CHARGE CONTROLLERS : SERIES, SHUNT, MPPT

Renewable Energy Innovation

The goal of my internship is to design and prototype an open-source charge controller for off-grid systems, with a focus on maximum power point tracking algorithms. The first step of the project will be to program a simple series controller and a shunt controller which will protect the battery from overcharging.

IUT PAUL SABATIER : Génie électrique et informatique industrielle

Supervisor : Matthew LITTLE

Student : Arnaud MOULAS

Teacher : Luiz LAVADO VILLA

2015-2016

Acknowledgement

I would like to thanks :

- Lythisha who welcomed me to her house during the internship. She was really nice and she has a a cosy nice house near a supermarket and the tram. It allowed me to buy food for a cheap price and easily go to work thanks to the tram.
- Luiz Lavado Villa to put me in contact with Matthew little from the company Re-innovation.co.uk. Luiz was very active and really helpful. He organized meetings every weeks before the internship to prepare us. It permitted me to understand a bit prior to the internship the work I would have to do.
- Matthew Little who is my supervisor. He was attentive to me so it was a pleasure to work with him. He taught me a lot about how to lead a project from scratch. I am always surprised when I have a problem that Matthew can find the solution quickly.

Synopsis :

Acknowledgement : **2**

Presentation of the company : **4-5**

Introduction : **6**

- Project 7
- What is a rechargeable battery ? 8
- What are the causes of damage to a battery ? 9
  - Damage from cell reversal 9
  - Damage from overcharging 9
  - Damage from overcharging 10
- Different type of charge controller 11
  - Series controller 11
  - Shunt controller 12
  - Maximum Power Point Tracking (MPPT) 13

Presentation of my work **14**

Algorithm **25**

- Series Algorithm 25
- Shunt Algorithm 26
- MPPT algorithm 27
- Arduino program for Series /Shunt algorithm : global view 28
- Arduino program for Series /Shunt algorithm : Enable menu 29

Outcome of my internship **30**

Algorithm **31**

Annexes **31**

Presentation of the company :

Re-innovation is a small business company created by Matthew little. It is mainly accessible for the customer through the website www.re-innovation.co.uk

Matthew's workshop is at the second floor

"Renewable Energy Innovation specialize in electrical and electronic systems for renewable energy projects, mainly solar, wind and micro-hydro. We focus on renewable energy based stand-alone power supply systems (off-grid systems). This includes power and energy monitoring, battery charge control and wiring systems. Projects we have been involved with range from small, portable solar-powered systems, through pedal powered devices up to large multi-kilowatt photovoltaic arrays. The projects, information and blog pages contain details of some of the projects we have worked on. We also design, build and test bespoke pieces of renewable energy equipment." ~Matthew Little

I worked at the Hopkinson gallery which is a vintage shop and a coffee at the same time. Matthew's workshop is on the second floor. I had a desk with all the equipment needed to program the arduino uno and test the algorithm. Matthew's desk is in the next room so it is easy for me to communicate with him.

I have been working in really great condition.

renewable energy innovation
electrical solutions for renewable energy systems

Logo of the company

Introduction :

I am a French student studying computer science. In order to get my degree I need to participate to an internship. Matthew Little responded positively to my mail and I am now happy to spend my internship in Nottingham for the company Re-innovation.co.uk. The plan for me was to design and test different charge controller's algorithm on an arduino uno.

Initially the project was decomposed in 3 big parts :

Series controller :

1) Algorithm
2) Program on arduino
3) Test on the prototype

Shunt controller

1) Algorithm
2) Program on arduino
3) Test prototype

Maximum Power Point Tracking (MPPT) :

1) Algorithm
2) Program it on arduino
3) Test the prototype

However in practice the first 2 steps were merged as their functioning was similar.

Project :

The goal of my internship is to design and prototype an open-source charge controller for off-grid systems, with a focus on maximum power point tracking algorithms.

The first step of the project will be to program a simple series controller and a shunt controller which will protect the battery from overcharging.

Series controller : A series charge controller or series regulator disables further current flow into batteries when they are full.

Shunt controller : A shunt charge controller or shunt regulator diverts excess electricity to an auxiliary or "shunt" load, such as an electric water heater, when batteries are full

Before working Matthew explained me the project and detailed me each steps. He draw simple graphics it permit me to understand how is working the series controller from an electrical point of view.

Exemple of solar panel controller found in shops

What is a rechargeable battery :

A rechargeable battery, storage battery, secondary cell, or accumulator is a type of electrical battery which can be charged, discharged into a load, and recharged many times, while a non-rechargeable or primary battery is supplied fully charged, and discarded once discharged. It is composed of one or more electrochemical cells. The term "accumulator" is used as it accumulates and stores energy through a reversible electrochemical reaction. Rechargeable batteries are produced in many different shapes and sizes, ranging from button cells to megawatt systems connected to stabilize an electrical distribution network.

Several different combinations of electrode materials and electrolytes are used, including lead–acid, nickel cadmium (NiCd), nickel metal hydride (NiMH), lithium ion (Li-ion), and lithium ion polymer (Li-ion polymer). Rechargeable batteries initially cost more than disposable batteries, but have a much lower total cost of ownership and environmental impact, as they can be recharged inexpensively many times before they need replacing. Some rechargeable battery types are available in the same sizes and voltages as disposable types, and can be used interchangeably with them.

Usually the battery used with off grid energy are big and heavy

What are the cause of damage to a battery ?

Damage from cell reversal

Subjecting a discharged cell to a current in the direction which tends to discharge it further to the point the positive and negative terminals switch polarity causes a condition called cell reversal. Generally, pushing current through a discharged cell in this way causes undesirable and irreversible chemical reactions to occur, resulting in permanent damage to the cell. Cell reversal can occur under a number of circumstances, the two most common being:

- When a battery or cell is connected to a charging circuit the wrong way around.
- When a battery made of several cells connected in series is deeply discharged.

In the latter case, the problem occurs due to the different cells in a battery having slightly different capacities. When one cell reaches discharge level ahead of the rest, the remaining cells will force the current through the discharged cell.

Many battery-operated devices have a low-voltage cutoff that prevents deep discharges from occurring that might cause cell reversal.

Damage from overcharging

During the final stages of charging, all lead-acid batteries break down some of the electrolyte in a battery into hydrogen and oxygen. With sealed batteries, such as gel cells and AGMs, the gases are normally contained within the battery, although in certain circumstances (notably, persistent overcharging), enough internal pressure can build up to open pressure release valves and vent the gases. With wet cell batteries – the type that need topping up from time to time – the gases are always vented.

Hydrogen, which is highly explosive, is much lighter than air, so typically will rapidly rise and disperse, so long as there is even minimal venting from the top of a battery box and from the top of the compartment in which the battery box is housed. However, if a pocket of gas forms, any spark (such as from a brushed electric motor kicking on) may set off the hydrogen, on occasion resulting in a powerful explosion. This is what blew the top off the Fukushima nuclear power plants in Japan. Occasionally, internal short circuits create a spark inside batteries that can set off a hydrogen explosion within the battery, blowing the case open. Note that this can occur with sealed batteries as well as with wet cell batteries.

When batteries are worked hard, they generate a fair amount of internal heat. This increases a battery’s ability to absorb charging current without its voltage rising, which effectively disables any voltage regulator on the charging device, resulting in gross overcharging. The battery can get into a condition known as ‘thermal runaway’ in which it soaks up all the charging current that can be thrown at it, converting this into hydrogen, oxygen and heat – a dangerously potent mix. The only effective way to guard against this is to have temperature sensing at the battery, linked to the charging device.

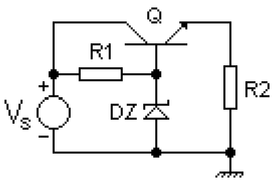
Damage from over dis-charging

If a cell's voltage reaches a designed value, that means the battery has discharged the stored capacity, but still the discharge action is continual, over-discharge will cause. Commonly the end voltage can be identified through discharge current, for example the end voltage is set at 1.0V/cell as discharged at 0.2C-2C, and 0.8V/cell at 3C or more, e.g. 5C or 10C. Over discharge may cause disaster, especially at heavy current or repeated over discharge. Commonly over discharge can make cell inner pressure raise, and the reversibility of activity materials both in positive and negative will be damaged. Even by charge only part can recover and the capacity is reduced obviously.

A laptop's battery can burn if too much damaged

Different type of charge controllers :

Series controller :



Electric scheme for serie controller

The series controller works by providing a path from the supply voltage to the load through a variable resistance (the main transistor is in the "top half" of the voltage divider). The power dissipated by the regulating device is equal to the power supply output current times the voltage drop in the regulating device.

Series controller can only be used for current limited power sources (such as solar panels).

