PROGRAMMABLE SWITCHED DC POWER SUPPLY

Under supervision of DR\Mohamed Hamdy

PRESENTED BY : (Groupe 3)

Name	Department	Groupe
Ramez Fayek Gerges	Power	3
Doha Ragab Abd El-Hamid	CSE	5
Abd El-Rahman Ahmed Abd Rab Elnabi	CSE	5
Youssef Ali Mohamed	CEE	2
Youssef Mohamed Abobakr	CEE	2

CONTENTS:

1
Unregulated DC power supply
Three main sub-blocks

2
DC/DC converter
Buck converter

3
Controller
Arduino

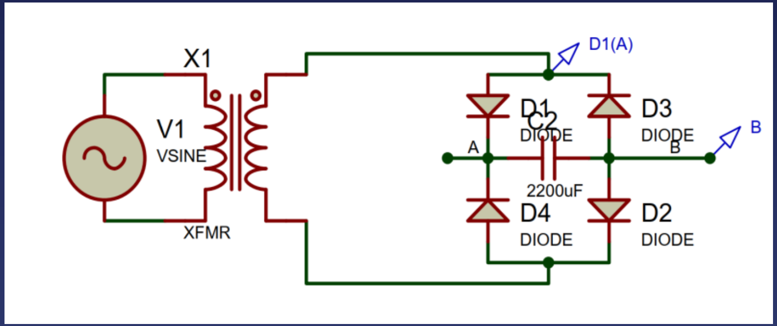
4
Mobile App
Remote XY

5
Summary
Get all together

1

Unregulated DC power supply

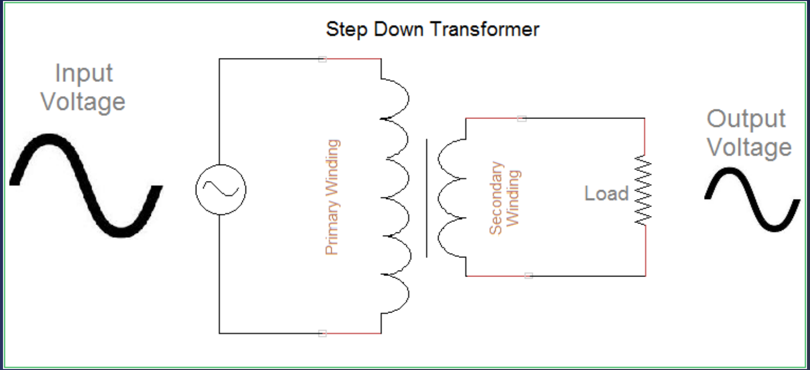
Unregulated DC power supply



- **The power Supply has the following three main sub-blocks.**

1. The Transformer
2. The Rectifier Circuit
3. The Filter

Unregulated DC power supply

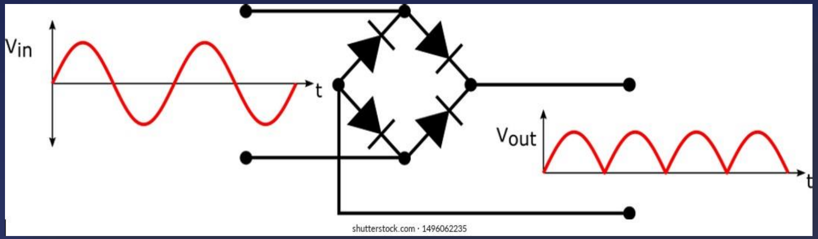


- **The input transformer**

AC has a voltage level of 220/120 V(RMS). We need the input transformer to step down the incoming AC to our required lower-level i.e. close to 16V (DC). **We Used 16V stepdown transformer**

Unregulated DC power supply

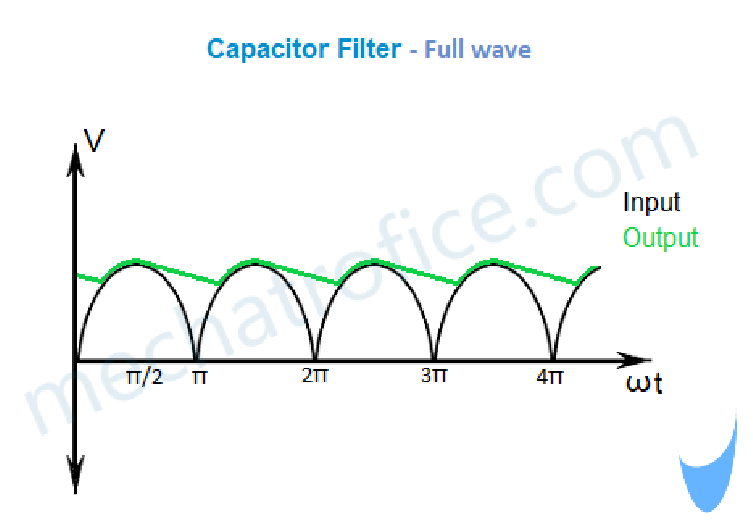
- **The Bridge Rectifier**



A rectifier circuit is the combination of diodes arranged in such a manner that converts AC into DC voltage levels.

