Ahsanullah University of Science and Technology

Project Report

Project Name : Microcontroller Based BatteryCharger and DC to controlabe AC

Course No	**: EEE-4228**
Course Title	**: Power Electronics Lab**
Semester	**: 4.2**
Section	**: A2**
Group	**:04**

Submitted By

Name	ID
Md. Rahatin Rahman Niloy	190105045
Mohsin Islam Rifat	190105046
Anika Tahsin Afia	190105047
Md. Rudro Raihan	190105048
Faisal Ariyan	180105129

Name of the Project

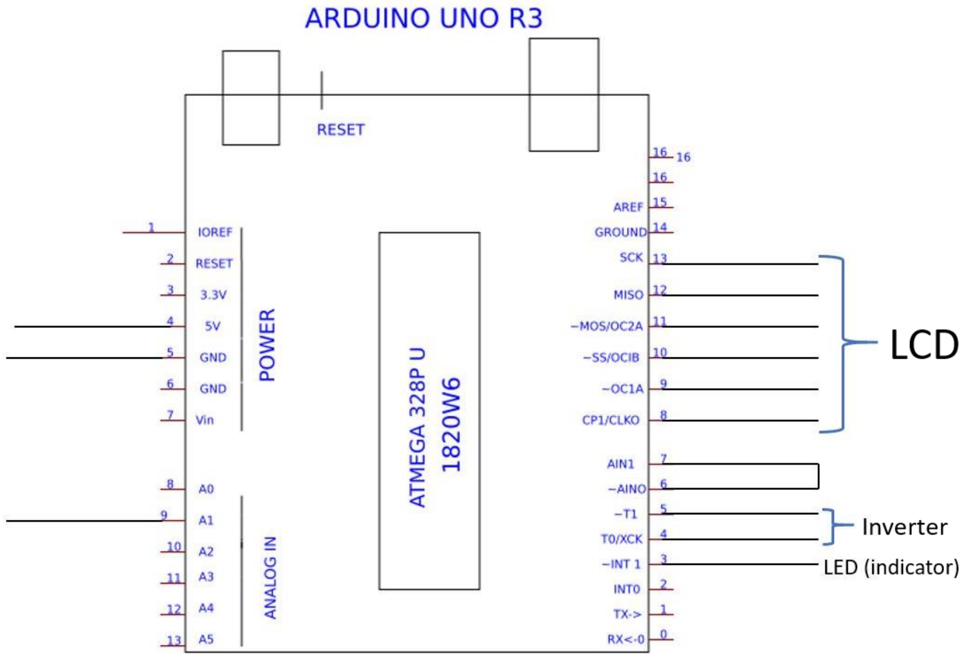
Microcontroller Based Battery Charger and DC to Controllable ACConverter.

Equipment

1. Arduino UNO
2. Liquid Crystal Display (LCD)
3. Transformer (220 volt to 6-0-6 volt)4. Diode
[1N4007]
5. Resistors (8.2kΩ, 10kΩ, 3.3kΩ, 4.7kΩ, 1kΩ, 100Ω,20k)
6. Capacitor (100μF, 0.1μF[2A104J])
7. Rechargeable Battery (8v)
8. Potentiometer/POT (500kΩ)
9. NPN Transistor [2N2222]
10. N-channel Power MOSFET [IRF540]
11. DIAC [DB3]
12. TRIAC [BT136]
13. LED (red, green)
14. Connecting wires / Jumper wires
15. Breadboard

Circuit Diagram

We have used three individual circuits: (i) Battery charger, (ii) Inverter, (iii) AC voltage controller.