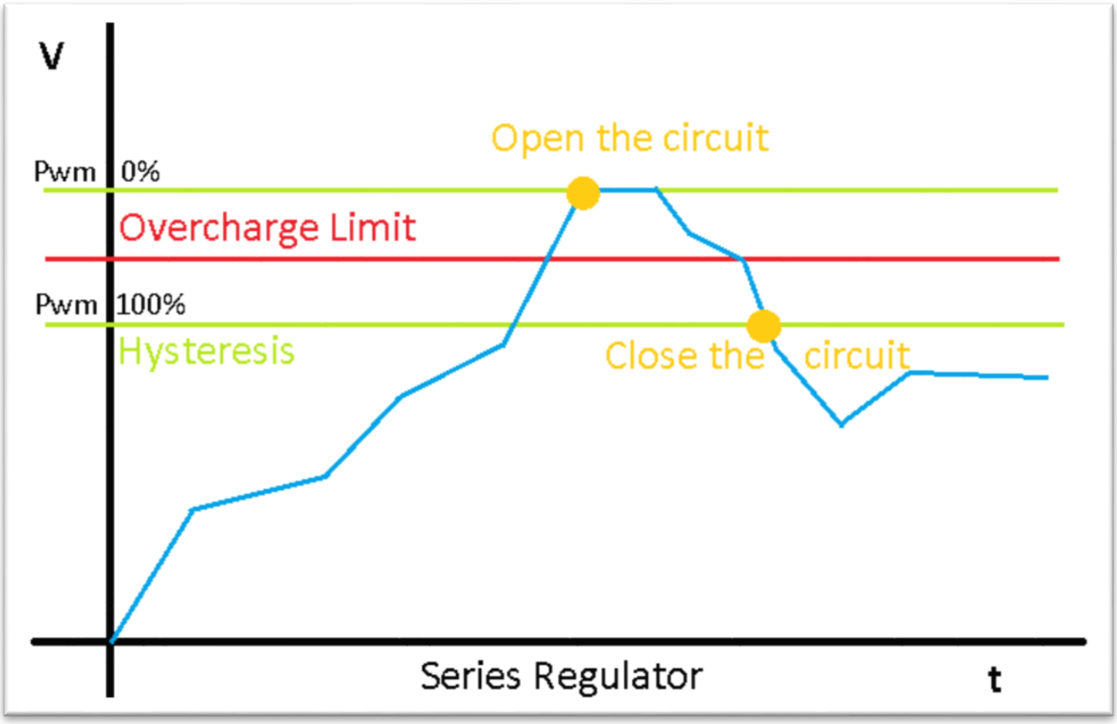
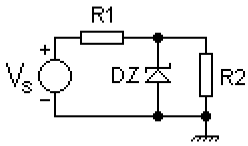


Illustration of the behavior of Series controller

Shunt controller :



Electric scheme for Shunt controller

The shunt controller works by providing a path from the supply voltage to ground through a variable resistance (the main transistor is in the "bottom half" of the voltage divider). The current through the shunt controller is diverted away from the load and flows uselessly to ground, making this form usually less efficient than the series controller. It is, however, simpler, sometimes consisting of just a voltage-reference diode, and is used in very low-powered circuits where the wasted current is too small to be of concern. This form is very common for voltage reference circuits. A shunt controller can usually only sink (absorb) current.

Series controller can be used for Solar, Wind and Hydro.

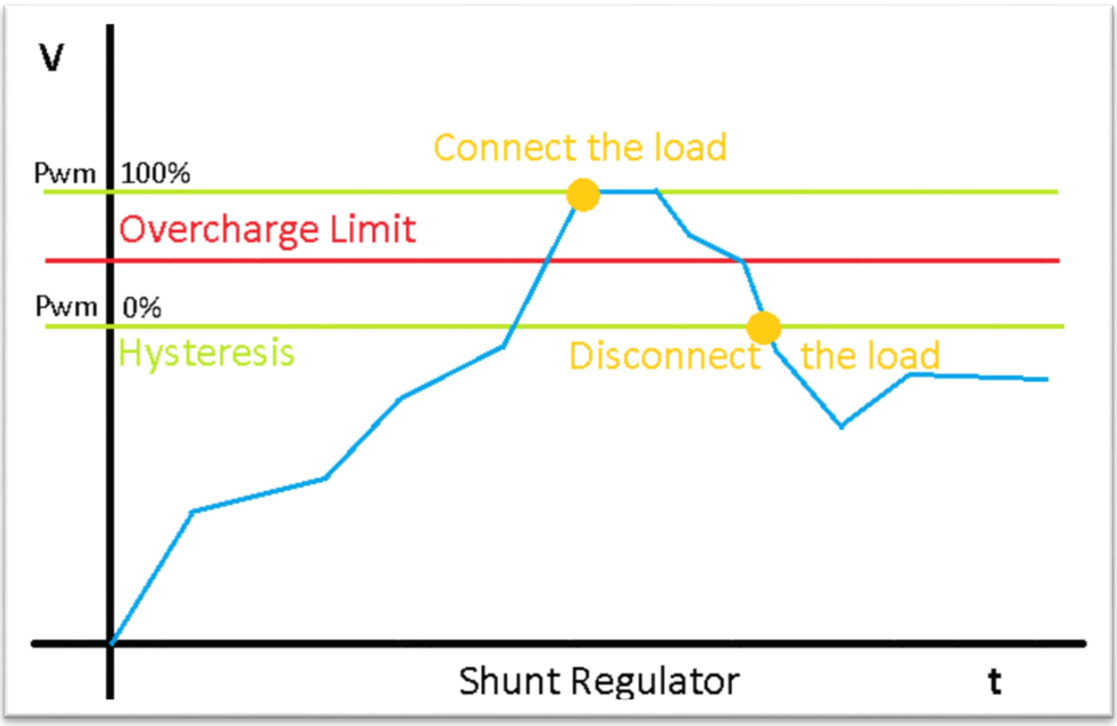


Illustration of the behavior of Shunt controller

Maximum Power Point Tracking (MPPT) :

Maximum power point tracking (MPPT) is a technique used with wind turbines and photovoltaic (PV) solar systems to maximize power output.

The central problem addressed by MPPT is that the efficiency of power transfer from the solar cell depends on both the amount of sunlight falling on the solar panels and the electrical characteristics of the load. As the amount of sunlight varies, the load characteristic that gives the highest power transfer efficiency changes, so that the efficiency of the system is optimized when the load characteristic changes to keep the power transfer at highest efficiency. This load characteristic is called the maximum power point and MPPT is the process of finding this point and keeping the load characteristic there. Electrical circuits can be designed to present arbitrary loads to the photovoltaic cells and then convert the voltage, current, or frequency to suit other devices or systems, and MPPT solves the problem of choosing the best load to be presented to the cells in order to get the most usable power out.