Unregulated DC power supply

- **The filter circuit**



The output of the rectifier is pulsating and is called pulsating DC. This pulsating DC is not considered good to power up sensitive devices.
So, the rectified DC is not very clean and has ripples. It is the job of the filter to filter out these ripples and to make the voltage compatible for regulation.

Unregulated DC power supply

$$V_{Ripples} = \frac{\left(Vo - 2V_{D,on}\right) * T}{2RC}$$

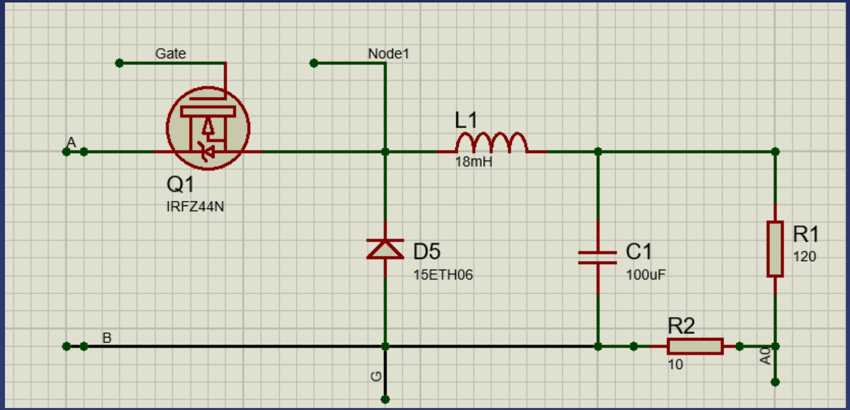
- The amplitude of ripples inversely proportion to the size of the capacitor as we see from the law.
- We use 2200uF cap to make ripples so small.
- Also we use diode from type 1N4007 in the bidge.

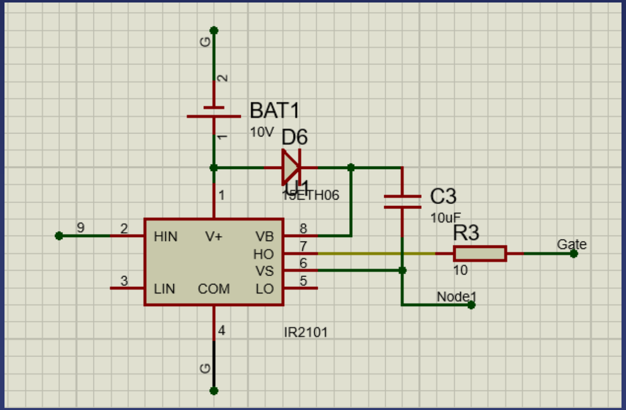
2

DC/DC converter

Buck Converter

- The Step Down Chopper or Buck converter reduces the input DC voltage to a specified DC output voltage.
- The output Voltage depends on the input voltage and the duty cycle.
- The Buck Converter provides much greater power efficiency as DC-to-DC converters than linear regulators, which are simpler circuits that lower voltages by dissipating power as heat, but do not step up output current.





Buck Converter

The Buck Converter has the following main parts:

- MOSFET
- Gate Driver
- Capacitor
- Inductor
- Diode
- Load Resistor

Buck Converter

MOSFET

- The MOSFET works under a Pulse Width Modulation (PWM) signal, which keeps the it on or off, (works as a switch).
- MOSFETS have less power consumptions and remarkably more efficient than BJTs.
- We used an N-channel MOSFET, IRFZ44N.

Gate Driver

- We used a gate driver to maintain the required voltage between the gate and the source.
- Due to the high frequency (20KHz) and the number of times of switching, the maintain the high switching speed of MOSFET.
- We used IC IR2110 as a gate driver for the MOSFET.

Buck Converter

Inductor

- The main purpose of the inductor is to store energy and thus help the converter to stay in the continuous mode.
- The inductor slow down the current, not decrease it.
- We used a 18mH inductor (2*9mH in series).

Capacitor

- The main function of the capacitor is to filter the ripples and maintain a relatively stable output voltage across the load resistor.
- We used a 100uF capacitor.

Buck Converter

Diode

- The diode blocks any reverse-inductor current flow and the voltage across the inductor goes to zero.
- We used a 1N4007 diode as a fast switching diode, because it have fast reverse recovery time and low forward conduction losses.

Resistor

- We used a 120Ω as the load resistor which has output voltage ranges from 2.5 to 12V.
- Another resistor 10 Ω was used as a shunt resistor based on the principle of voltage divider as the Arduino board can’t afford more than 5V.