Circuits are drawn below individually :-

Fig : Arduino UNO connections

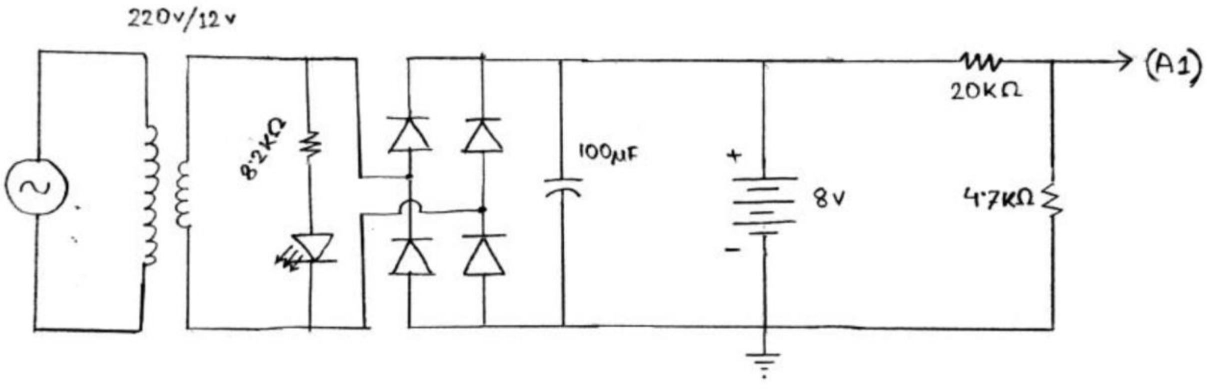


Fig (i) : Battery charger circuit