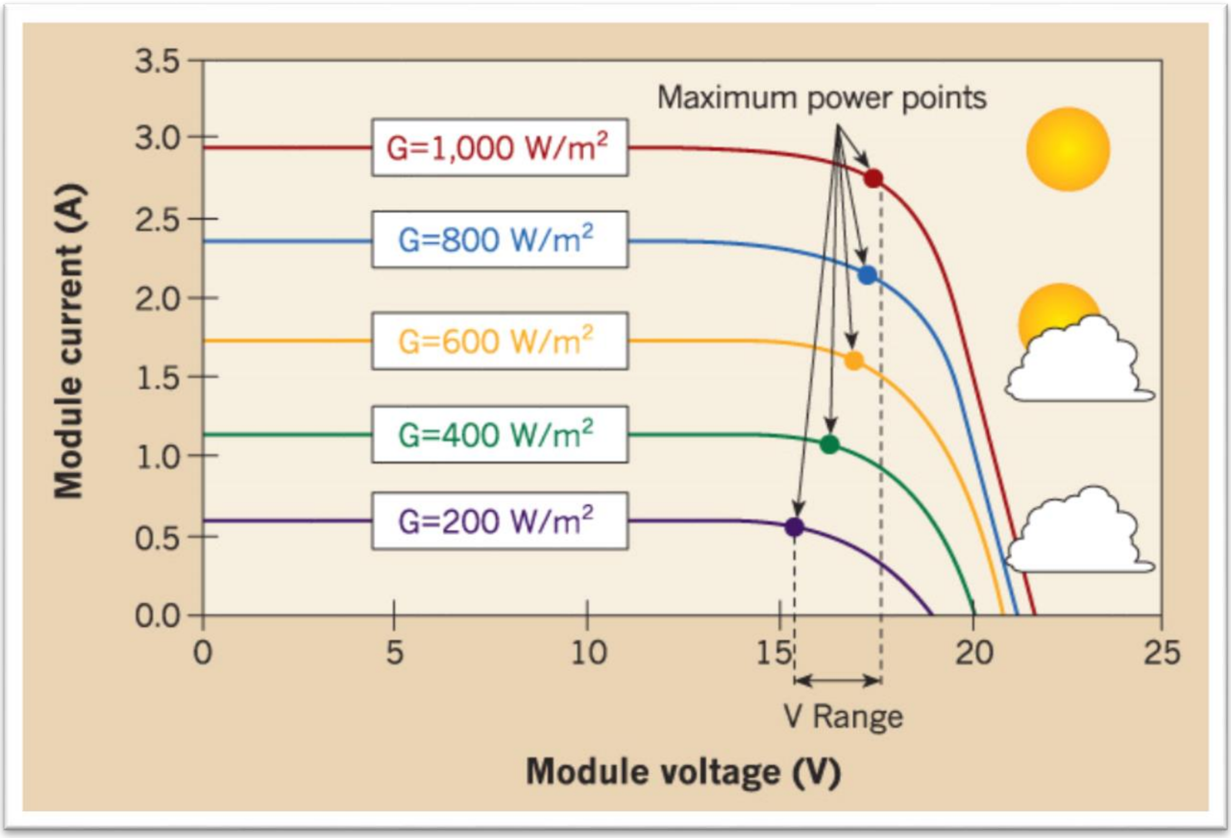


Illustration of the behavior of a MPPT controller

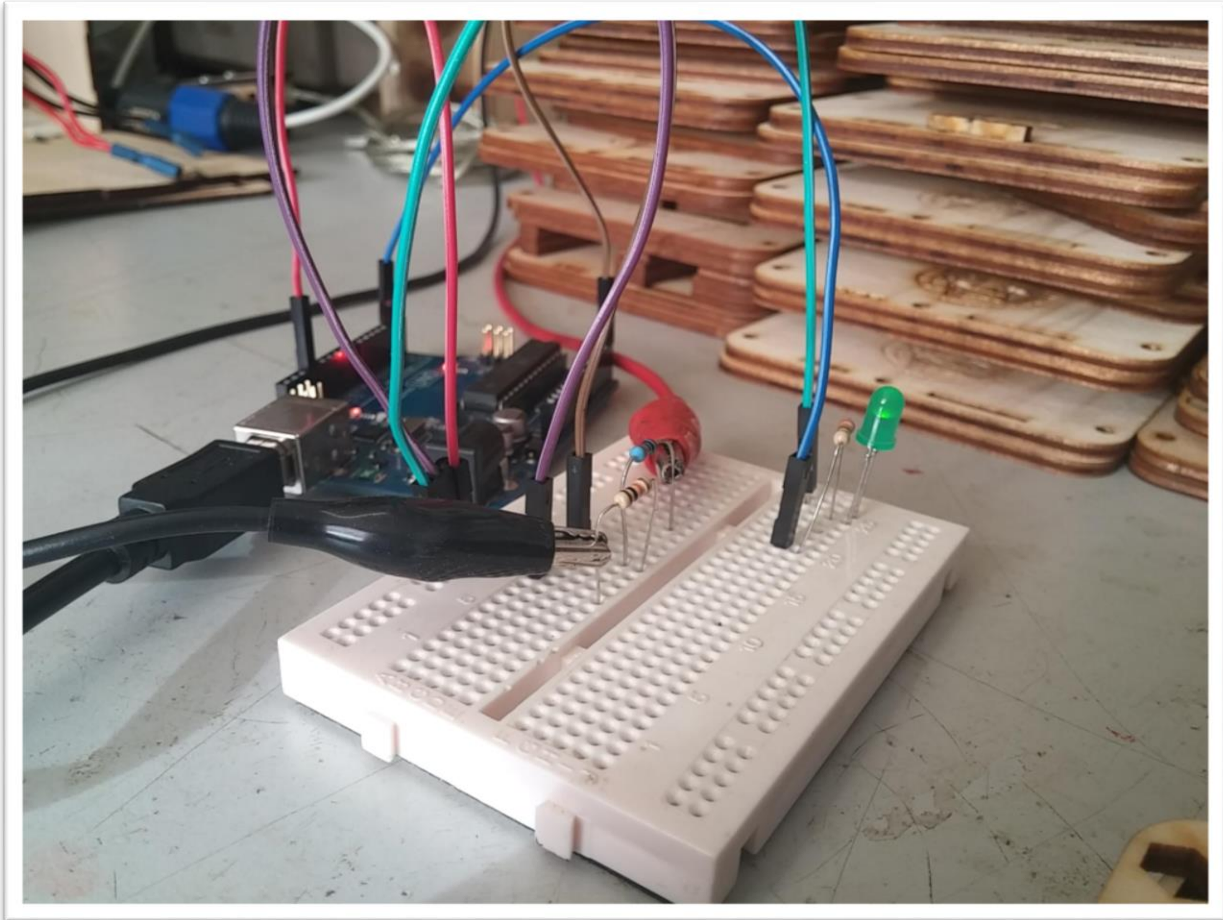
Presentation of my work

Preparation of the test board :

During the first week I did some research to understand the functioning of charge controllers (series, shunt, MPPT). The goal of my first program was to simulate the behavior of the series controller on the serial output. I programmed the arduino in order to process the input measured voltage. It will write “light on” or “light off” depending on the state of the battery. (overcharged or not)

Once this was working Matthew proposed to test it by using a prototype board with a led. The led will represent the state of the mosfet. (open = light on , close = light off)

The white piece of plastic is the prototype area. It is used to easily connect different component.



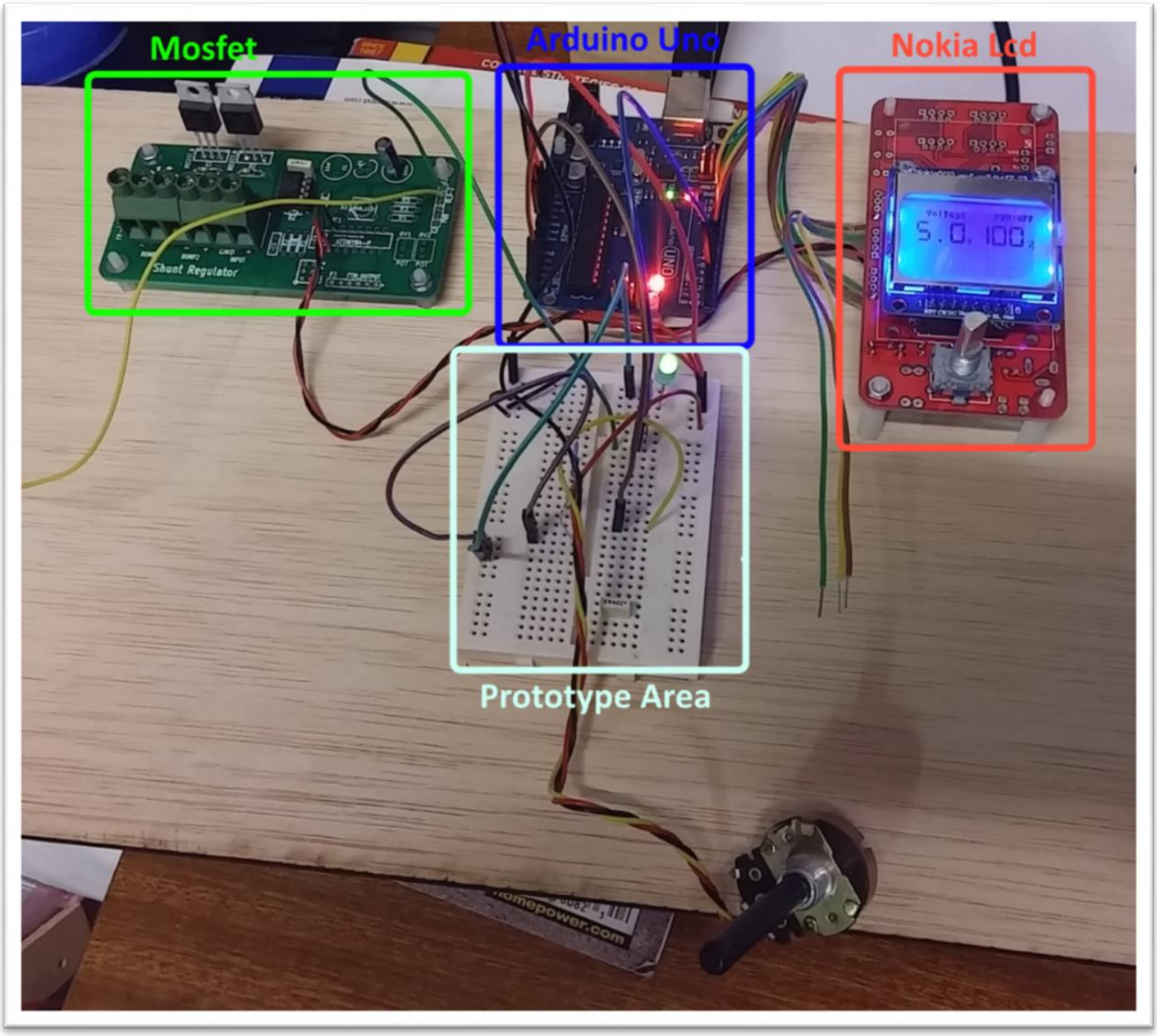
Test on the prototype :Processing the voltage input to determine the state of the led

Currently I am measuring the voltage after the potential divider on the analogue Input of the arduino. I can change the voltage thanks to a power supply and if the voltage is higher than the security limit of the battery I will light the light off the led.

I took some time to learn how to convert the analogue value in voltage (I used the sensibility in voltage of the arduino to convert the value from analogue to volt) and I needed some time to get used programming the arduino in C.

First contact with the fully functional prototype of series/shunt controller

Matthew built me a prototype of series / shunt controller with a board of wood, and arduino, a nokia LCD, a LED and some other electrical component.



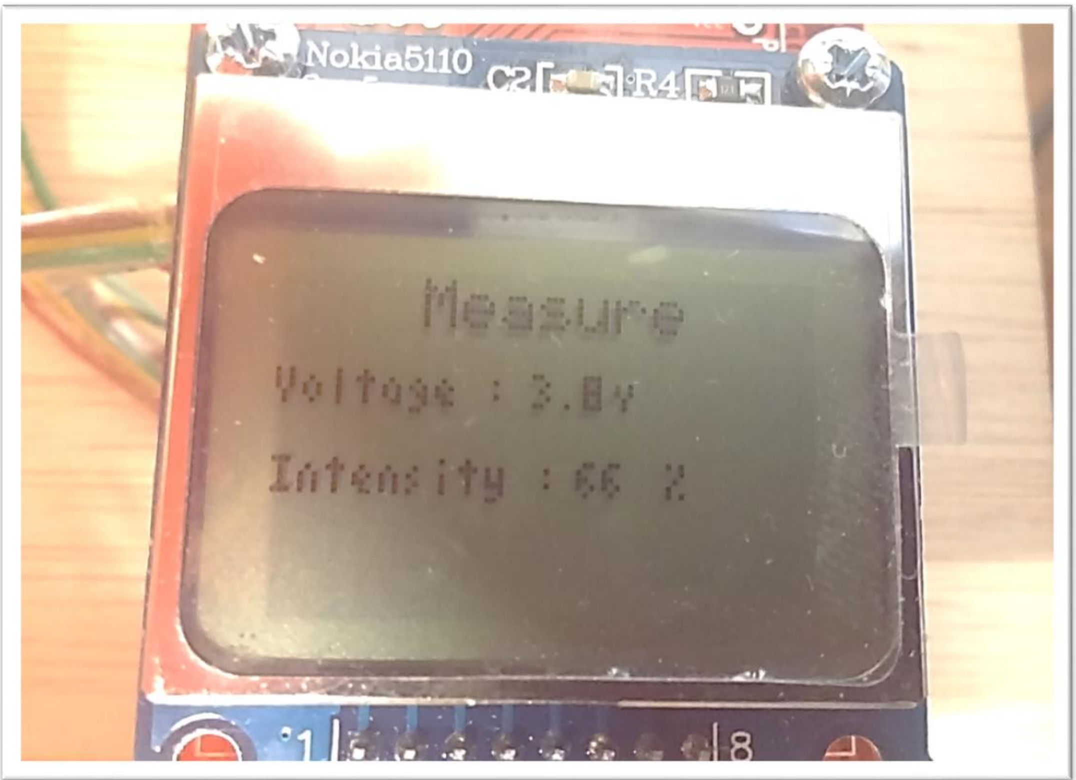
here is the prototype with the lcd working, it's a bit ahead of what I programmed at the moment

I was surprised by the quality of the prototype he screw the electronic board on the wooden board. Thanks to it's robust structure I could bring the prototype home without doubts.

- The green board is the mosfet part.
- The blue board is the arduino uno which will control all component in function of the measured voltage.
- The white one is used to easily connect or remove component
- The red one was designed to be for GPS use but Matthew only connected the nokia lcd and a digital encoder (button)

It took me some time to understand how to use the different components. In particular the Nokia LCD and the digital rotary encoder. After some research I decided to use an online library to control the lcd. I tested the different libraries to find the easiest one to use and which provide a lot of function.

