This part’s goal is the calculating the current , remaining time of SOC and percentage and showing to the LCD. The system contains the two resistors to understanding formulas in this code. Resistor values are 5mOhm and 10KOhm. Also system uses the INA168 for the calculating the current.

Arduino Uno’s limit is up to 5 volts so the system have to use 5/1023 formula to measure voltage. This system contains 2 input one of them is for the input current and other is for output current and input current connecting to A0 pin of Arduino and output current connecting to the A1 pin of Arduino. There is only one LCD to show these measurements. This program calculate the current and voltage first by the critical calculations using the INA 168 current sense amplifier. The main issue in this program is integer overflow error. This is because program contains so many big values and formulas in loops. Program uses the uint32\_t to solve this problem. Uint32\_t is a numeric type that guarantees 32 bits, the value is unsigned, meaning that the range of values goes from 0 to 232 – 1.

In last part, program measures the remaining time of SOC and percentage of SOC and displays to the LCD screen.

#define potpin1 A0 //connecting A0

#define potpin2 A1 //connecting A1

float deger1=0;// değer tanımı

float deger2=0;

float SOC = 3250 \* 3600 \*6;

float SOC\_STATUS = SOC;

#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 10, 5, 4, 3, 2);

void setup()

{

Serial.begin(9600);

Serial.println("Reading Value");

}

void loop()

{

deger1 = analogRead(potpin1);

float voltage = deger1 \* (5.0 / 1023.0);

if ( deger1!=0){

float input\_current = voltage/(0.005\*10000\*(200\*0.000001));

// saat hesaplamaları vs vardı eskiden

uint32\_t period = 500L; // 500 mseconds

for( uint32\_t tStart = millis(); (millis()-tStart) < period; ){

input\_current+=input\_current;} // 500msaniye olayı bura

SOC\_STATUS = SOC\_STATUS - input\_current;

float hour = SOC\_STATUS/3600;

float minute = (SOC\_STATUS - hour\*3600)/60;

float sec = SOC\_STATUS - hour\*3600 - minute\*60;

lcd.clear();

lcd.setCursor(0, 0);

lcd.print(hour +String("hour") + minute + String("min")+ sec + String("sec") );

float percentage = SOC\_STATUS/ SOC \*100;

lcd.setCursor(0, 1);

lcd.print(String("battery") + percentage + "%");

}

else {

deger2 = analogRead(potpin2);

float voltage2 = deger2 \* (5.0/1023.0);

float output\_current = voltage2/(0.005\*10000\*(200\*0.000001));

uint32\_t period = 500L; // 500 mseconds

for( uint32\_t tStart = millis(); (millis()-tStart) < period; ){

output\_current+=output\_current;} // 500msaniye olayı bura

SOC\_STATUS = SOC\_STATUS - output\_current;

float hour = SOC\_STATUS/3600;

float minute = (SOC\_STATUS - hour\*3600)/60;

float sec = SOC\_STATUS - hour\*3600 - minute\*60;

lcd.clear();

lcd.setCursor(0, 0);

lcd.print(hour +String("hour") + minute + String("min")+ sec + String("sec") );

float percentage = SOC\_STATUS/ SOC \*100;

lcd.setCursor(0, 1);

lcd.print(percentage);

lcd.print(String("battery") + percentage + "%");

}

}