Buck Converter Calculations

Buck Converter (step down)

Summary

$V_o = V_s D$ (3) *for specified output voltage*

$I_{max} = I_L + \frac{\Delta i_L}{2} = V_o\left(\frac{1}{R} + \frac{1-D}{2Lf}\right)$ (5) *for the inductor*

$I_{min} = I_L - \frac{\Delta i_L}{2} = V_o\left(\frac{1}{R} - \frac{1-D}{2Lf}\right)$ (6) *for the inductor*

$L_{min} = \frac{(1-D)R}{2f}$ (8) for continuous current

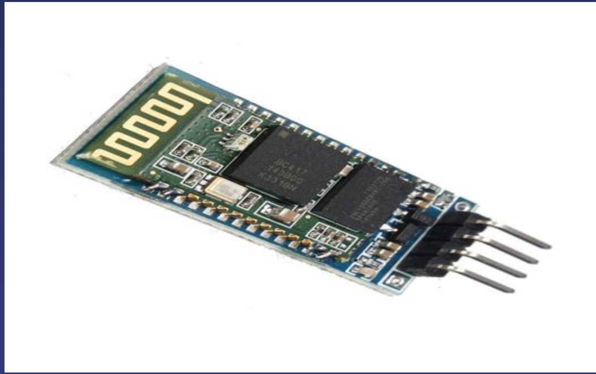
$L = \frac{V_o(1-D)}{\Delta i_L f}$ (9) *for specified peak to peak inductor current change.*

$C = \frac{1-D}{8L(\Delta V_o/V_o)f^2}$ (11) *for a specified out voltage ripple*

3

Controller

Controller



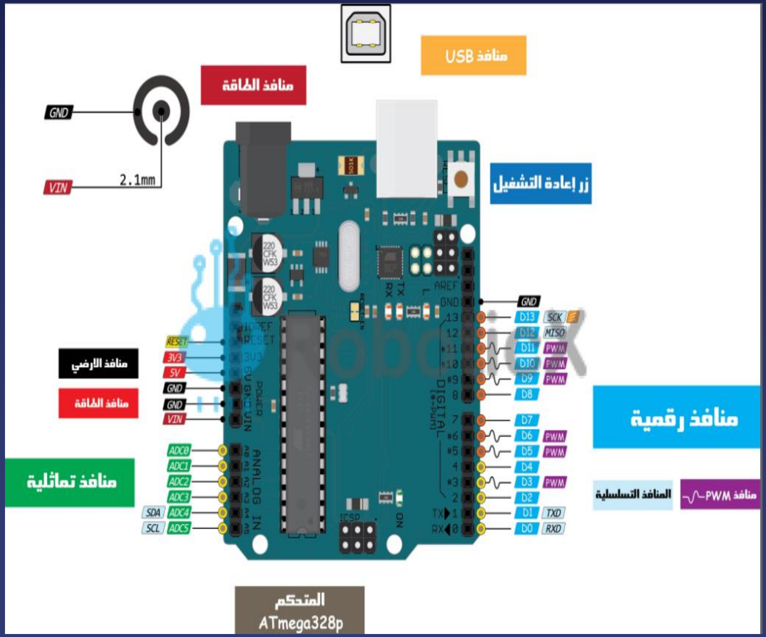
The Controller has the following two main sub-blocks…

- Arduino UNO.
- Bluetooth module HC_06 .

Why did we use Arduino Uno specifically?

We used that it has a lot of advantages that help us in the project like..

- Controller: ATmega328p .
- Operating voltage of the electrical system: 5 volts.
- Voltage (recommended): 7-12V.
- Voltage (maximum and min): 6-20 volts.
- Number of digital ports (in/out): 14.
- Ports for PWM control: 6.
- Number of analog ports (input): 6g.
- DC output to 3.3V: 50 mA.
- DC output (input/output) digital: 40mA.
- Memory space: 32 KB.
- Clock speed: 16MHz.



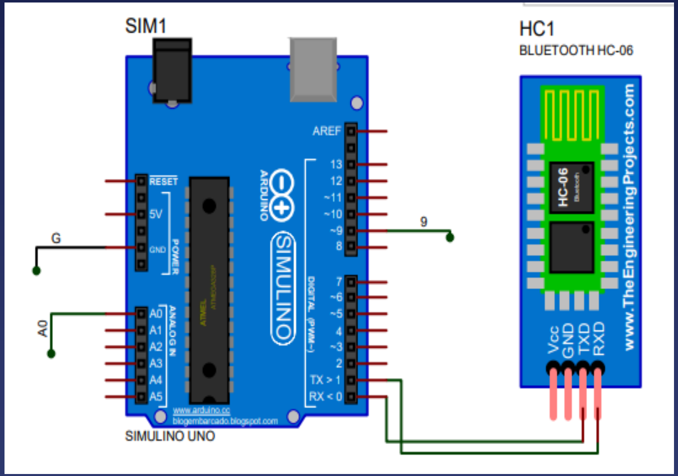
controller

How did we use the legs of the Arduino to get what we want?

- We used Pin 9 to input the frequency ,as it is one of the Pins that supported by the Timer One library, which we used to control the frequency value.. Since the default frequency with which the Arduino is made is approximately 490Hz but we changed it to the value we needed, which is 20KHz.

- We also used the Pin A0 To read the output voltage value.

Controller



- Then we burned the code that we will need on the microcontroller using the Arduino code program.
- We also used the Pins 5v & GND as a power to the Bluetooth module HC06.
- Then we use the Pins TX & RX in Arduino To connect it to the Pins TX & RX in Bluetooth module on the reverse.