My first display was this one :



First grip on the use of library and nokia LCD

It was very basic but I managed to get the value actualized in real time. It was good enough to test my series / shunt controller at the moment.

The next step was designing four different functions which will be used to control the PWM. It means the led will light according to the quantity of current. (the led is a representation of the current the controller let pass) Here are the four functions :

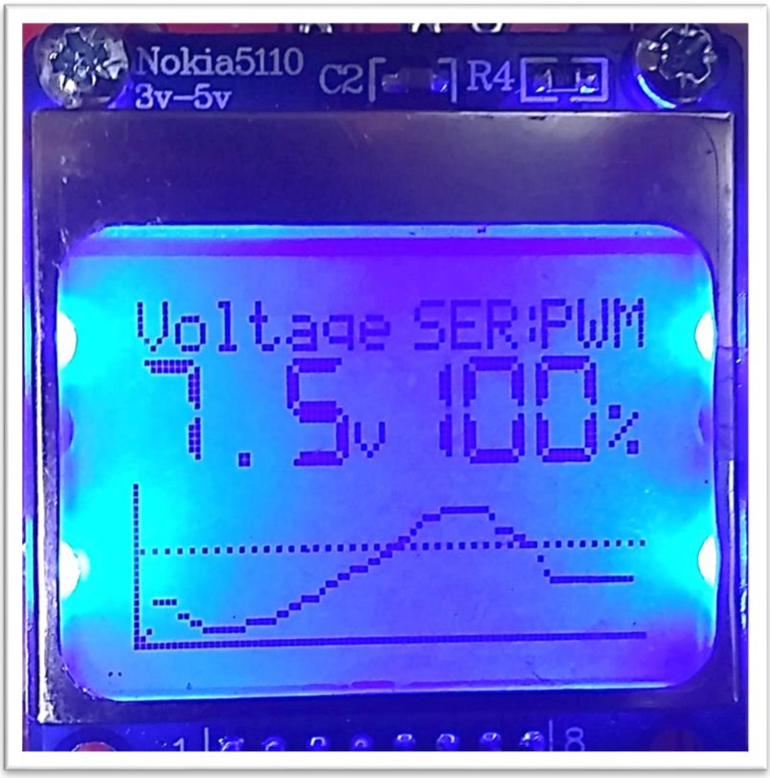
- Series controller
- Series controller with pwm
- Shunt controller
- Shunt controller with pwm

Implementation of the algorithm on the prototype board

I started to control the arduino with more advanced function. I used the function “map” in order to interpret the voltage value. These values are then used to draw a graphic on the screen which represent the voltage value over time. Thanks to the map function it is compatible with every voltage range. However under 22 volt of range it will not draw a nice curve because there is 22 dot on the y axis. If there is less than 1 volt per dot it will still draw the point put the curve will look like a “stair”.

Then my work got focused to control the digital rotary encoder in order to switch between the different control mode. When you turn the digital encoder to the right every five click it will switch to next mode.

Another little function I added is the Sleepmode for the lcd. However it was not really interesting the way I controlled it in the first place. (because it was in confrontation with the normal use of the controller) so I kept it as a draft and will implement it later.



Nokia LCD : Display result

Program the board for easy control

The main goal was to make accessible the interface. I had to think how to facilitate the user's utilization of the prototype. The prototype had only one button which is a digital rotary encoder. In order to realize this I deplaced the sleep function of the lcd. From now when the button is pushed it will enter the menu.

In the menu, the user will be able to change every parameter in order to adapt the charge controller to his battery. It will be possible to activate the sleep mode in the menu.

The biggest difficulty I encountered was the lack of memory (ram). I had to use the eprom and store in it the all the text (print).

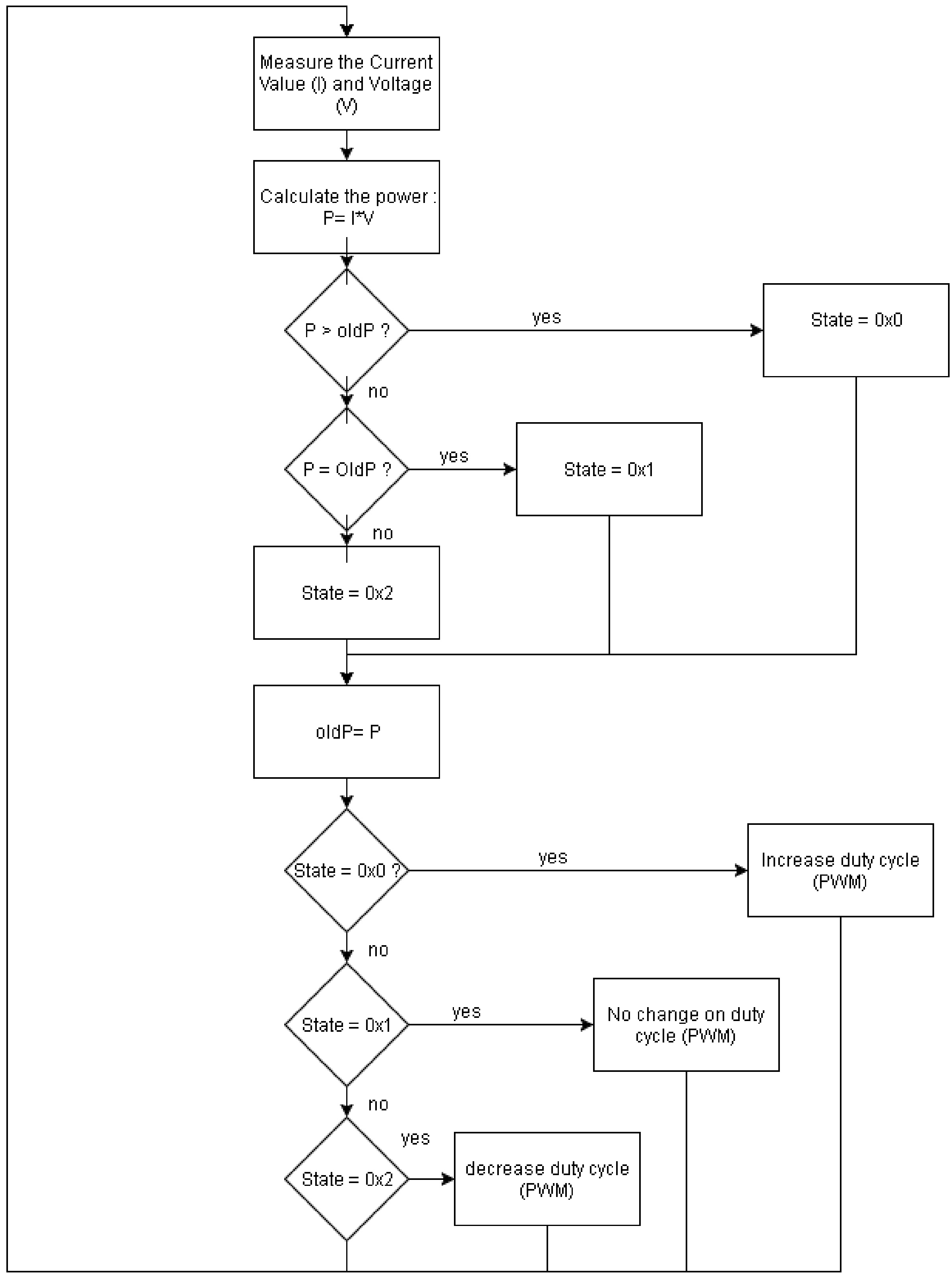
After that Matthew told me to store the values in the eprom to prevent the loss of the information if there is a reset. Once I did it, I got some difficulty to read the value. I add to find 'complex' arduino function on the internet (they were not written explicitly on the arduino website but they are in the base arduino library).

At this point the program and the prototype looks like a finished product.

Major difficulties for the Series and Shunt controller program :

- Convert the theory of a charge controller into a program
- Initialize and use the different components (lcd, digital encoder)
- Need to free memory (ram)
- Write and read the eprom correctly
- Adapt the prototype to the user utilization

Algorithm and first program of an MPPT



I started to work on the MPPT. The initial step was to understand the theory behind an MPPT. There was multiple methods to design one. Matthew gave a lot of good thesis or documents to find information about differents MPPT. We choose the “hill climbing method” because it’s easy to understand.

At this point Matthew was thinking how to improve the prototype in order to allow me to test my program on it.

