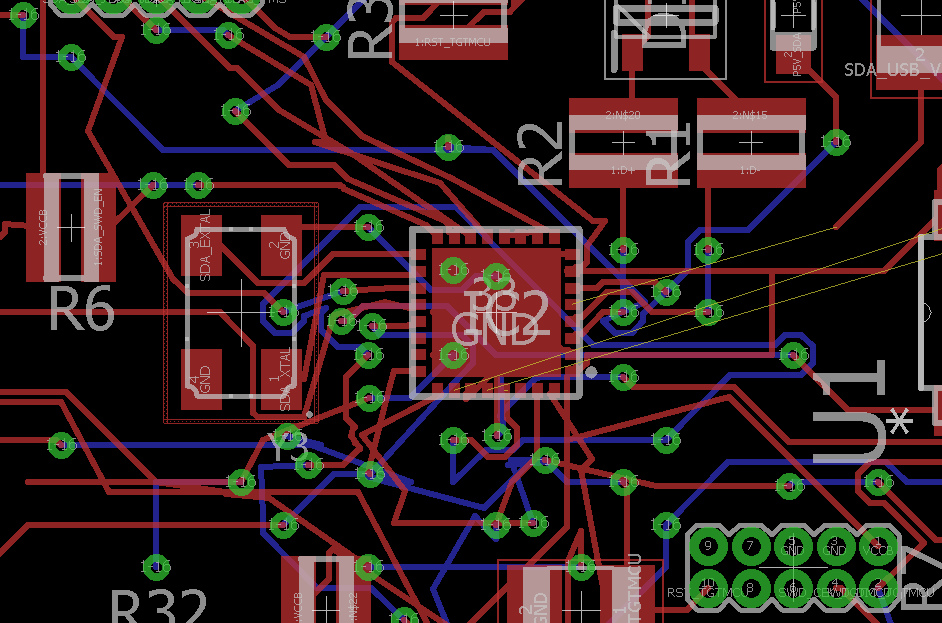


Figure 13 - SDA Processor with Poor Routing

Routing manually allowed for the optimal placement of the components, this included:

1. Placing all the CT circuits in their individual circuits off to the side
2. Increasing the free space around each processor for routing
3. Allow room between the FRDM headers for routing around the outside
4. Placing the power supply together in a corner
5. Relocating the USB outside of the FRDM Headers

The finished PCB is shown in Figure 14 and the areas of noted above are shown with their corresponding letters.

B

E

B

A

D

C

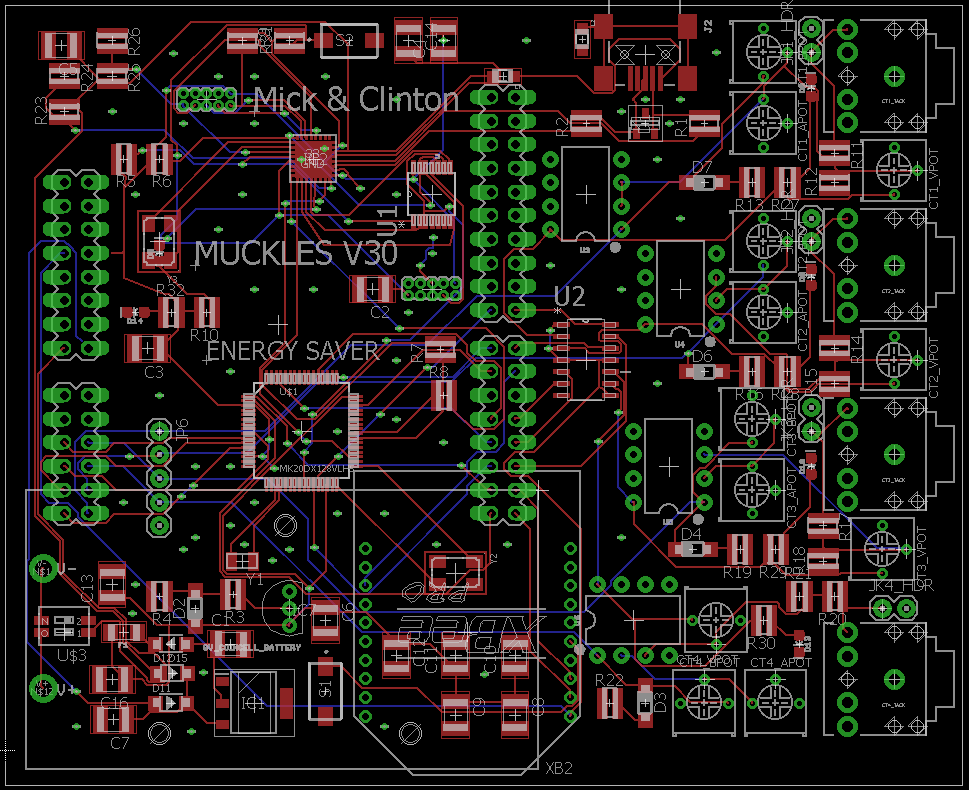


Figure 14 - Finished PCB

# SOFTWARE DESIGN

## Kinetis Code

### Analog to Digital Converter

The Kinetis code was developed by adding on each task successively. The first step was to setup the project and get it receiving data from the channels. The pins for the Analog to Digital Converter (ADC) channels were setup when designing the schematic. The advantage of using FRDM-K20D50M processor was the familiarity with the syntax of C and the onboard ADC.

### Timing

Figure 15 shows the operation of the circuit with respect to timing and defines a phase, cycle and period (note the units are seconds).

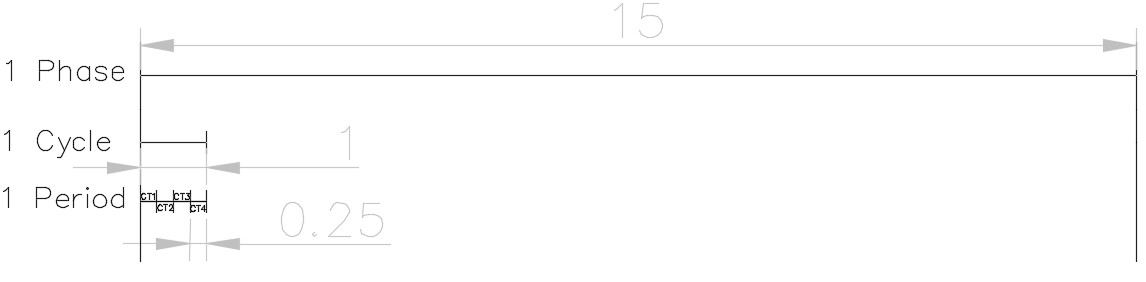


Figure 15 - Definition of Timing in Circuit (seconds)

Figure 15 shows there were three loops timing sessions occurring; a phase, a cycle, and a period. The program was sent into a wait for interrupt phase on startup while enabling Timer 1 which was a 15 second timer that timed a phase. This would interrupt and disable itself starting Timer 2. Timer 2 would interrupt itself every 1ms and on each interrupt a piece of data would be captured. After 250 of these data acquisitions a switch statement would turn on another channel until all channels were had data recorded. The switch statement would turn on a single channel at a time. A minimum value was set up that if a channel did not read above a (xx) minimum value then it was disabled for 40 phases which was 10 minutes.