4

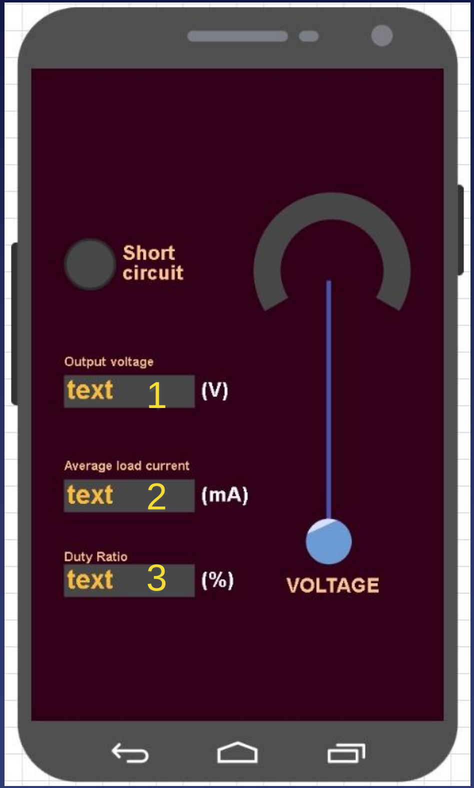
Mobile App

Remote XY

Mobile App

- To design the mobile app interface we use Remote XY site to help us make it
- We design our interface and we use Arduino c language to program our controller and app

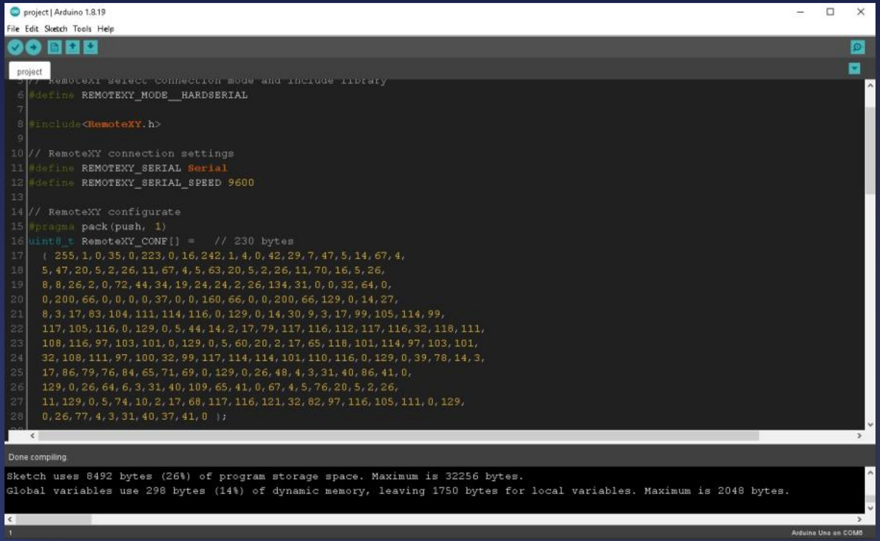
Mobile App



- **Our interface :**

1. **Slider** to control the duty cycle value.
2. **Circular bar** to show the value of slider.
3. **Led** to give us the indication of short circuit.
4. **Text 1** to show the value of the output voltage.
5. **Text 2** to show the value of the average current.
6. **Text 3** to show exactly the value of duty ratio.

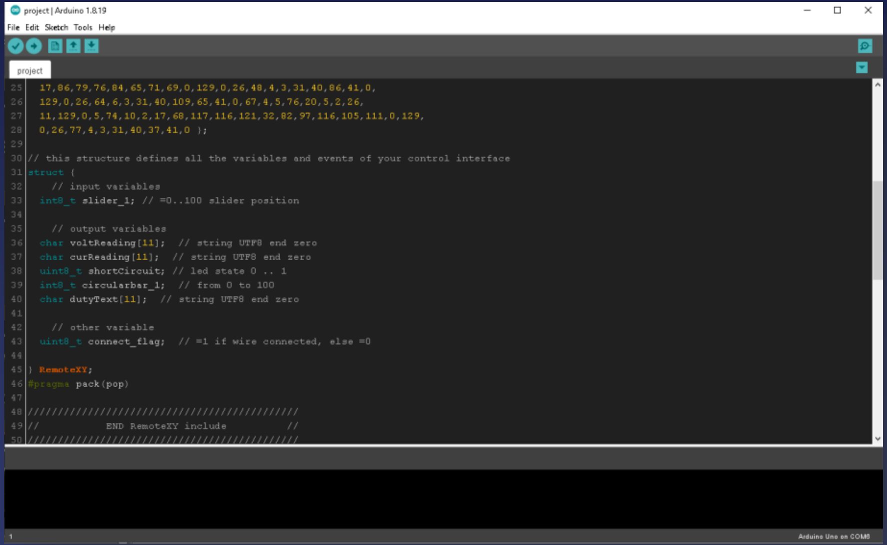
Mobile App



- <u>**The code :**</u>

The first part of our code is the initialization of the interface from remote xy site and definition of the RX and TX pins.