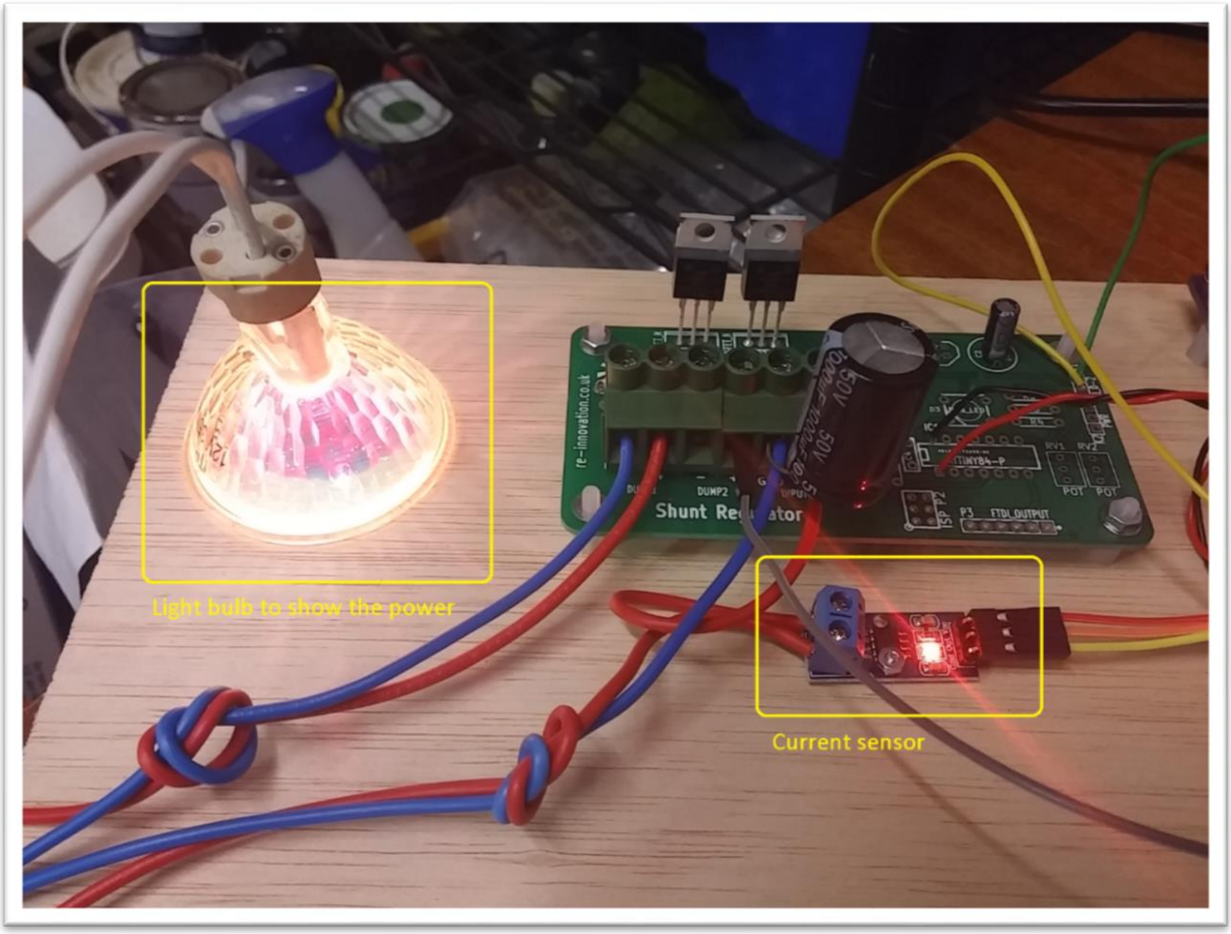
I designed a first algorithm which I sent to Yoann (another student from GEII which is in Peru !) in order to share our work on the subject . Here is my first design (on the previous page)

This design has many flaw but I did not realized it before testing it on the prototype.

During the first testing I though the algorithm was working as intended because of the model of power supply I was using always deliver the highest current. Basically my algorithm was just increasing the value of the pwm.

However the behavior of a solar panel is completely different. I will detail this problem a bit later.

Prototype board upgraded to support an MPPT algorithm:

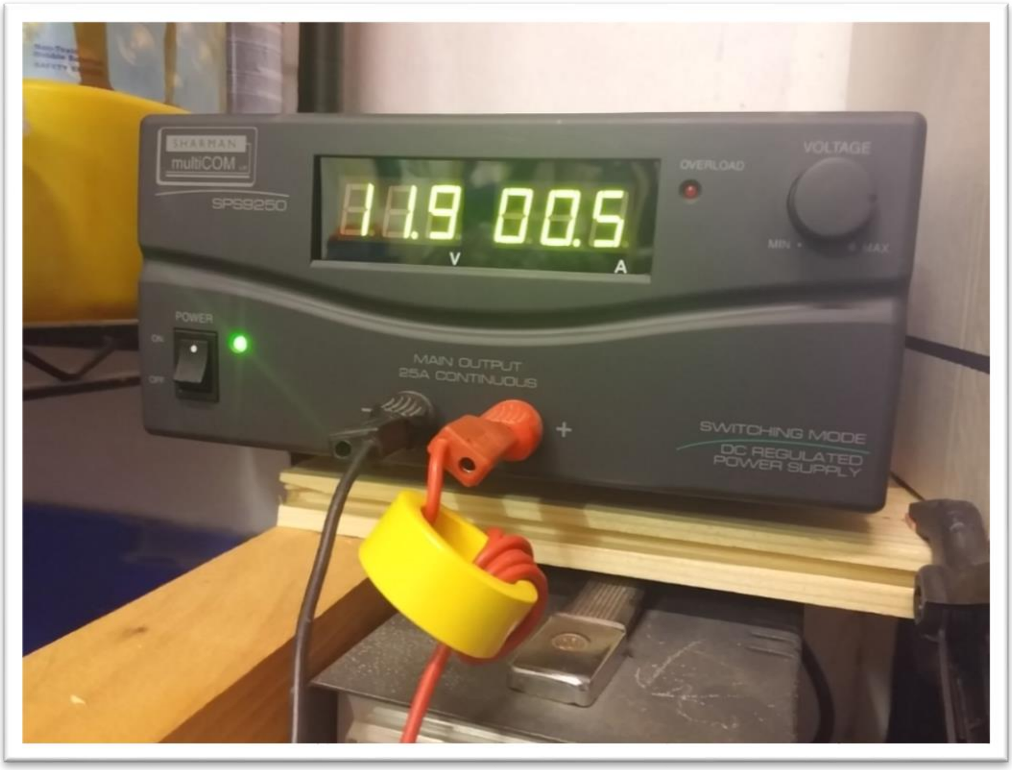


Matthew adapted the prototype to work with an MPPT program. As you can see on the picture he added a current sensor (which is used to convert the current into a voltage which is read on the arduino analog input)

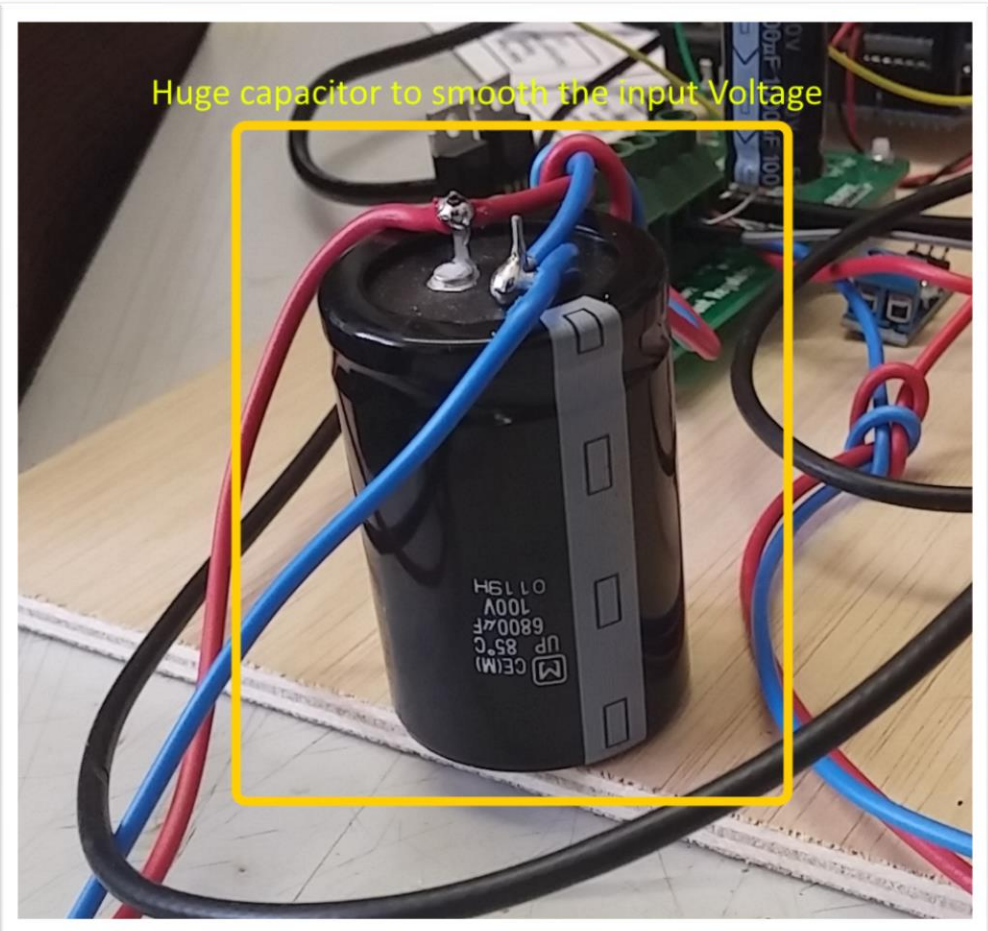
However we got a problem of electricity noise. It damaged the nearest electronics. Matthew fixed this problems by forming an inductor with the cable and adding a capacitor on the input voltage from the power supply. At first I thought my program was working as intended because this power supply will always deliver enough current to reach the higher power.

The maximum power point I was looking for was moving on a linear line. When we then proceeded to test the prototype whit Matthew Power Supply (it is possible to limit the current on this one) We saw that my program reached its limit : it got further than the MPPT.

I also did a scan function which will only execute itself 1 times at the startup. It will measure the power for each values of the pwm (0 – 255) and then save the pwm’s value with the higher power. It will start the main program at the MPPT and then the main program (in the arduino’s loop) will track the maximum power point according to the variations.



I had high quality equipment at my disposal



This big capacitor was really needed to smooth the input voltage

(otherwise the measured value was not stable enough)

Difficulties for the MPPT algortihm :

- Wrong tracking when going the other side of the curve.

  On my first try to design an algorithm I was using a power supply which has no limitation on the current. It always deliver the most current to reach the highest output possible. The result was that I designed an algorithm to climb a straight line. When I tested with Matthew the algorithm on a power supply with limitation of current the algorithm could not track the maximum power point. Indeed it was lost on the right side of the curve.