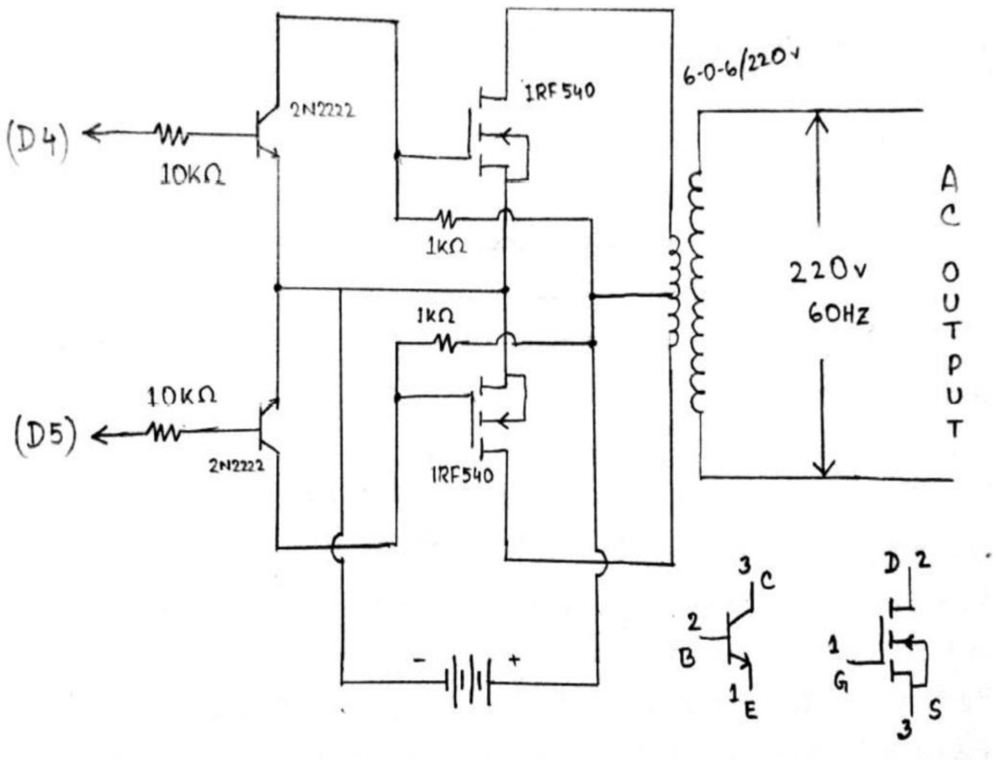


Fig (ii) : Inverter circuit

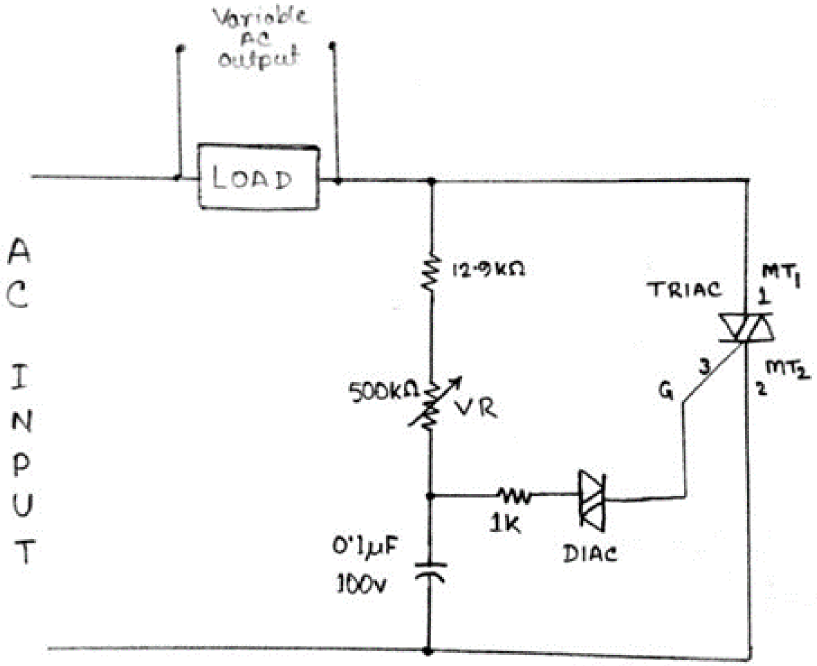
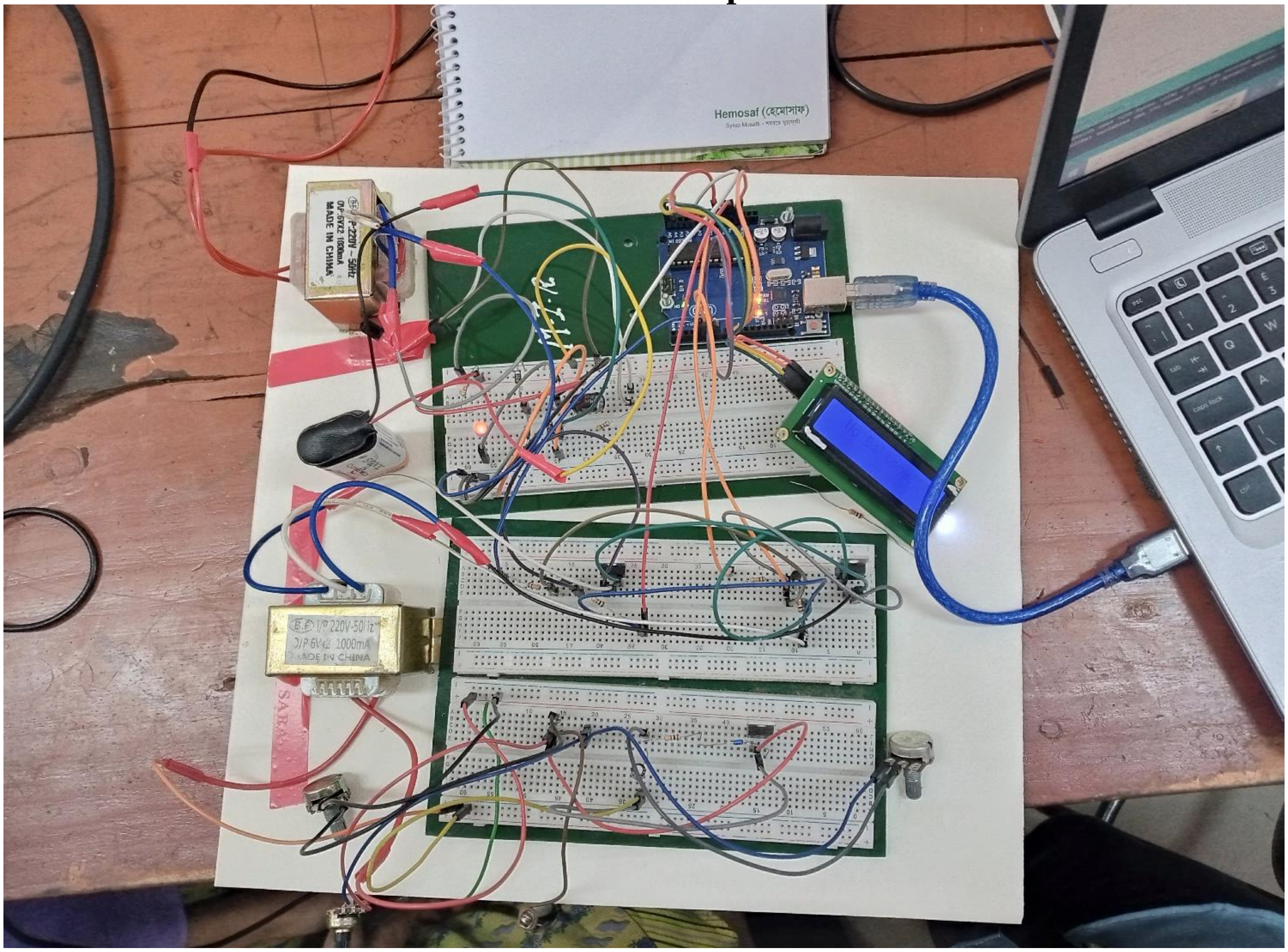