### Data Percentage

The values to for each channel were sent to a putty to verify when testing and display the state of each channel. This was important during the calibration phase and a clipping range was introduced. This clipping function occurred if the input ranges of the ADC were above 3.300 V which translated to 16-bit number of 65,535- 0, while a buffer of 100 was placed on each side shifting it to 65,435-100. It was decided that it was easier to map this to a percentage which could be calibrated to the type of sensor (10A, 20A or 25A) during setup. This removed the necessary gain adjustment.

### Data Calculation & Mapping

The dual voltage from the input CT’s was shifted to a positive waveform using an operational amplifier circuit and the effect on the waveform is shown Figure 16.

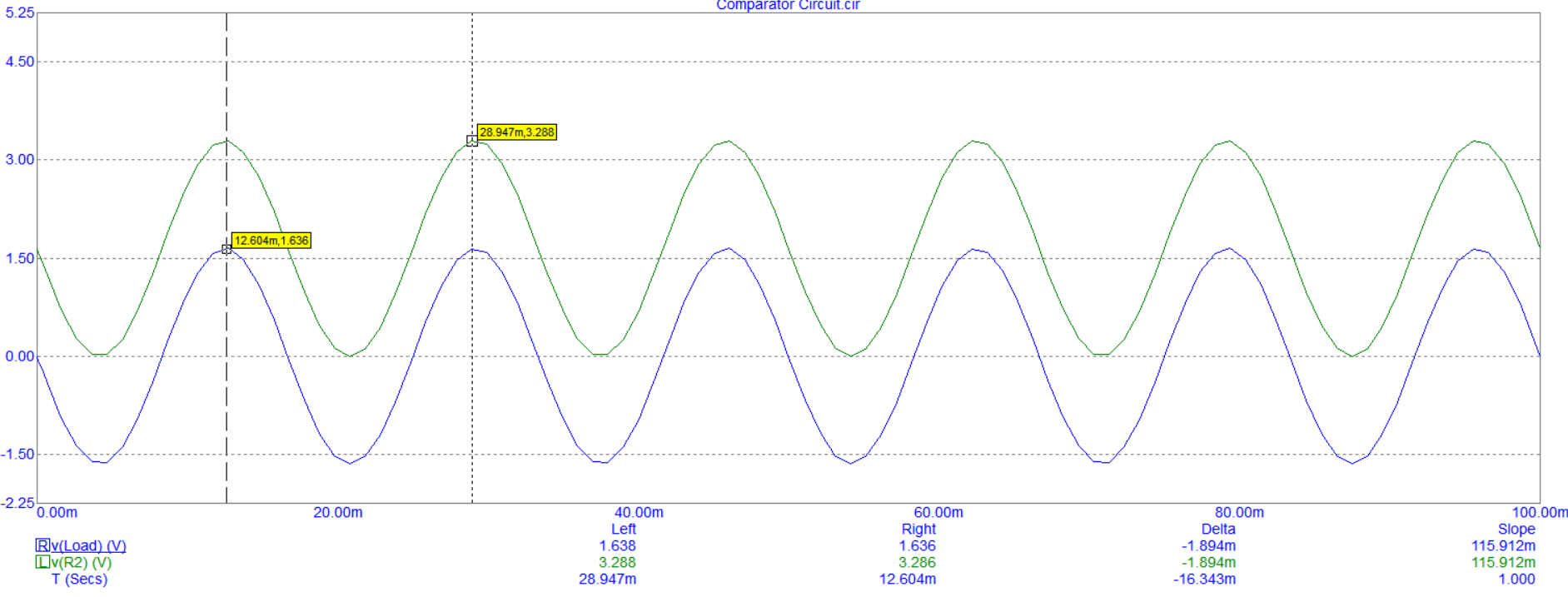


Figure 16 - Shifted Waveform from CT to ADC Input

Data was collected at each millisecond for 250 samples and stored into an array. The centre of each sample was calculated by basic averaging. The Root Mean Square (RMS) of voltage in an alternating current (AC) circuit is defined as a common mathematical method of finding the effective voltage compared to a direct current (DC) system. The voltage of the system can be used to determine the RMS as found in equation (2).

(2)

The voltage which will be fed into the ADC will be between 3.3V and 0V, which is equivalent to using the equation (3).

(3)

The mapping of the data went through four distinct stages and shown in Figure 16. It depicts the peak-to-peak voltage from the CT, being shifted to a positive voltage, being converted to a 16-bit number and the internal mapping to corresponding amperage.

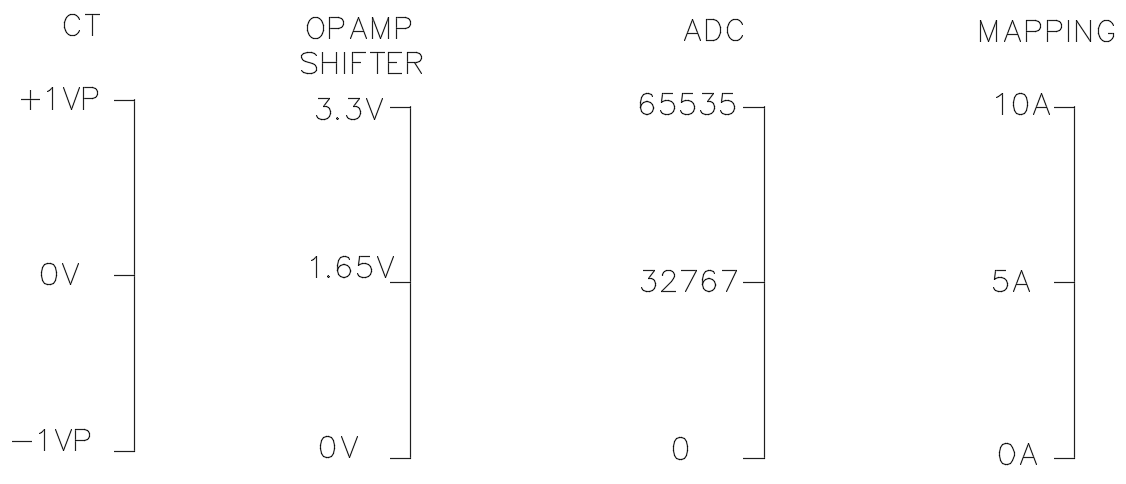


Figure 17 - Mapping Layout of Data

### ZigBee Transmission

The transmission of the data over the Xbee module will be done by placing the data into array and using the Asynchroserial function to send one character at a time. The code shows the current sensor values placed into the buffer named message with the “*strlen"* function assigning the message size which is important before the API structures can be constructed.

**static** **char** message [100];

**snprintf**(message,100,"%f,%f,%f,%f\n",CT\_Current[0],CT\_Current[1],CT\_Current[2],CT\_Current[3]);

**int** message\_size = **strlen**(message);

Xbee’s provide a mode called Application Programming Interface (API) which provides uses with a structured interface where data is communicated the serial interface in organised packets in a determined order. Data transmitted in the form of API packets or data frames have a very well define structure and understanding this structure is crucial to derive data from the frame.

Table 2 - API Fame Structure

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Start Delimiter | Length | | Frame Data | | | | | | | | Checksum |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | … | n | n+1 |
| 0x7E | MSB | LSB | API-specific structure | | | | | | | | Single byte |

An API frame has the structure as shown in Table 1. The specific structures can be found by connecting the Xbee to the XCTU program whilst the specific data is being sent. The picture below shows the API structures for the CT Monitoring Assignment.