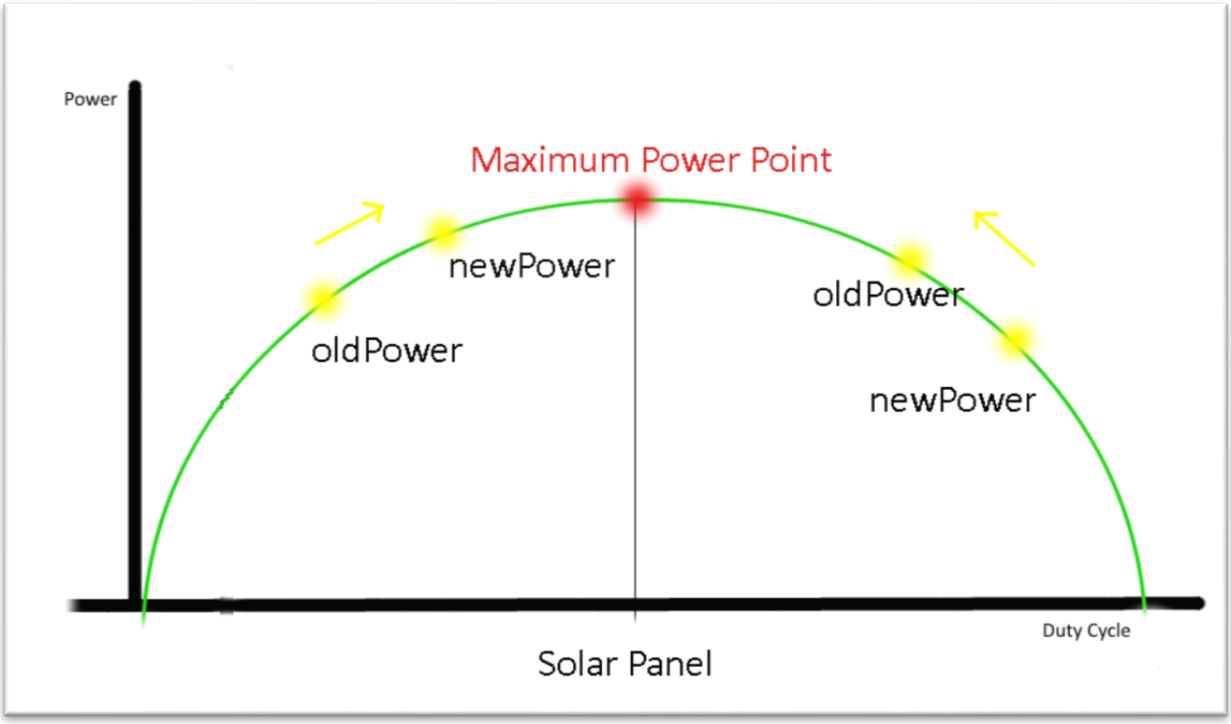


Illustration to show the need to adapt the algorithm to the side of the curse

- Second MPPT problem : Multiple Maximum

Once the MPPT algorithm fixed there was another problem. While the tracking of the maximum power point was working there was the problem of being stuck on a small value (See below). I designed a scanning function which will execute itself every 10 min in order to find the highest point. It will prevent the algorithm of being stuck on a small pike.

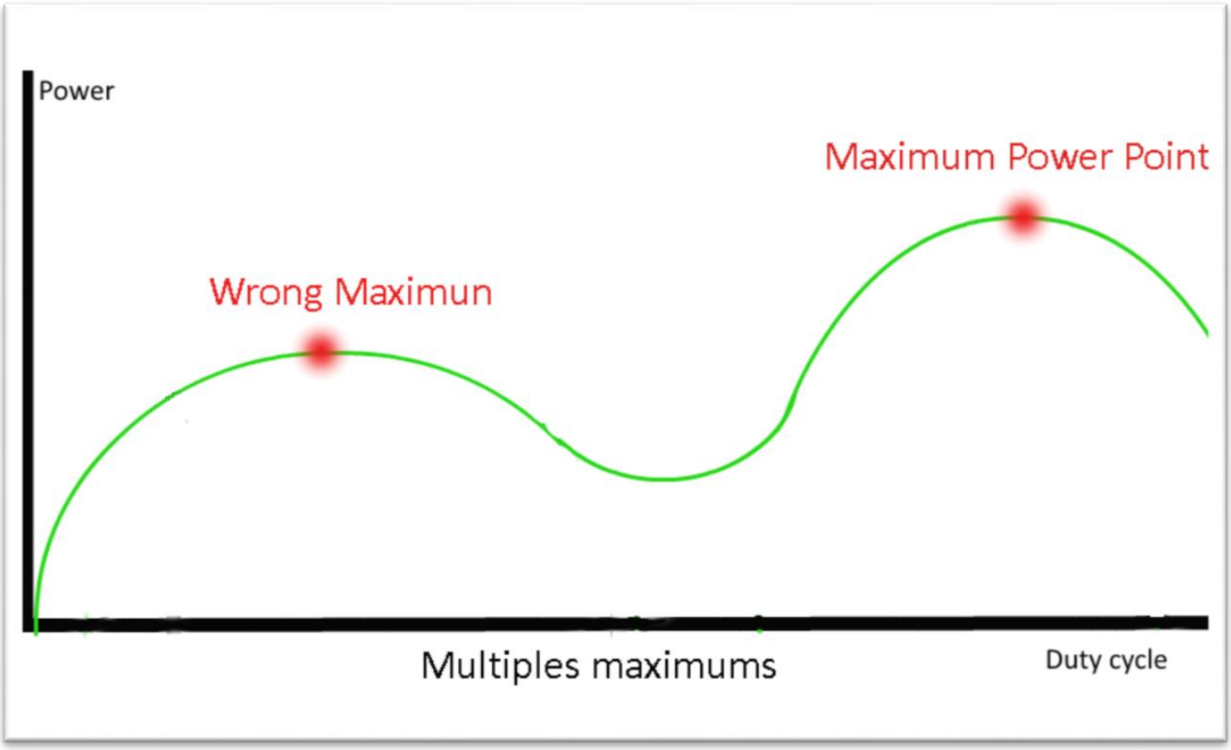
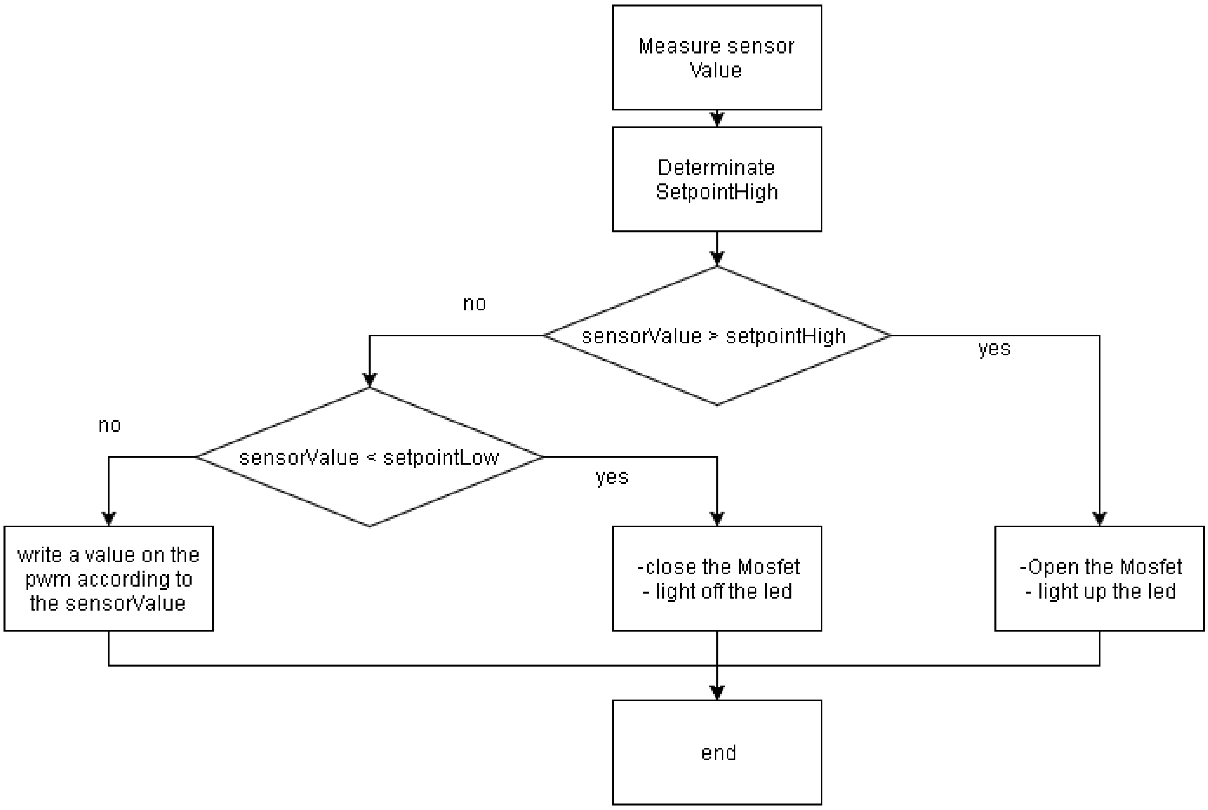


Illustration to show the risk of being stuck at a wrong maximum

The next improvement will be to create a function which will adapt the variation of the pwm value to improve the speed of the tracking. It will need to decrease to the minimal variation of the pwm (+1/-1) once it's near the MPPT. (To prevent from going to much farther of it)