Fig (iii) : AC voltage controller circuit

Circuit setup



Arduino Code

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 13, 8, 9, 10, 11);
int pwm = 6;
// pwm ~6 can generate 61.03515625 Hz int pwmRead = 7;
int pulse1 = 4; int pulse2 = 5;
int charging = 3;
// battery charging indicator LED
float R1 = (9.76+9.75)*1000;
 // resistor1 measured value float R2 = (4.64)*1000;
// resistor2 measured value
int refreshTime = 50;
// Refreshing unsigned long mils, timer;
void setup()
{
pinMode (pwm, OUTPUT);
 pinMode (pwmRead, INPUT);
pinMode (pulse1, OUTPUT);
pinMode (pulse2, OUTPUT);
pinMode (charging, OUTPUT);
TCCR0B = TCCR0B & 0b11111000 | 0x05;
 // setting for 61.035 Hz pwm Serial.begin (9600);
```

```
lcd.begin(16, 2);
}
void loop()
{
float x, y, z, battVolt;
analogWrite (pwm, 127);
// 127 for 50% duty cycle z = digitalRead (pwmRead);
{
if (z == HIGH)
{
digitalWrite (pulse1, HIGH); digitalWrite (pulse2, LOW);
}
else
{
digitalWrite (pulse1, LOW); digitalWrite (pulse2, HIGH);
}
}
{

x = analogRead (A1);
y = x*(5.13/1023.0);
// Arduino +5v pin out +5.13v battVolt = y/(R2/(R1+R2));
{
mils = millis();
if (mils - timer >= refreshTime)
{
timer = mils;
{
if (battVolt <= 0.3)