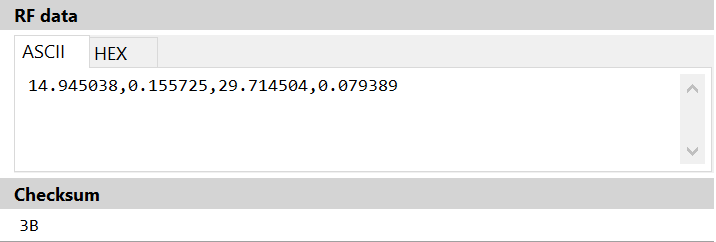


Figure 18 - Frame Details

This information is used to write the code in Kinetis to construct the Xbee API packet as seen in Figure 19.

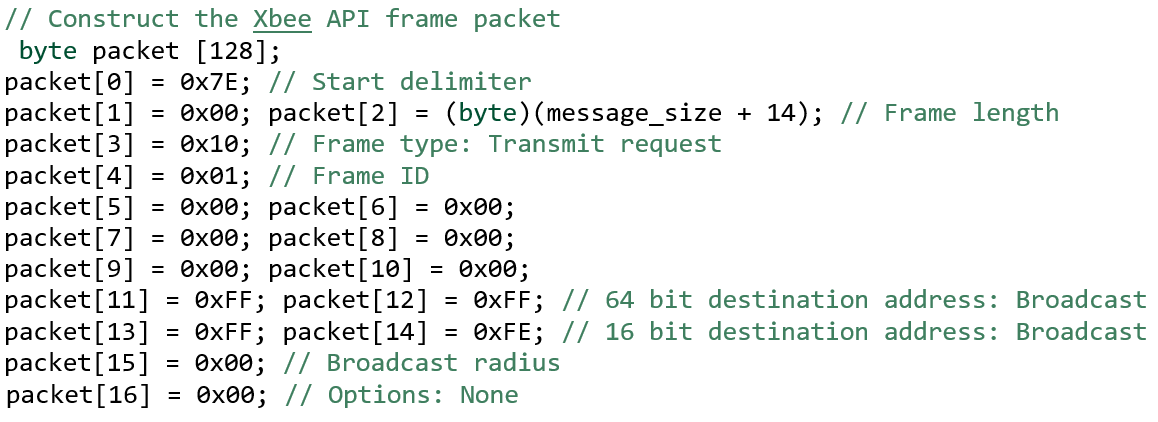


Figure 19 - Xbee API Packet

The message can then be placed into the frame packet once the API structures are established as depicted in Figure 20.

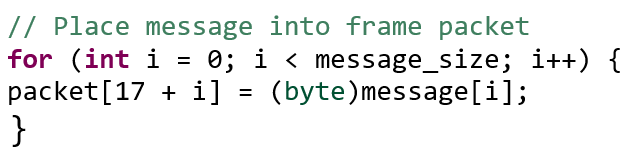


Figure 20 - Creation of Frame Packet

The Checksum is the last byte of the frame and helps test the data integrity. To calculate the checksum of an API frame all the bytes of the packet excluding the start delimiter and length are added. The lowest 8 bits only are kept from this result and this quantity is subtracted from 0xFF. If the checksum is incorrect frames sent through the serial interface will never be processed. The Checksum calculation and the data being sent using the asynchroserial function is shown in Figure 21.

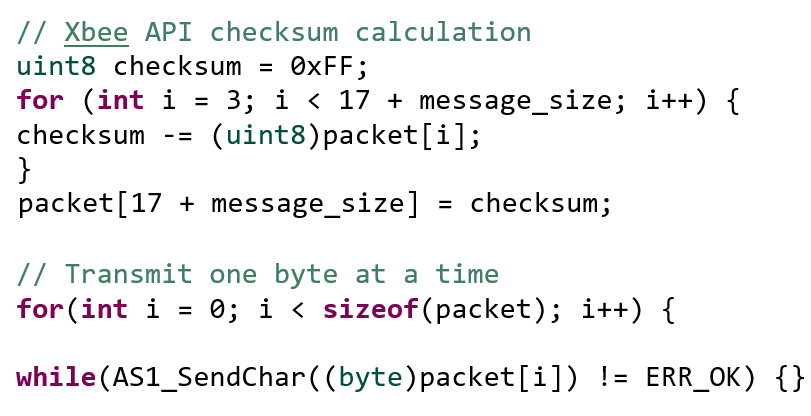


Figure 21 - Checksum and AS Transmission

### Calibration

The system was calibrated by using known currents and developing a scaling factor. Several factors affected the CT% such as supply voltage, POT adjustment, and current transformer tolerances. The scaling factor for the CT was 0.1603.

|  |  |  |
| --- | --- | --- |
| Verifed Amps | CT % Avg | Amp/CT%Avg |
| 0.2 | 1.368 | 0.14619883 |
| 0.2 | 1.282 | 0.15600624 |
| 0.5 | 3.338 | 0.149790294 |
| 0.5 | 3.322 | 0.15051174 |
| 8.8 | 48.808 | 0.180298312 |
| 8.8 | 49.044 | 0.179430715 |
| Scaling Factor |  | 0.160372689 |

### Boot Loading

The SDA processor which we were going to use to convert the signals into serial for reading on the main processor had to be boot loaded. This is accomplished by including a button which is held down during connection to the computer. Windows then opens up an explorer window on the thumb drive called “BOOTLOADER”. The new firmware has to be downloaded from pemicro.com/opensda. In our case it was the “MSD-DEBUG-FRDM-K20D50M\_Pemicro\_v118.SDA”.

## C++ Code

### ZigBee Data Acquisition

The receiving of the data packets on the raspberry pi required the use of unions and structures of the C++ programing language. A union in C programming is a user defined variable which may hold members of different sizes and type which all members share the same memory location. A structure is a convenient tool for handling a group of logically related data items. Structure help to organize complex data is a more meaningful way. The following code was implemented to handle the Xbee protocol in Figure 22.

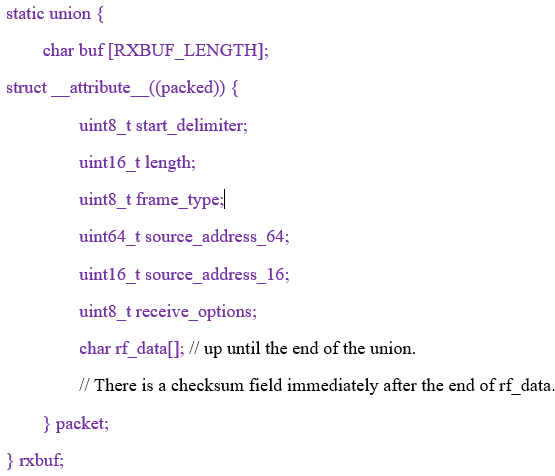


Figure 22 – Receiving Code from Zigbee

The incoming bytes are placed into rxbuf.buf[]. The struct rxbuf.packet allows for named access to particular fields within the binary protocol. The packet receive function was written with several if statements that reads from the serial port into rxbuf until a complete packet has been received. It returns the length of the data payload or -1 if receive failed. Before the loop of the function was run the buffer was zeroed and an index was initiated to 0.

memset(&rxbuf, 0, sizeof(rxbuf));

rxbuf\_idx = 0;

The API frame structure is known, therefore within code there are checks, as we are expecting the start delimiter, the following was written:

*if ((rxbuf\_idx == 0) && (c != 0x7E)) {*

this *checks* the data to see if the first byte is the expected start delimiter. If this does not occur we are not synchronised with the Xbee and we discard bytes by restarting this loop body until we see a start of frame delimiter. If the start delimiter is received then the characters are saved into the buffer. Once rxbuf\_idx is 3, this acknowledges that we have received the length of the packet. The received packet length is in big endian format. Endianness refers to the sequential order in which bytes are arranged into larger numerical values, when stored in computer memory, or when transmitted over digital links.