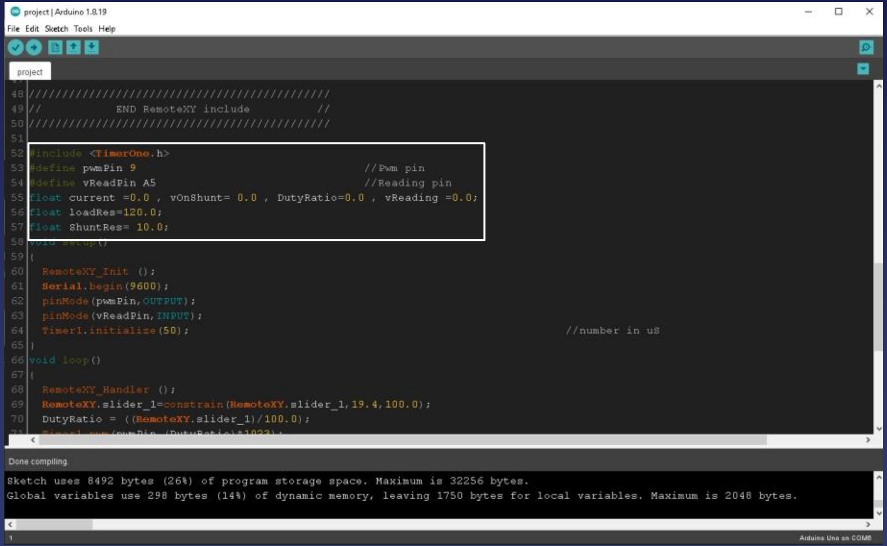
Mobile App



- **The second part** of our code is a structure called RemoteXY
- The members of the structure is the variables of the interface :

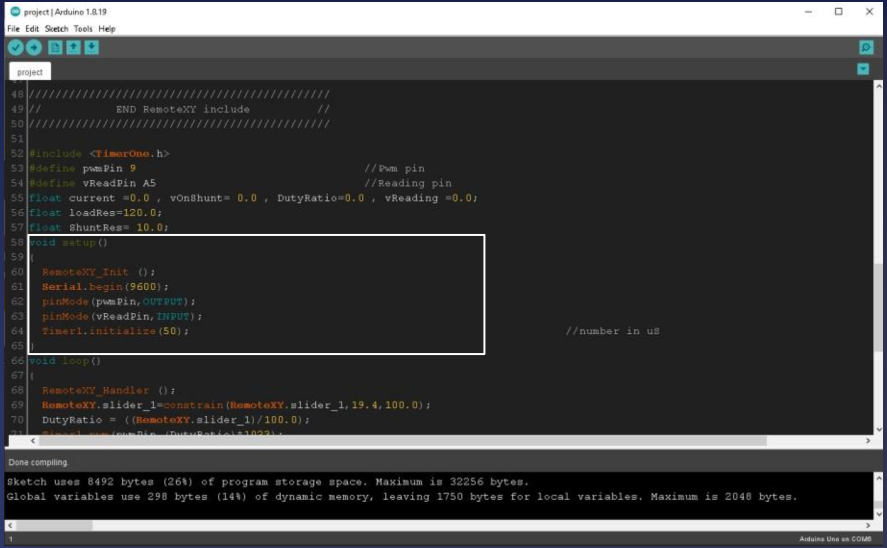
1. Slider_1 → stores values from 0 to 100.
2. voltReading → stores string value.
3. curReading → stores string value.
4. shortCircuit →stores 1/(HIGH) or 0/(LOW).
5. Circularbar_1 →stores values from 0 to 100.
6. dutyText →stores string value.

Mobile App



- **The third part** of our code is our variables , the libraries we use and the setup function :
  1. We include timer one library to help us control the frequency and duty cycle.
  2. Defining the pins.
  3. Declaring some variables which we will use.

Mobile App



In setup function :

1. RemoteXY initializer

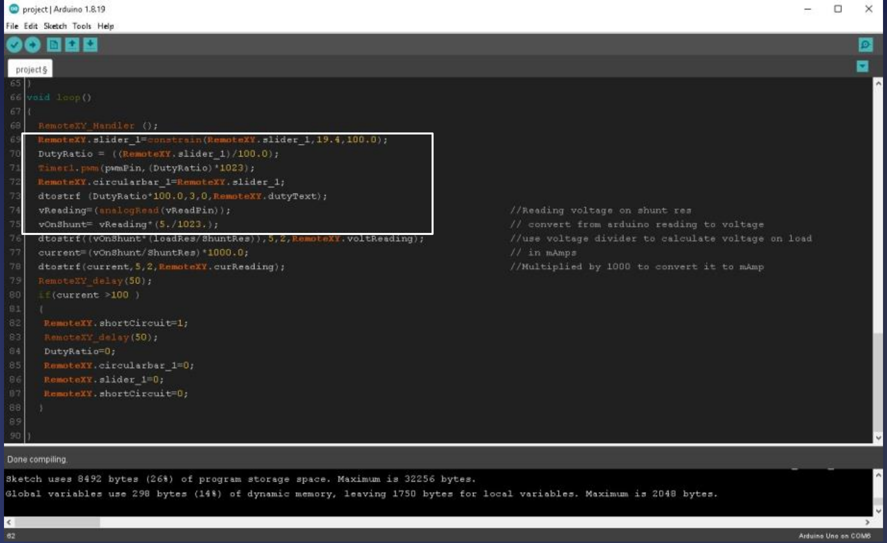
2. Setting pin mode for each pin :

Pin(9) as output pin because the pwm will generated from it

Pin(A0) as input pin because we will use it to read the voltage on the shunt resistance.