{
digitalWrite (charging, LOW);
Serial.print ("No Battery !!");
Serial.println ("");
lcd.clear(); lcd.setCursor (2, 0);
lcd.print ("No Battery!");
}
else if (battVolt > 0.3 && battVolt <= 9.2)
{
digitalWrite (charging, LOW);
Serial.print ("Battery Voltage= ");
Serial.print (battVolt);
Serial.print (" volt");
Serial.println ("");
lcd.clear();
lcd.setCursor (0,0);
lcd.print ("Battery Voltage");
lcd.setCursor (0, 1);
lcd.print (battVolt);
lcd.setCursor (6, 1);
lcd.print ("volt");
```

```
}
else if (battVolt > 9.2 && battVolt <= 12.0)
{
digitalWrite (charging, HIGH);
Serial.print ("Plugged In, and");
Serial.println ("");
Serial.print ("Charging           ");
Serial.println ("");
Serial.println (""); lcd.clear();
 lcd.setCursor (0,0);
lcd.print ("Plugged In, and");
lcd.setCursor (0, 1);
lcd.print ("Charging    ");
}
else if (battVolt > 12.0 && battVolt <= 25.0)
{
digitalWrite (charging, LOW);
Serial.print

("Battery Voltage= ");
Serial.print (battVolt);
Serial.print (" volt");
Serial.println ("");
lcd.clear();
lcd.setCursor (0,0);
lcd.print ("Battery Voltage");
lcd.setCursor (0, 1);
lcd.print (battVolt);
lcd.setCursor (6, 1);
 lcd.print ("volt");
}
else if (battVolt > 9.2 && battVolt <= 12.0)
{
digitalWrite (charging, HIGH);
Serial.print ("Plugged In, and");
Serial.println ("");
Serial.print ("Charging           ");
Serial.println ("");
Serial.println (""); lcd.clear();

lcd.setCursor (0,0);
lcd.print ("Plugged In, and");
 lcd.setCursor (0, 1);
lcd.print ("Charging    ");
}
else if (battVolt > 12.0 && battVolt <= 25.0)
{
digitalWrite (charging, LOW);
 Serial.print ("Plugged In");
Serial.println ("");
Serial.print ("No Battery !!");
```