*rxbuf.packet.length = be16toh(rxbuf.packet.length);*

When rxbuf\_idx is greater or equal to 4 this indicates that the number of bytes received is rxbuf\_idx, this is due to 4 bytes not being counted in length which are the start delimiter, the 2 bytes of length and checksum.

} else if (rxbuf\_idx >= 4) {

The complete packet is received when we have the length plus 4 bytes.

if (rxbuf\_idx >= rxbuf.packet.length+4) {

The function performs a number of checks including the check sum calculation, to confirm it is indeed the required data, the function returns the data by subtracting 12 off the length as there are 12 bytes of header included in packet.length.

return rxbuf.packet.length - 12;

### WiPi Upload

To connect the Pi to internet the Wi-Pi module needed to be configured to the network that we the device was operating through. The network configurations were modified in the command line editor to set up a wi-fi connection on the JCU network. transmitting the logged data is by sending specially crafted HTTP requests. A simple HTTP library is called “libcurl” is installed on the Pi. Libcurl is a free and easy-to-use client-side URL transfer library, it is a computer software project providing a library and command-line tool for transferring data using various protocols.

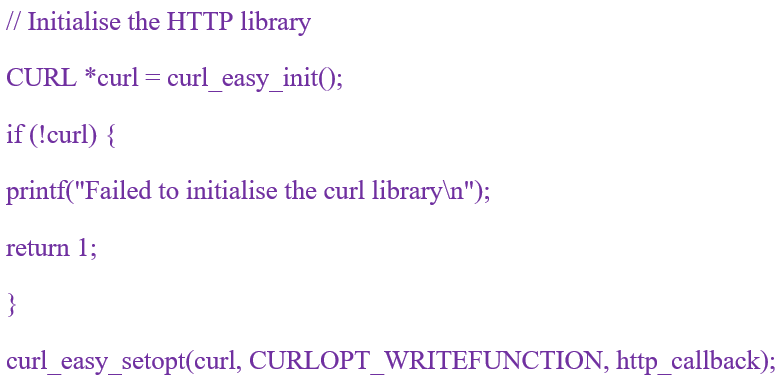


Figure 23 - Libcurl Code

### ThingSpeak Display

The data stream was created by accessing the Thingspeak website, by signing up for a account, and creating a new channel at *thingspeak.com/channels.* Four independent data fields were specified as this corresponded to the four sensors to be read by the board. Once the channels are created the writing API key was found and copied into the code. The writing key is required to upload data to the channel, it provides a URL to use in the code.

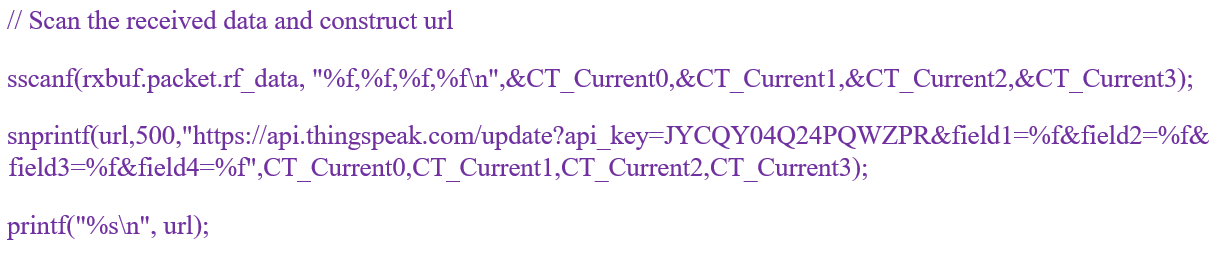


Figure 24 - Transmission with API Key to ThingSpeak

## Webpage Code

### Hosting

The domain name was selected as wwww.ctm-3501.com and was purchased from an online source. The Zuver hosting source is shown in Figure 25 and was where the website hosted.

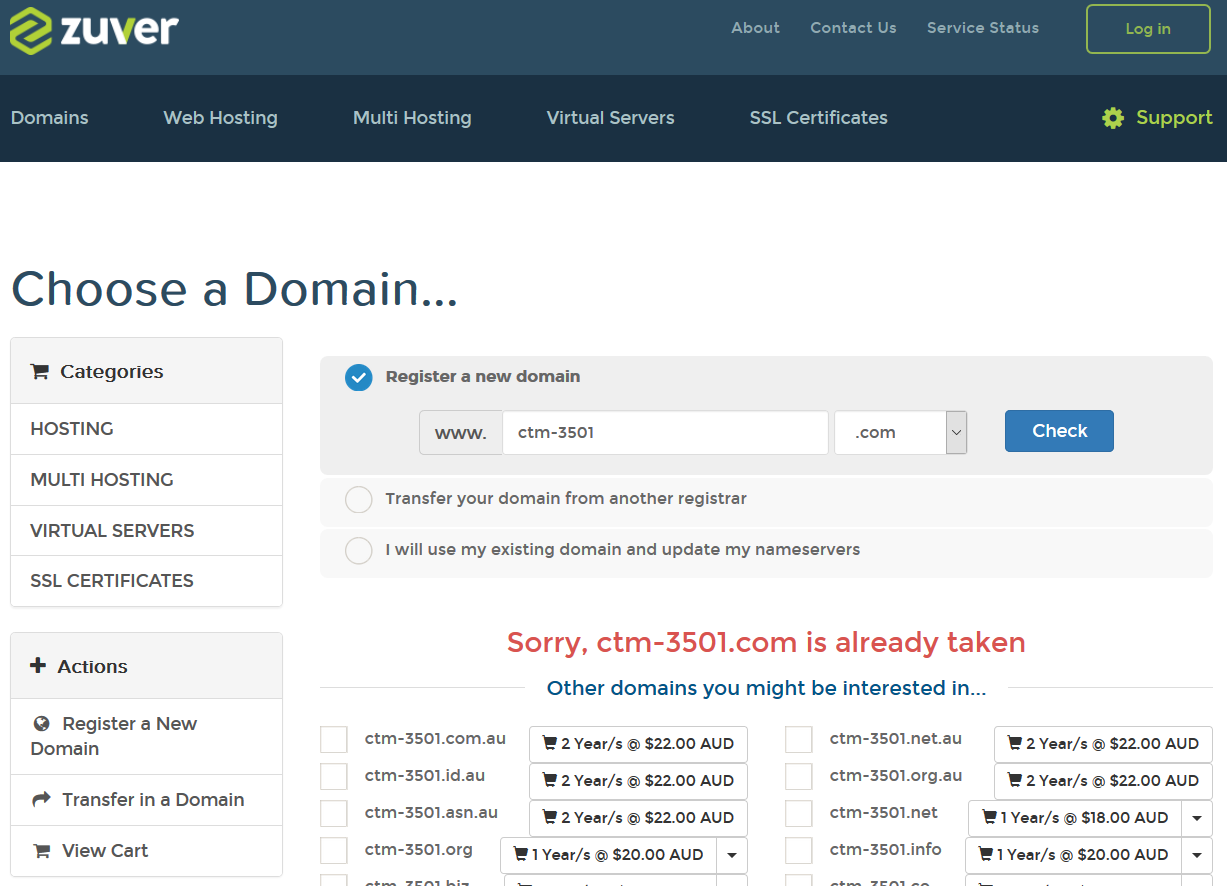


Figure 25 Hosting Website

### Hyper Text Markup Language (HTML)

The webpage template was compiled from online research and downloadable bootstrap templates and the starting code is shown in Figure 25 and in full in the Appendix.