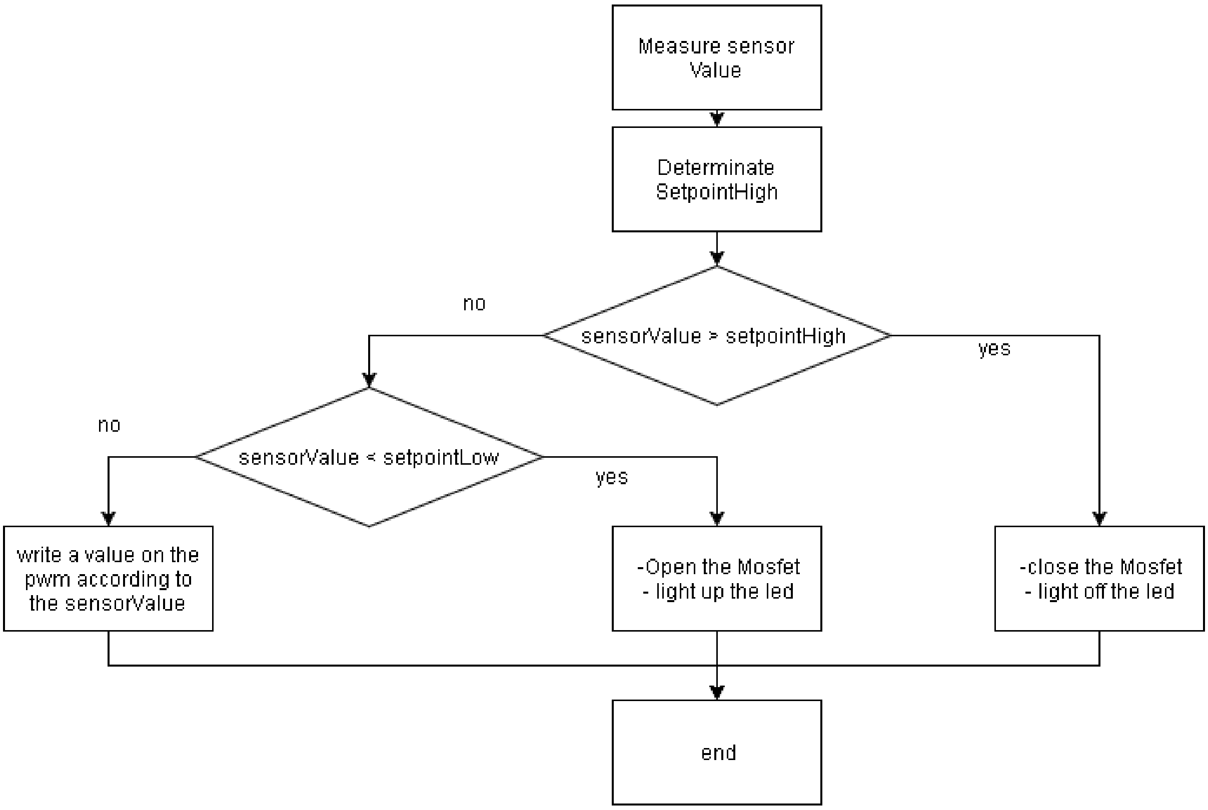
Algorithm:

Series algortihm :



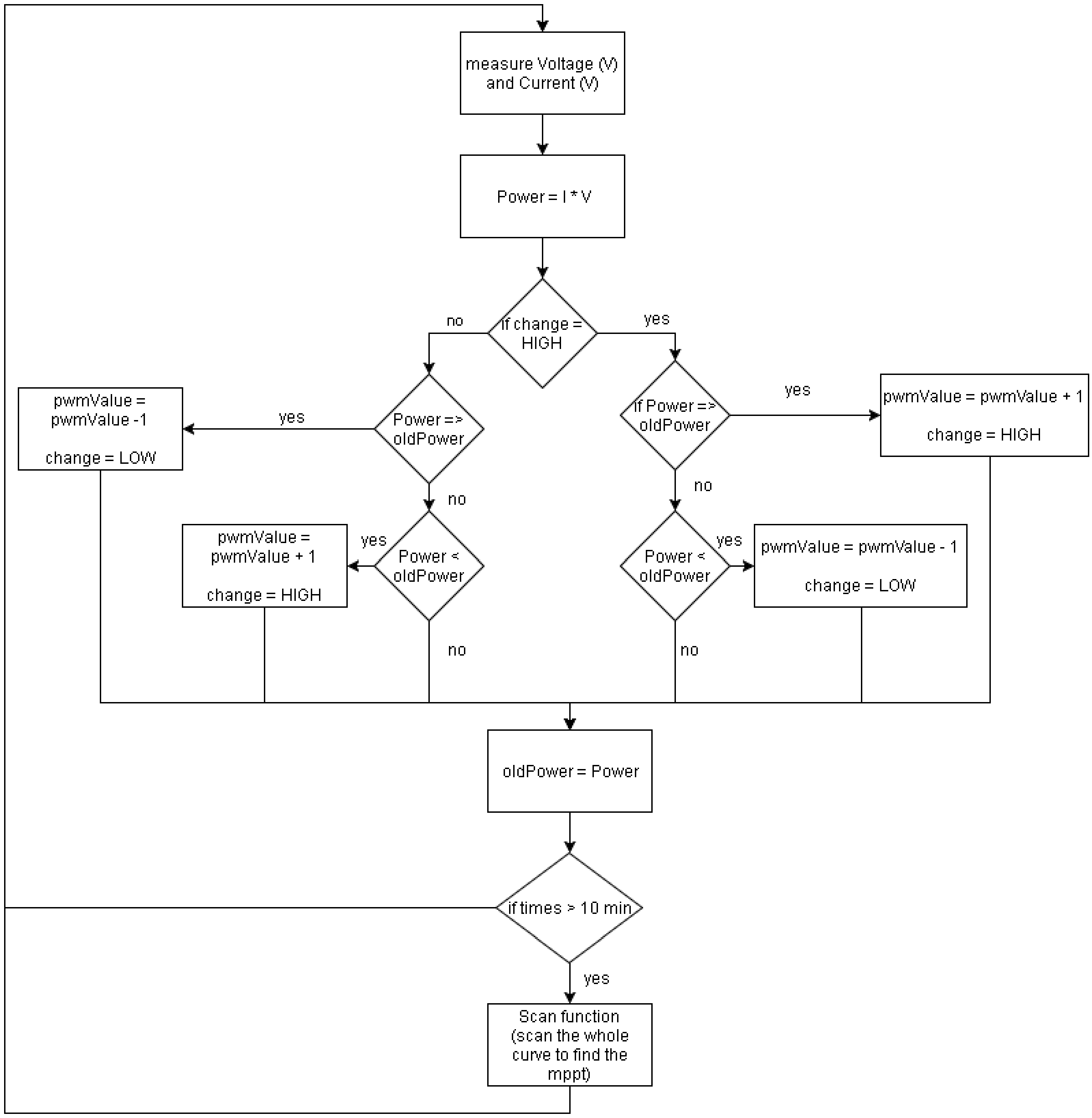
Series Algorithm

Shunt algortithm :



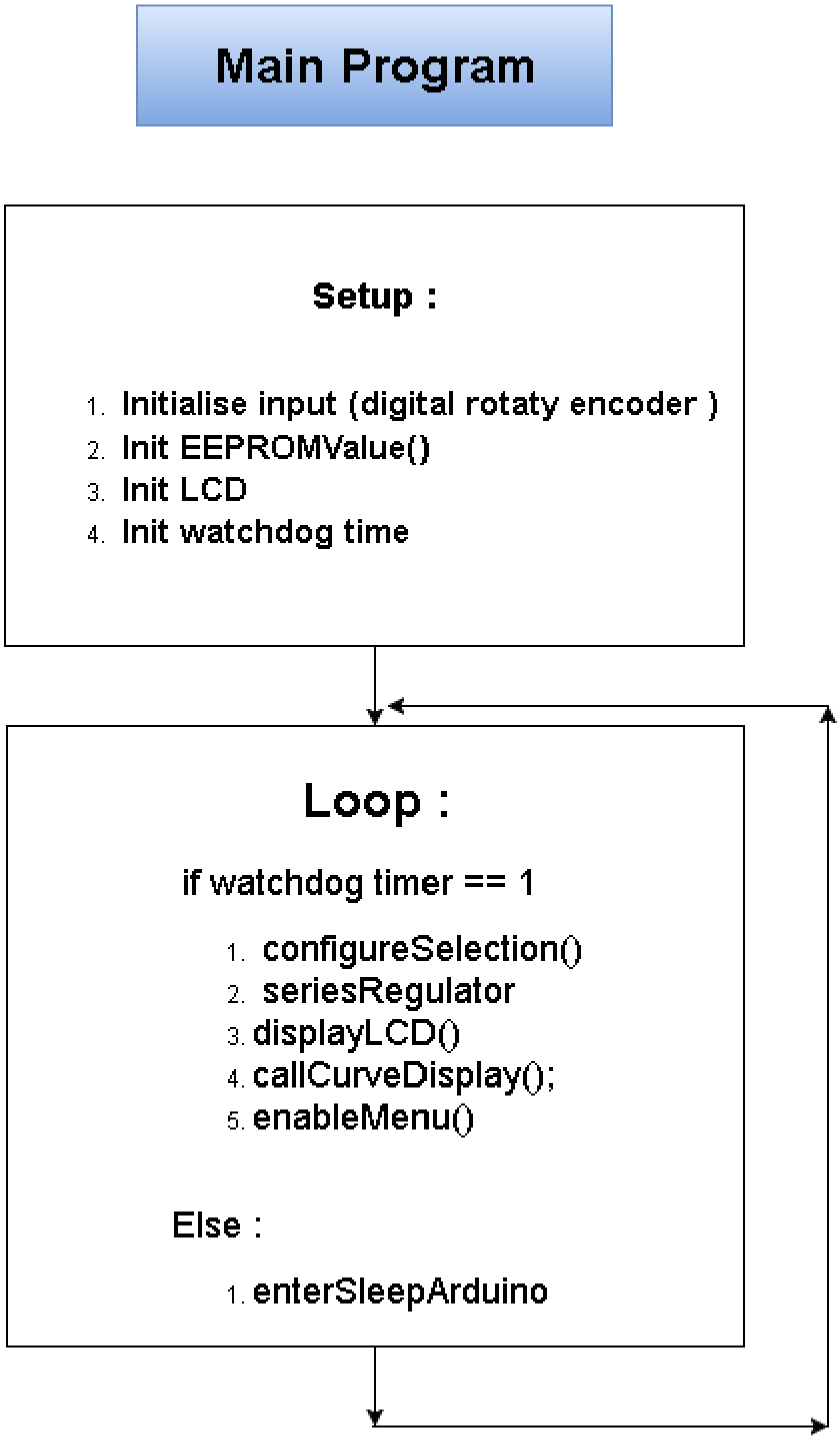
Shunt Algoritm

Mppt algorithm :