```
 Serial.println ("");
Serial.println ("");
lcd.clear();
lcd.setCursor (0,0);
 lcd.print ("Plugged In");
lcd.setCursor (0, 1);
lcd.print ("No Battery!!");
}
}
} } } }
```

Circuit Operation

Circuit (i): Battery Charger Circuit

- An AC supply (220 v) is connected to a step down transformer’s primary sideand from the secondary side it gives approximate 12 v AC voltage.
- A LED as an indicator is connected with the secondary site of the transformerby a resistor.
- There is a bridge rectifier connected with the 12 volt AC and it gives 12 voltDC.
- The capacitor removes the ripple voltage.
- This DC is connected to a rechargeable battery and the battery will berecharged if there is a supply (AC) to the transformer’s primary side.
- To show the battery voltage into the LCD display (and serial monitor) we usethe ARDUINO Analog Read feature.
- Battery voltage goes to the Arduino Analog Pin (A1) via two resistors (usingVDR), Arduino reads the voltage and monitors into the LCD.
- In Arduino program we have used the equations,

$y=x*(5.0/1023.0)$ $battVolt=y/\{R2/(R1+R2)\}$

Circuit (ii): DC to AC Converter / Inverter Circuit

- In our country we use 50~60 Hz AC frequency.
- Arduino UNO’s Digital Pin number ‘6’ can generate a (square wave) PWM of62500 Hz frequency, where cycle length is 256 (0~255).
- But, by the help of a divisor, we can generate (62500/1024 =) 61.035 Hz. Forthis purpose, following setting is needed to use:

TCCR0B = TCCR0B & 0b11111000 | 0x05

{Reference: http://playground.arduino.cc/Main/TimerPWMCheatsheet}

Here the duty cycle 50% is achieved, in the Arduino code Analog Write(255*50% ≈) 127.

- This PWM signal is read by a digital pin of Arduino and another two digital pins are giving two individual pulses, while pulse(1) is HIGH, pulse(2) is LOW and vice versa. [Fig(a)]