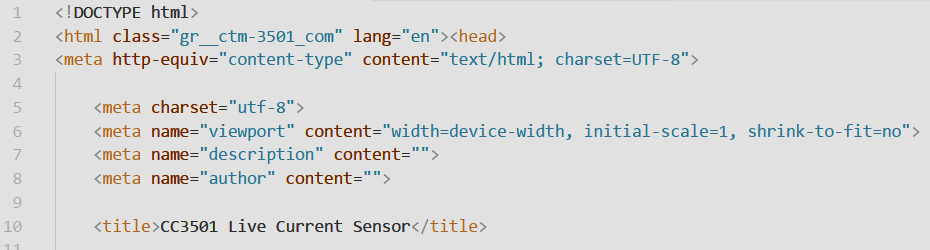


Figure 26 - HTML Initial Setup

### Cascading Style Sheet (CSS)

The fonts for the webpage were created to keep information in the proper display format. The background proved difficult to manage and both the z-index and height were changed as shown in Figure 27.

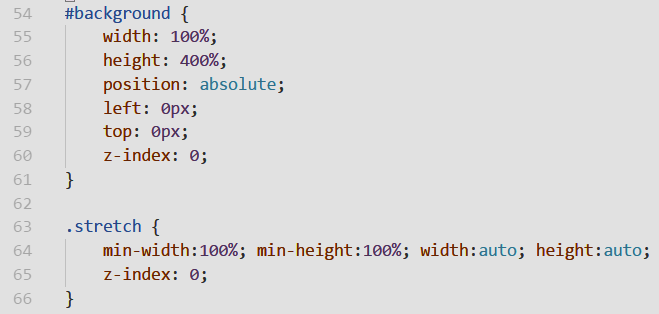


Figure 27 - Background Style Sheet

### File Transfer Protocol (FTP)

FTP is used to transfer files between computers on a network. FTP is used to exchange files between computer accounts, transfer files between an account and a desktop computer, or access online software archives. FileZilla is a powerful and free software for transferring files over the Internet. FileZilla is a very popular FTP client and is used by webmasters from all over the world and was selected to transfer the HTML and CSS to the Zuver host account.

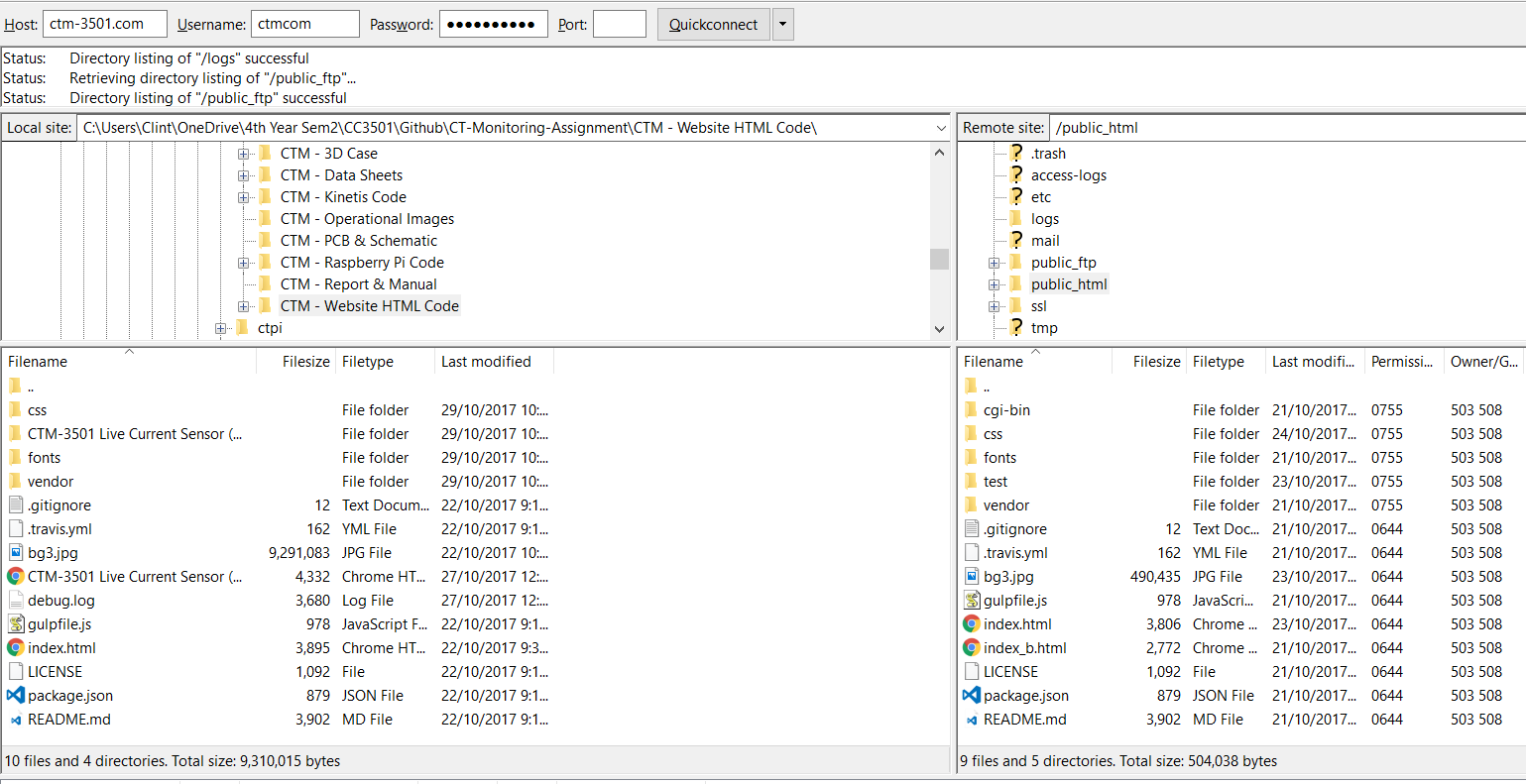


Figure 28 - Transferring Files to the Host

# TESTING & RESULTS

## Real-World Set Back

Due to unforeseen circumstances, the vicissitudes of fate shadowed the project and overheated our PCB while in the oven. The result is shown in Figure 25 and demonstrates that the high temperatures caused the silk screen layer to peel off and the solder paste to boil.