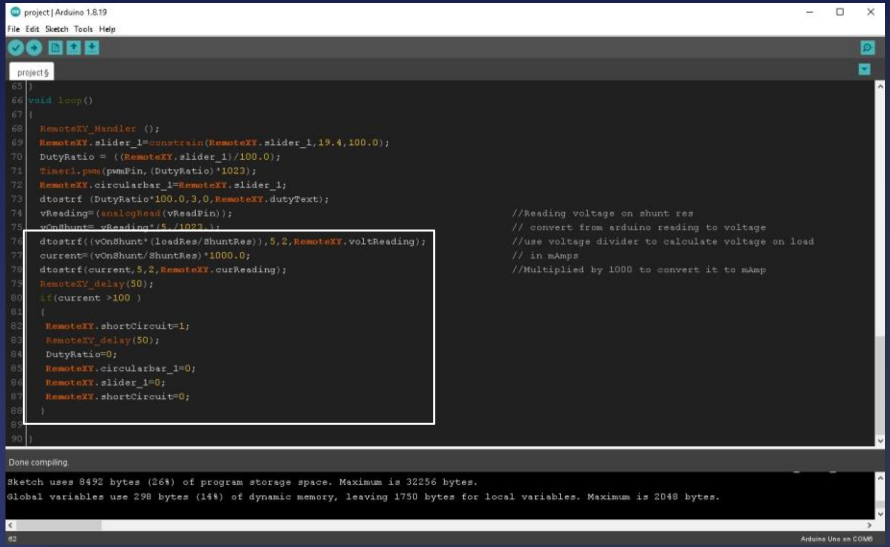
3. Setting the frequency to 20KHz by setting the time period to 50us.

Mobile App



- **The final part** of our code is the loop function :
  1. First we limit the minimum value of the slider, which represents the duty ratio, to 19.4% which give us about 2.5V on load.
  2. Next we generate pwm at pin (9) by duty ratio which the user put it into the slider.
  3. We can't assign a decimal value into a string variable so we use (dtostrf) function to help us display all decimal values in our app.
  4. We read the value on shunt resistance and multiply it by (5.0/1023) to convert it from Arduino strange reading to readable voltage value.

Mobile App



5. We use voltage divider law to convert the reading to voltage on load.
6. We calculate the current from Ohm's law.
7. We use if statement to make our short circuit indication
8. If the current goes over 100mAmps means that there is a short circuit