- There are two NPN transistors, which are controlled by the Arduino pulses.
- The operation is like this- while transistor(1) is forwardly biased therefore current conducts from Collector to Emitter; transistor(2) is reversely based, so no current conducts.
- Transistors’ Collectors are connected to two N-channel Power MOSFETs. And the Emitters are Grounded.

- We know that, the n-MOSFET is switched ‘ON’ if the Gate voltage is higher than its threshold voltage (say, VGS = 1); switched ‘OFF’ if the Gate voltage is lower than its threshold voltage (say, VGS = 0).
- Here, first half period of one cycle, when transistor(1) is in forward biased, there is no MOSFET(1) Gate voltage because the current is directly bypassed to the ground via transistor(1) and the MOSFET(1) goes OFF state.
- During this period transistor(2) is in reverse biased and MOSFET(2) goes ON state.

- Next half period of that cycle, same but opposite operation occurs.
- This allows the DC voltage to be produced across the primary of the transformer at alternate intervals. It is 6v AC.
- The 6v AC signal across the primary of the transformer is then stepped up to 220v AC signal across the transformer secondary.
- Thus we get converted 220 volt AC from 6 volt DC. [Fig(b,c)]

Circuit (iii): AC Voltage Controller Circuit

- The inverter’s output voltage (220v AC) can be controlled by an AC controller.
- An AC voltage controller contains a DIAC, a TRIAC, a Capacitor, resistors and a Potentiometer (POT).
- The DIAC is a full-wave or bi-directional semiconductor switch that can be turned on in both forward and reverse polarities. The TRIAC is also bi-directional, just like a DIAC, but it needs a Gate signal to conduct current.

LCD monitoring overview