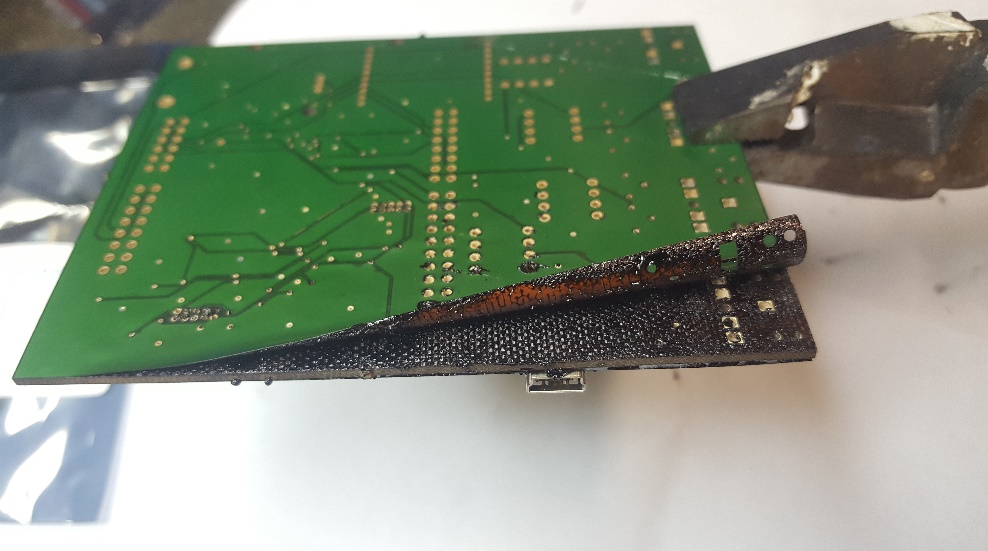


Figure 29 - Burnt PCB

There was another board in the which was printed but the bottom tracks were not. This was circumvented by manual soldering wires where these tracks should have been. This resulted in limiting the amount if CT channels to one, as shown in Figure 25.

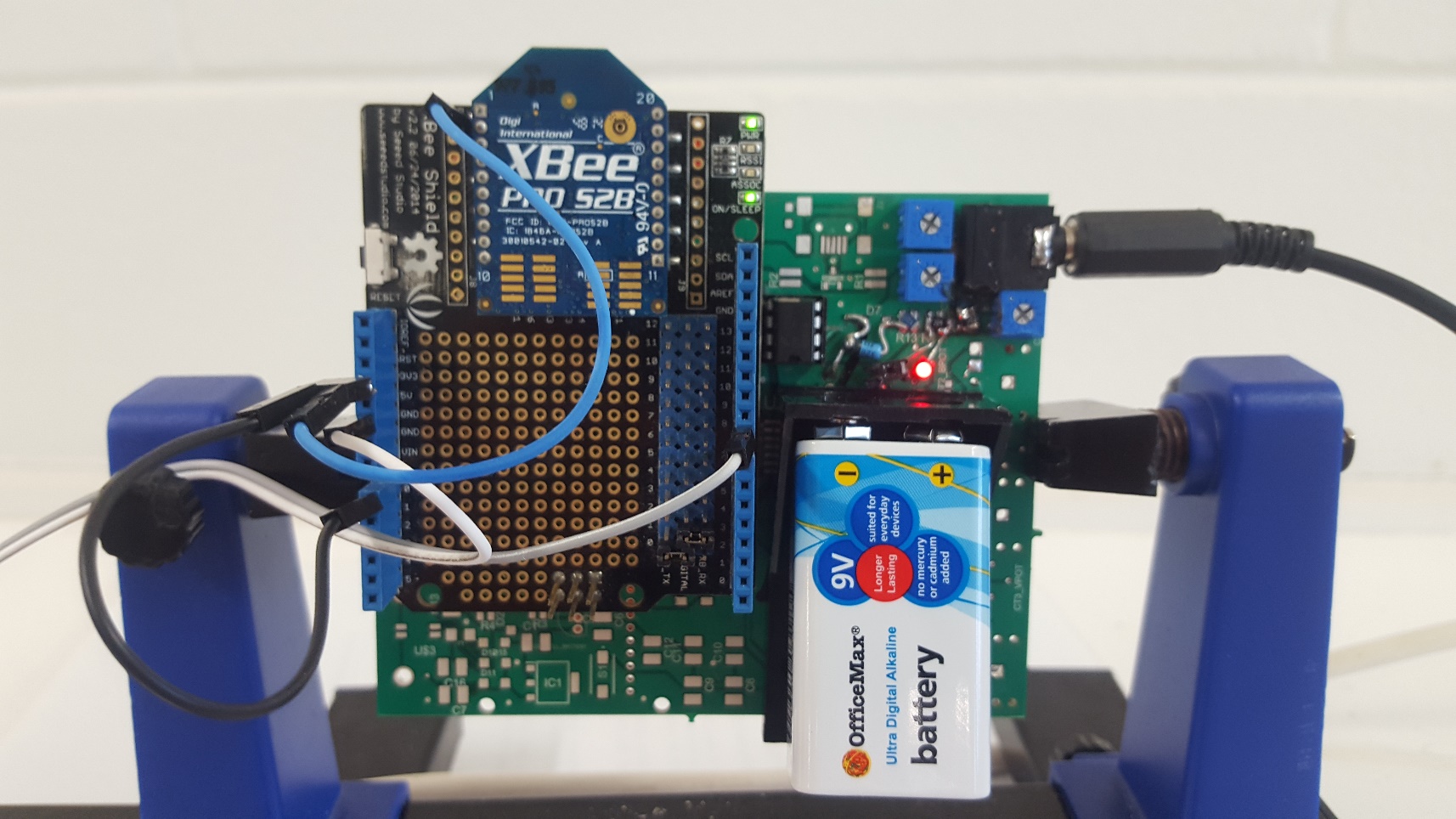


Figure 30 - Finished PCB and One CT Connected

## Final Product

A case was 3D printed for the PCB which allowed connection of the CT circuits, allowed the ZigBee to transmit data, had provision for vents and cooling fan, and finally, LED indication of when data was being transmitted.



Figure 31 - Final Product with Case and One CT Connected

## Webpage Display

The values were checked from the Putty being sent, to reception both on the Pi and webpage.