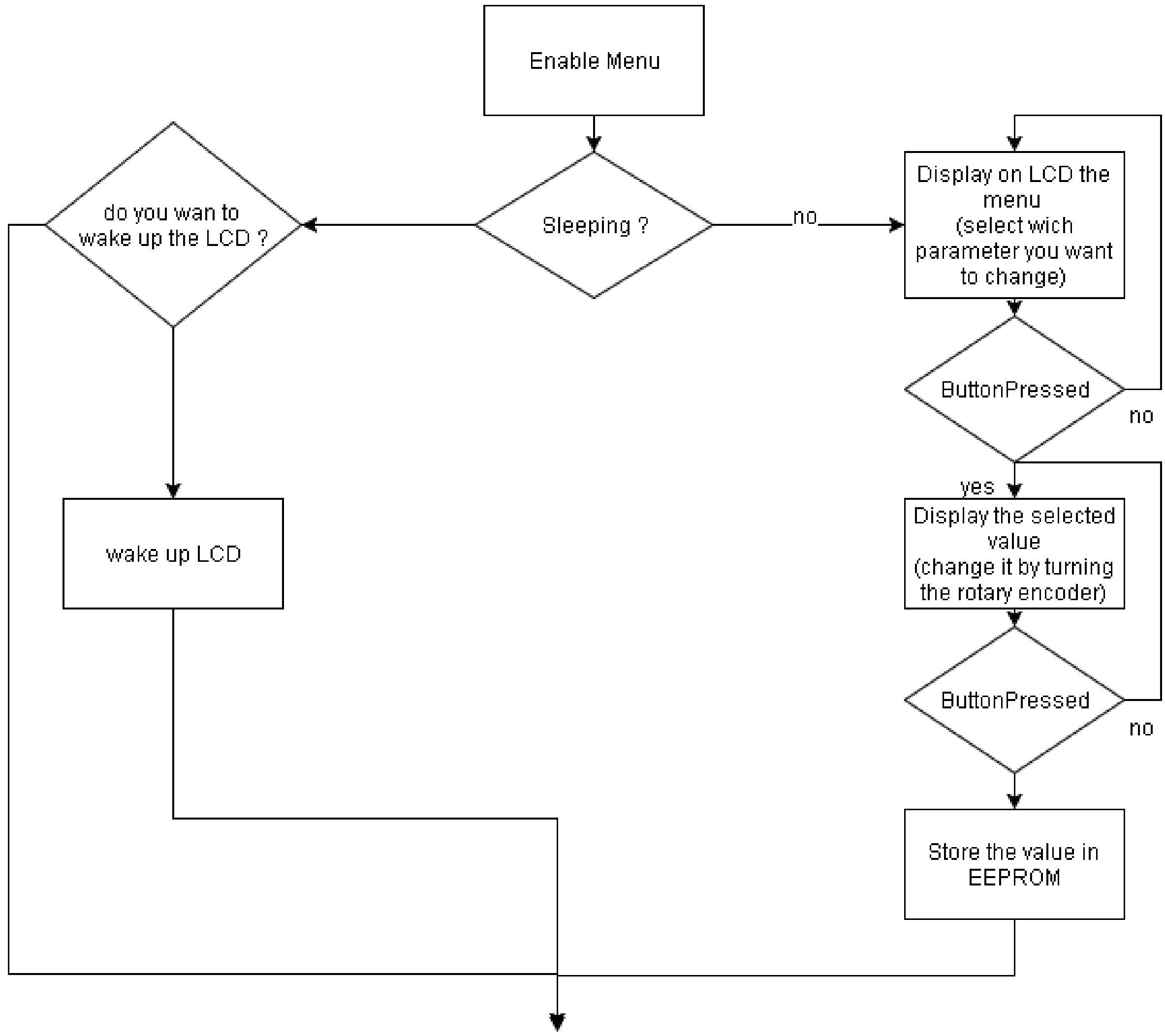
MPPT algorithm

Series/Shunt Arduino program :



Arduino program of the Series / Shunt controller

Enable menu algorithm :



Algorithm of the function "Enable menu" in the arduino program

Outcome of my internship :

For Re-innovation.co.uk

I designed and implemented a series and shunt controller program. I believe this program is well finished because I had the time to test it with Matthew. We corrected several little problem which would occur for a normal user.

It is possible to sell kit for people who want a customizable Series / Shunt controller.

For me :

I learned a lot at the side of Matthew. He is a very dynamic person and is always listening to me. I learned how to lead a project from scratch. That means starting with a design on paper to a fully functional prototype.

It developed the skills I learned from my formation GE2I (in both electronic and programming).

Furthermore on the personal side it confirmed me that I would like to pursue with engineer study especially in the domain of programming.

Nottingham is a nice city, it is near London and Birmingham so it is easy and cheap to do sightseeing. It’s a smaller than Toulouse but the architecture of the city center is way more contemporary than Toulouse one. There is a more three and public garden too. I really liked doing my jogging around the city.

I really liked to work with the Luiz’s association, it is a pleasure to share my work with Yoann Rey. I really feel the impression to be part of a bigger project.

Bibliography

- Wikipedia : Theory behind the Series , Shunt , MPPT controller
- Matthew's book *"Engineering in development Energy" :* Theory about Series / Shunt / MPPT
- Batteryspace.com : Information about the batteries (like causes of the damage to a battery)
- MPPT algorithm : Thesis from Esram IEEE 2007 *"Hill climbing method"*

Annexes

Code in C for arduino Uno :

(most important functions)

- Annexe 1 : DrawBatteryLimit
- Annexe 2 : DrawCurve
- Annexe 3 : PinA
- Annexe 4 : PinB
- Annexe 5 : MPPT Algorithm
- Annexe 6 : Series Algorithm
- Annexe 7 : Shunt Algorithm

ANNEXES

Code in C for arduino Uno :

(most important functions)

- Annexe 1 : drawBatteryLimit – Draw the overcharge limit on the nokia lcd
- Annexe 2 : drawCurve _ Draw the evolution of the input voltage overtime and display it on the nokia LCD.
- Annexe 3 : PinA _ Used to detect the rotation of the digital rotary encoder
- Annexe 4 : PinB _ Used to detect the rotation of the digital rotary encoder
- Annexe 5 : MPPT algorithm _ Code of the MPPT algorithm
- Annexe 6 : ControlSeriePWM _ Code of the Series controller with PWM
- Annexe 7 : ControlShuntPWM _ Code of the Shunt controller with PWM

Annexe 1 : drawBatteryLimit

```
void drawBatteryLimit(void)
{ float mappedVinMax;
  char positionX;
  bool test =0;
  mappedVinMax=map(VinMax,0,maxControllerVoltage, 0, 22);
  for (positionX=1; positionX <= 83; positionX++)
  {
    myGLCD.setPixel(positionX,(46-mappedVinMax));
    if(test==0)
    {
      test=1;
    }
    else
    {
      myGLCD.invPixel(positionX, (46-mappedVinMax));
      test=0;
    }
  }
}
```


Annexe 2 : drawCurve

```
void drawCurve(byte numberArray)
{ float mappedValueVin;
  byte testNumberArray;
  mappedValueVin=map(Vin,0,maxControllerVoltage, 0, 22);
  arrayVin[numberArray]= mappedValueVin;
  for (testNumberArray=1; testNumberArray <= numberArray; testNumberArray++)
  {
    myGLCD.setPixel(testNumberArray,(46-arrayVin[testNumberArray]));
  }
}
```

Annexe 3 : PinA

void **PinA**(){

cli(); //stop interrupts happening before we read pin values

reading = PIND & 0xC; // read all eight pin values then strip away all but pinA and pinB's values

if(reading == B00001100 && aFlag) { //check that we have both pins at detent (HIGH) and that we are expecting detent on this pin's rising edge

encoderPos --; //decrement the encoder's position count

bFlag = 0; //reset flags for the next turn

aFlag = 0; //reset flags for the next turn

}

else if (reading == B00000100) bFlag = 1; //signal that we're expecting pinB to signal the transition to detent from free rotation

sei(); //restart interrupts

}

Annexe 4 : PinB

void **PinB**(){

cli(); //stop interrupts happening before we read pin values

reading = PIND & 0xC; //read all eight pin values then strip away all but pinA and pinB's values

if (reading == B00001100 && bFlag) { //check that we have both pins at detent (HIGH) and that we are expecting detent on this pin's rising edge

encoderPos ++; //increment the encoder's position count

bFlag = 0; //reset flags for the next turn

aFlag = 0; //reset flags for the next turn

}

else if (reading == B00001000) aFlag = 1; //signal that we're expecting pinA to signal the transition to detent from free rotation

sei(); //restart interrupts

MPPT algorithm

Annexe 5 : MPPT algorithm

```
if (change == HIGH )                    //algorithm of the mppt
 {
  if (newPower >= oldPower)
  {
   pwmValue = pwmValue + (int)pwmVariation;
   change = HIGH;
  }
  else if (newPower < oldPower)
  {
   pwmValue = pwmValue - (int)pwmVariation;
   change = LOW;
  }
 }
 else if (change == LOW)
 {
  if(newPower >= oldPower)
  {
   pwmValue = pwmValue - (int)pwmVariation;
   change = LOW;
  }
  else if (newPower < oldPower)
  {
   pwmValue = pwmValue + (int)pwmVariation;
   change = HIGH;
  }
 }
if (pwmValue >= 255)                    //Prevent the pwmValue to go higher
than its limit
{
 pwmValue = 255;
}
else if (pwmValue <= 0)                 //Prevent the pwmValue to go lower
than its limit
{
 pwmValue = 0;
}
 oldPower = newPower;                   // save the newPower as the oldPower
analogWrite(5, (pwmValue));
```

Annexe 6 : controlSeriePWM

```
void controlSeriePWM(void)

{

 if(verificationSetpointHigh(sensorValue,setpointHigh)==1) //Voltage higher than setpointHigh

 {

  digitalWrite(overchargeIndicatorLed, LOW);          // Light up the led

 }

 else if (verificationSetpointLow(sensorValue,setpointLow)==1)

 {

  digitalWrite(overchargeIndicatorLed, HIGH);          // Light off the led


 }

 else

 {

  PWMwrite (sensorValue, overchargeIndicatorLed);

 }

}
```


Annexe 7 : controlShuntPWM

```
void controlShuntPWM(void)

{

 if(verificationSetpointHigh(sensorValue,setpointHigh)==1) //Voltage higher than setpointHigh

 {

  digitalWrite(overchargeIndicatorLed, HIGH);          // Light up the led

 }

 else if (verificationSetpointLow(sensorValue,setpointLow)==1)

 {

  digitalWrite(overchargeIndicatorLed, LOW);          // Light off the led

 }

 else

 {

  PWMwriteShunt(sensorValue, overchargeIndicatorLed);

 }

}
```