If battery (and charge supply) is not connected to the circuit-

No Battery!

If Battery is Connected

Battery Voltage
X.XX volt

If charger is ON and battery is charging-

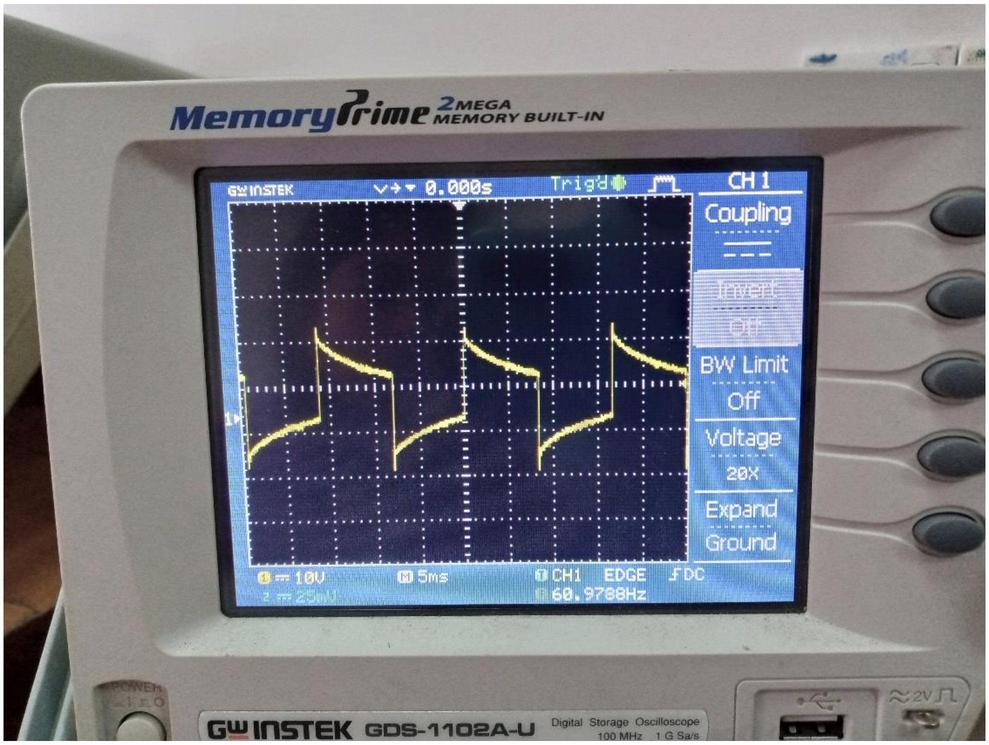
Plugged In, and
Charging....

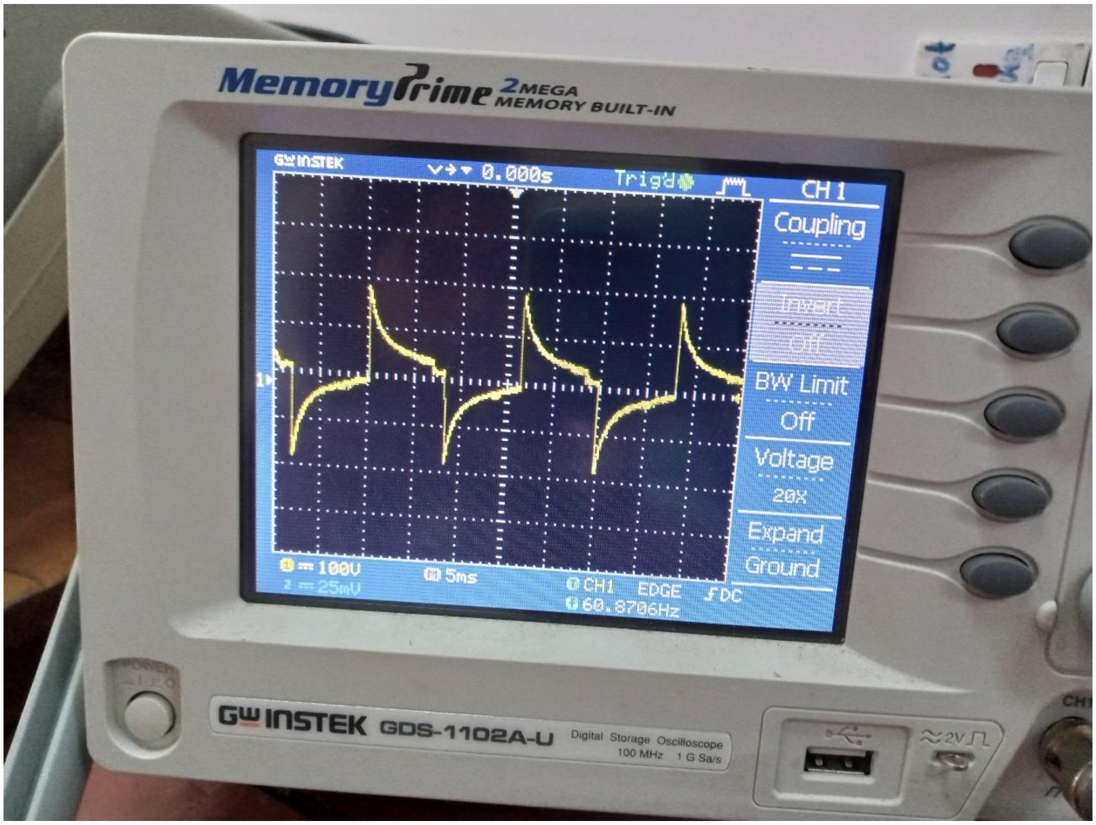
- If charger is ON but battery is disconnected from the circuit-

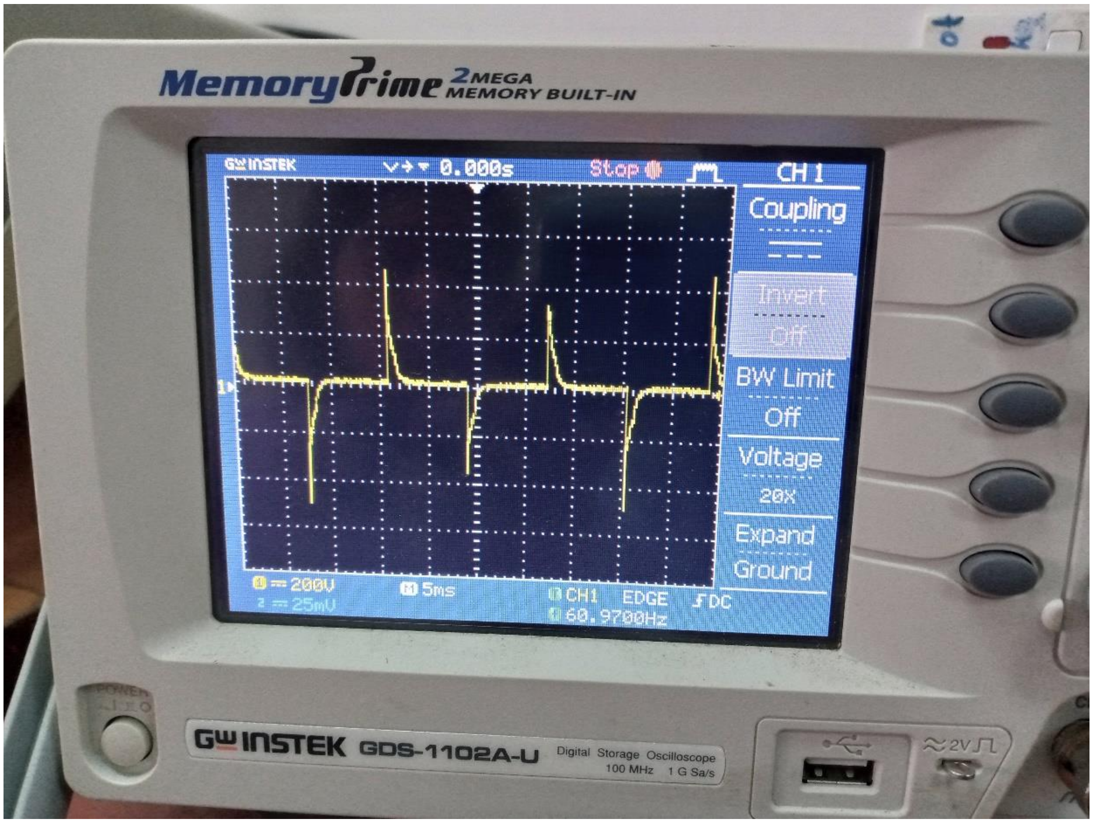
Plugged In
No Battery!!

<u>**Some Pictures as Document :**</u>

60 Hz pulse generated into two digital pins ofArduino







Limitations

In our project we have designed Microprocessor (Arduino Uno) based Battery Charger, Inverter and AC Voltage Controller with some limitations. The battery charger circuit have some features like it can monitor the battery level and also can show it in the LCD display, but the limitation is when battery is fully charged, the charging circuit can’t disconnect the battery from the supply.

The second portion of our project is Inverter (DC to AC converter) circuit. An inverter circuit is designed which can provide AC output up to 230 volts but the limitation is the output is square wave rather than sine wave. The square wave has harmonic components where the pure sine wave has one basic frequency. Harmonics in the electrical power distribution system combine with the fundamental frequency (50Hz ~ 60Hz) supply to create distortion of the voltage and/or current waveforms. This distortion creates a complex waveform made up from a number of harmonic frequencies which can have an adverse effect on electrical equipment and power lines. So, further modification is needed to modify the AC square wave as a pure sine wave for the safety of electrical devices. Thus, AC variable circuit is designed successfully without any complication.