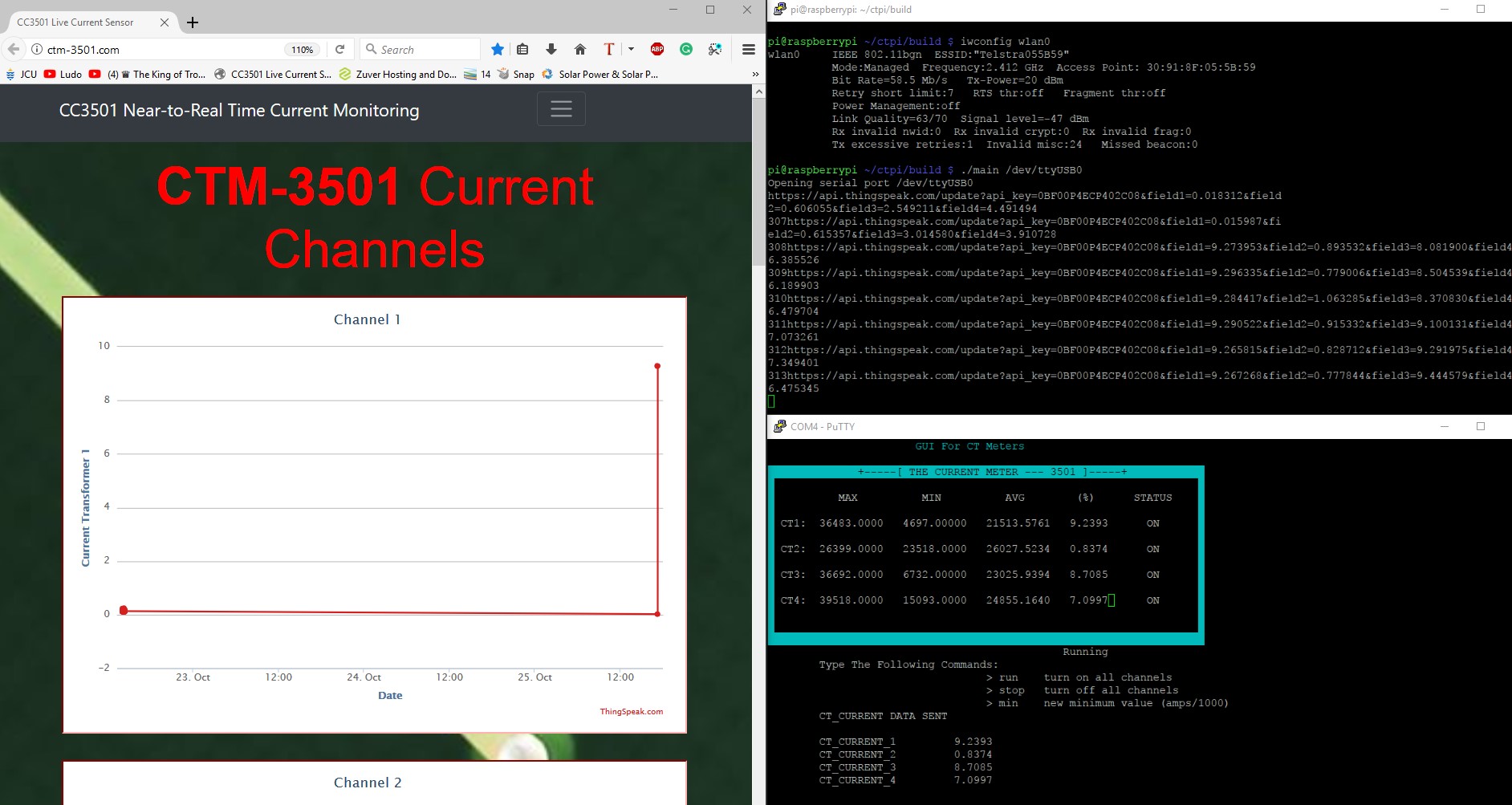


Figure 32 - The Checking Of Data Being Sent By CTM-3501 And Being Displayed On The Pi And Internet Webpage

## Game-Day Performance

The entire circuit including: CT, CTM-3501, ZigBees, Raspbeery Pi, Wi-Pi, and Webpage.

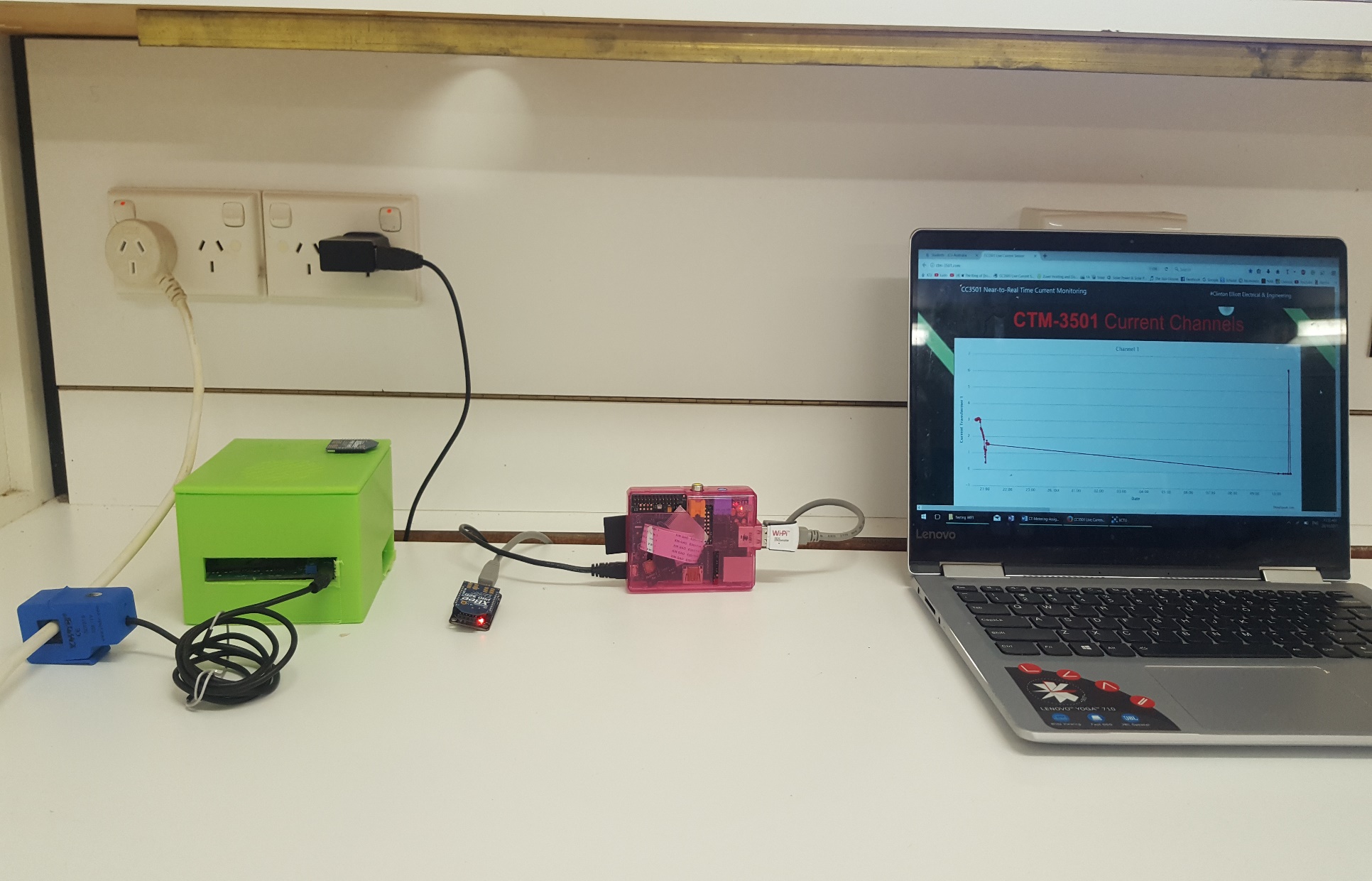


Figure 33 - The CTM Setup

## Results

The range of the CT which was installed was between 0-10A. Therefore, a vacuum cleaner which drew approximately 6A was used. During the session two types of amp meters were used to verify the current draw of the vacuum clean was between 6.2-6.4A. The result from the CTM-3501 is shown in Figure 34.