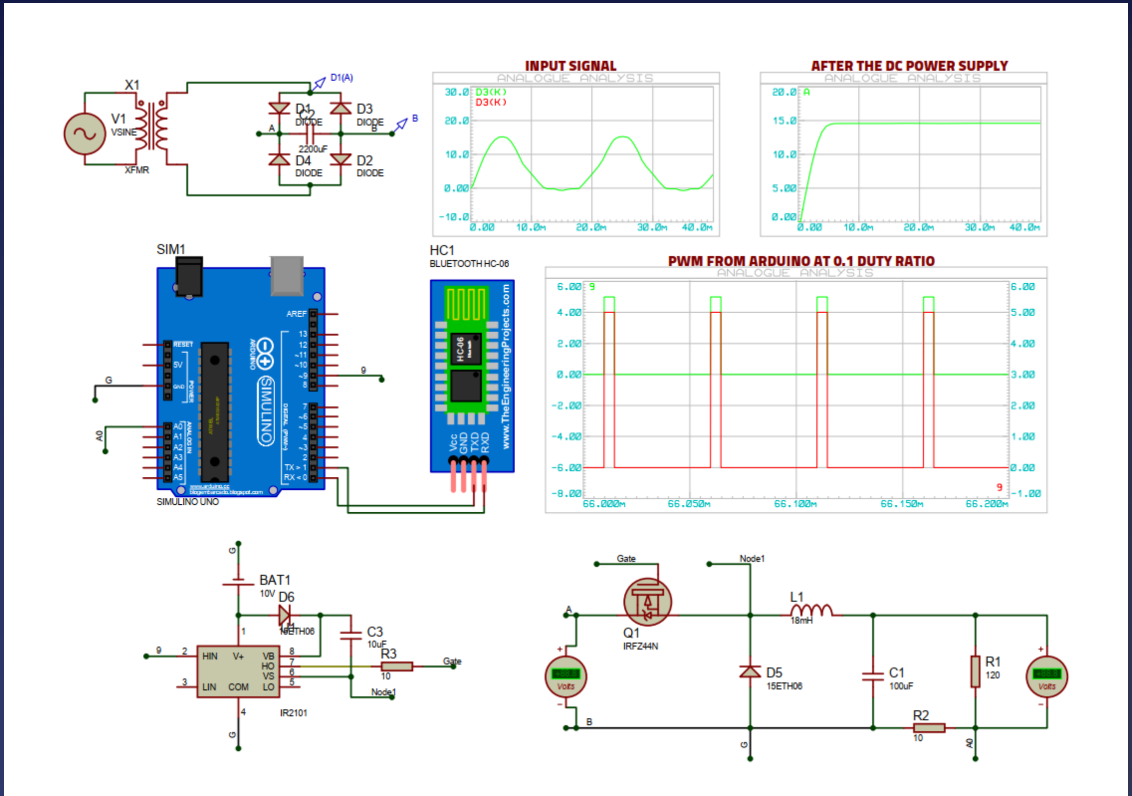
5

Summary

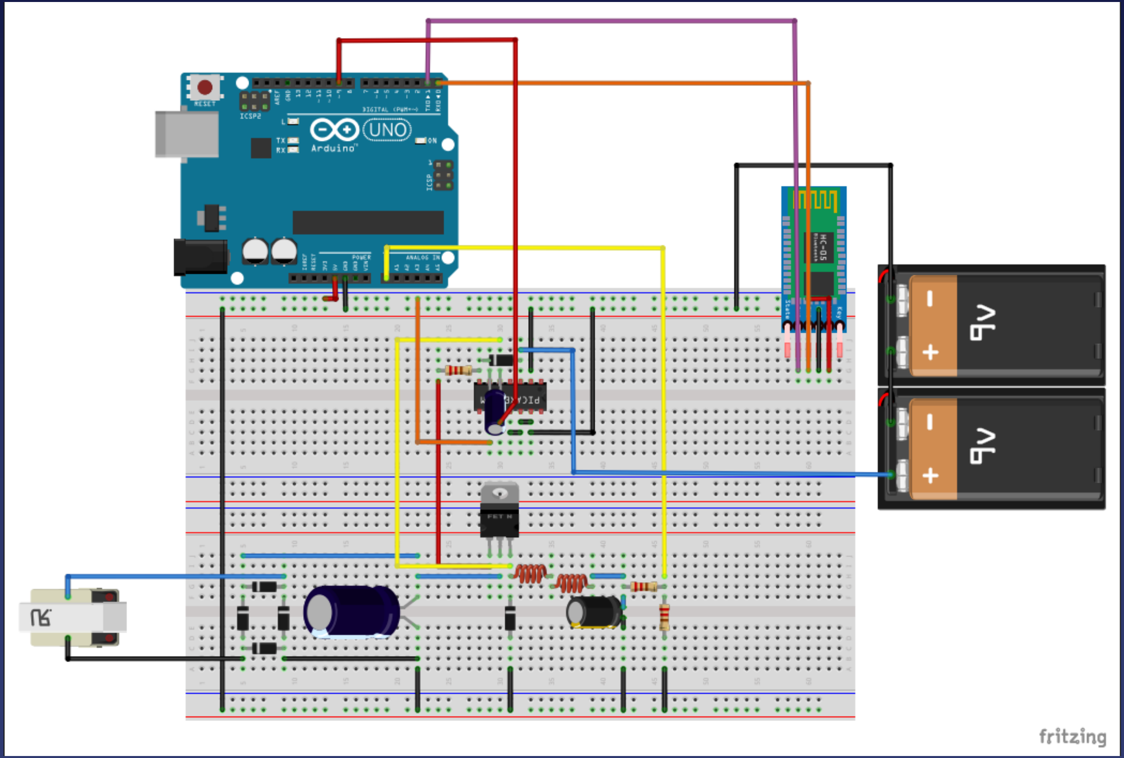
Summary

In next few slides we will put all of these parts together

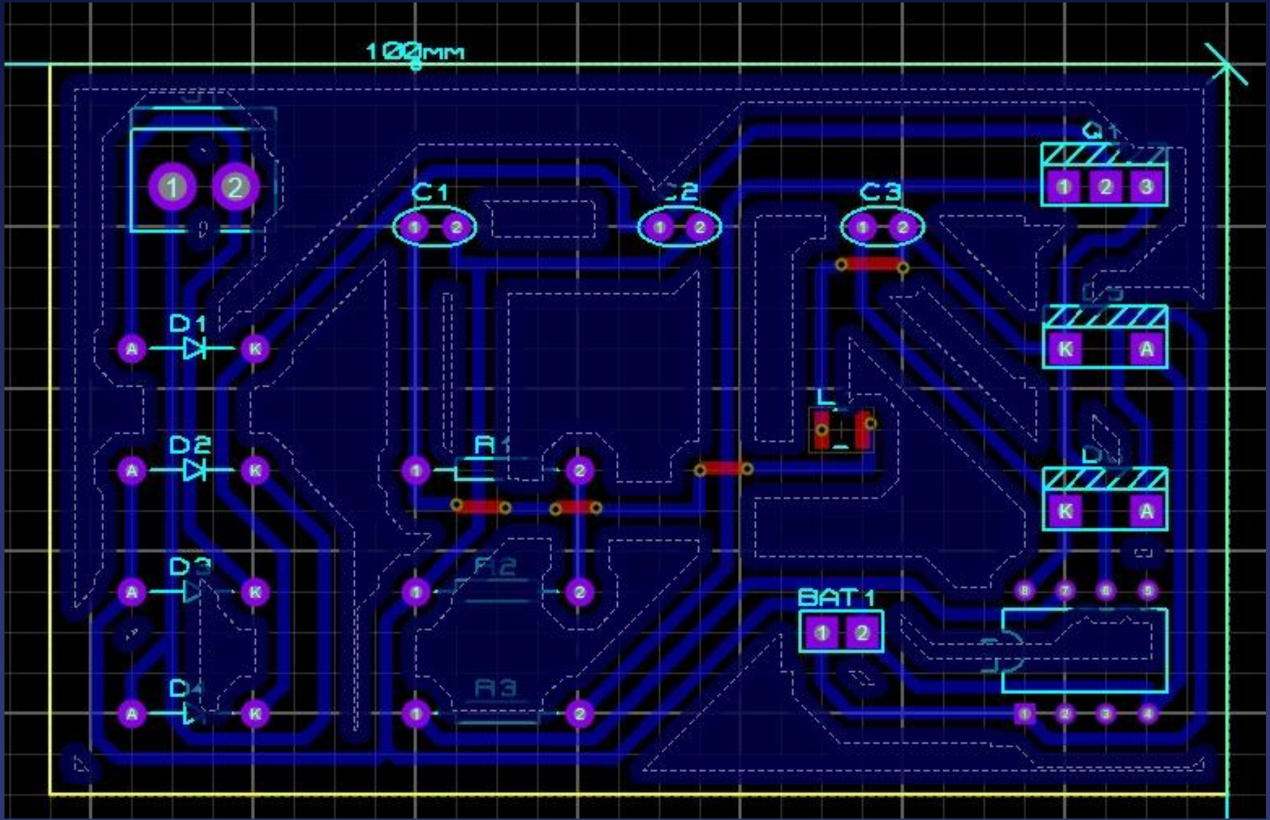
Schematic of the project (Proteus)