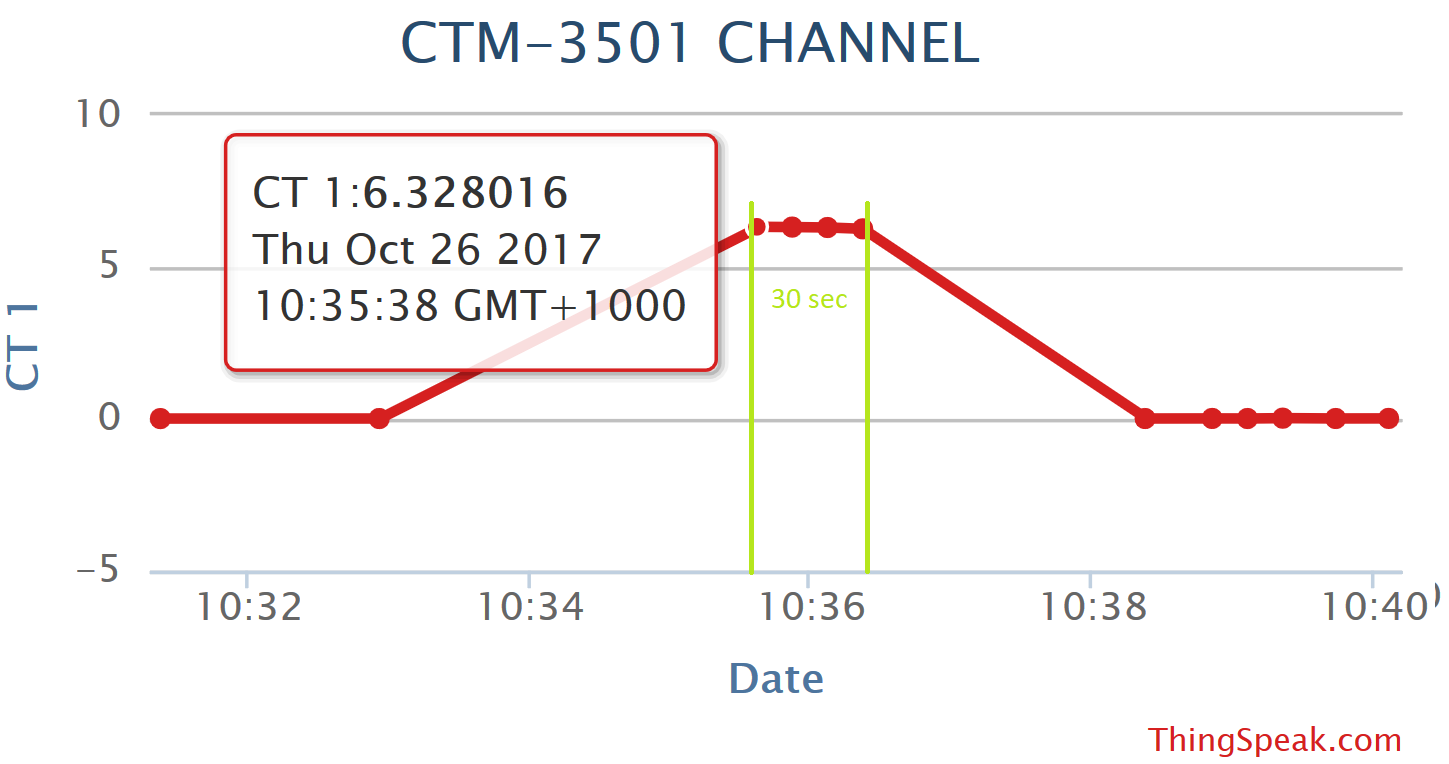


Figure 34 - Current Draw of Vacuum Cleaner

From Figure 34 it is clear that the test lasted approximately 30 seconds and the current draw was consistent during this period. The value recorded was 6.32A and was within the limits of the 6.2-6.4A. This is approximately

# LIMITATIONS & RECOMMENDATIONS

## Limitations

There were several limitations both of the design and process which are discussed below:

* The number of CT channels was limited to four. This could be increased depending which application. Although, this would not be possible to build in Eagle as the board dimensions limited.
* The operation time for the board was calculated at 14 minutes, which is why power-saving strategies were implemented.
* The distance of transmission of the ZigBees was not tested.
* The power dissipation of the CTM-3501 was not verified.
* The entire design could not be tested.
* The time for the PCBs to arrive caused the entire process to be delayed.
* Some parts were not ordered from Element14 and they should have advised us.
* Slow internet speeds affected uploading and caused a lag.
* The Raspberry Pi and Wi-Pi had to be started by exciting a script at the command line.

## Recommendations

There were several recommendations both of the design and process which are discussed below:

* It was noted that switched both the multiplexer and op-amps on at the same time, the data was not consistent, so this was circumvented by having the mux already powered.
* WFI were used with timers in the Kinetis coding, while it would be considerably quicker if RTOS was utilised with MUTEX and Semaphores.
* It is recommended to use a bigger board (>100x60mm) with larger components, e.g. 1210 packages, during the prototyping phase. This makes the entire process easier and in the event of soldering, it is especially helpful. We had to change many components to save space due to board size restrictions.
* The slow internet connection in Annandale affected uploading to the internet. It is recommended to put something into the code to prevent this affecting the design.
* We only utilised one op-amp on the dual op-amp packages. This was done for fault finding purposes, but the design could be minimised by removing two op-amps.
* Reducing the clock speed further would save power.
* Using a linear voltage regulator instead of the dissipation method is more practical.
* The first 3D box was printed with the red filament and did not work. It is recommended to make the words large on the box for printing purposes and at a slower speed.
* Build a GUI for the Pi or have an executable script which could connect to the internet and run the program and at points if the internet speed is slow, it could empty the buffer.
* It is recommended that the calibration process for the CT channels is simplified.
* It is recommended to buy a larger processor, or have them pinned out on small sample PCBs.

# CONCLUSION

The aim of this project was to design and implement the CTM-3501. This report details the design, implementation, and verification. The CTM-3501 was implemented successfully using C language, C++, and HTML. During testing it was successfully verified to have an error of <2%. The design allowed a sensor node to operate analogy circuitry, biased to the midpoint of an ADC range, by a 9V battery without connection to mains electricity. The worst-case current draw reduced standalone operation to 14 minutes, whilst this limitation was successfully circumvented using scavenging power-saving strategies. The CTM-3501 allowed wireless communication between the sensor node and the Raspberry Pi base station which allowed near-to-real time uploading to the CTM-3501 hosted webpage.

In conclusion, the CTM-3501 project was a very interesting multi-faceted design project. It allowed our group to explore aspects of battery life, micro processing, Linux language, web design, and communication protocols which we had not encountered before. It was unfortunate that the ambition to replicate the FRDM-K20DX1285 was not realised. This was proved a test overall, this project was a success in both being as interesting as it was educational,

# REFERENCES

[1] E. Hub. (2017, 17/10*). Current Transformer*. Available: <http://www.electronicshub.org/current-transformer/>

[2] J. C. University, "CC3501 - Lecture Notes," 2017.

[3] Datasheets, “Github” – Various data sheets, 2017, https://github.com/clintonelliott23/CT-Monitoring-Assignment.

LIST OF APPENDICES

[APPENDIX A – SCHEMATIC 1](#_Toc496114080)

[APPENDIX B – PCB LAYOUT 2](#_Toc496114081)

[APPENDIX C – BILL OF MATERIALS 3](#_Toc496114082)

[APPENDIX C – CODE 4](#_Toc496114083)

APPENDIX A – SCHEMATIC

APPENDIX B – PCB LAYOUT

APPENDIX C – BILL OF MATERIALS

APPENDIX D – CODE

APPENDIX E – FINAL PRODUCT