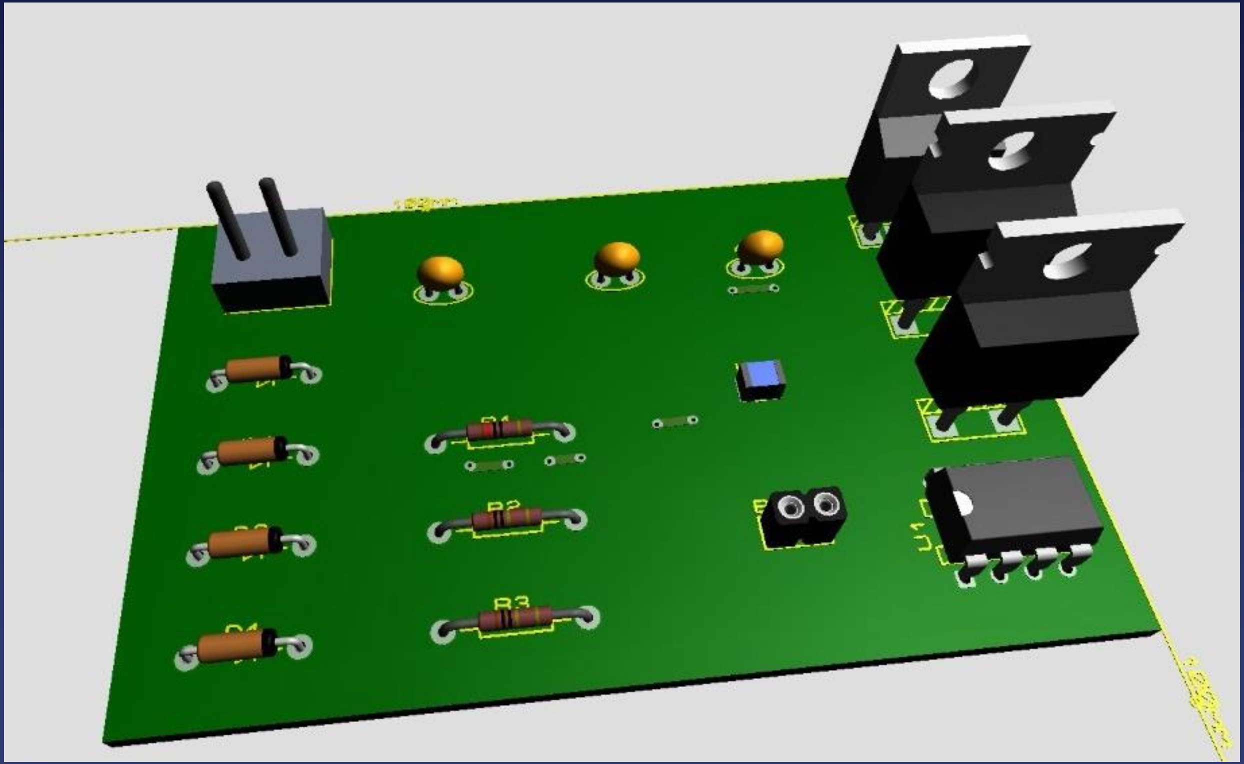
Sketch of the project



PCB design



3D of PCB design



Simulation results

Duty ratio	Output voltage (Volts)
10%	2.26
30%	5.77
50%	7.94
70%	9.46
90%	11.7

Thanks

For your attention