```
/*------Libraries--------*/

#include <Wire.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <LiquidCrystal.h>
#include <SD.h>

/*-----( Declare Constants )-----*/

#define ONE_WIRE_BUS 29/*-(Connect to Pin 2 )-*/
const int DS1307  = 0x68;// Address of DS1307 see data sheets
const int chipSelect = 53;
/*-----( Declare objects )-----*/
LiquidCrystal lcd(22, 24, 26, 25, 27, 28);//lcd pinleri set ediliyor
  OneWireourWire(ONE_WIRE_BUS);
  DallasTemperaturesensors(&ourWire);

/*-----( Declare Variables )-----*/

int LDR_Pin = A14;
int analogInPin = A12;
int  batMonPin   =   A11;
int mpptmon = A10;
int sensorValue=0;
int outputValue=0;
int tempPin = 0;

float ds18b20_temp;

byte second = 0;
byte minute = 27;
byte hour = 23;
byte weekday = 0;
byte monthday = 0;
byte month = 01;
byte year = 14;
//---------------------------------------------------------------------------
//SETUP İS START THERE
void setup()
{

  lcd.begin(16, 2);//16*2 lik lcd olduğu belirleniyor 16 sütun 2 satır olarak
  lcd.print("Sistem Aciliyor");//lcd ye başlangışta system on yazılıyor

  delay(2000);

  Wire.begin();
  Serial.begin(9600);

  delay(1000);

  pinMode(33, OUTPUT);

  sensors.begin();

  // see if the card is present and can be initialized:
  if (!SD.begin(chipSelect)) {
    Serial.println("Card failed, or not present");
    lcd.setCursor(0, 0);
    lcd.print("Kart Yok!");
    // don't do anything more:
    return;
  }


  lcd.setCursor(0, 0);
  lcd.print("Kart Yukleniyor");  Serial.println("Card Initialized.");
  delay(2000);
  lcd.clear();
}

// SETUP İS OVER
//------------------------------------------------------------------------
// internal tep. sensor value

int tempC=0;

//battary voltage values

int   val   =   0;// variable for the A/D value
float pinVoltage = 0;// variable to hold the calculated voltage
float batteryVoltage = 0.0;
float ratio  =  3.018;// Change this to match the MEASURED ration of the circu
float bat=0;
int ort=0;

//mppt voltage values

int val1;
float pinVoltage1 = 0;
float ratio1 = 3.2;
float mpptVoltage =0.0;

//ldr voltage values
int ldrvout=0;
int lux=0;
//LOOP function start-----------------------------------------------
void loop()
{
  //WRİTİNG DATE AND TİME



 lcd.setCursor(0,0);
 printTime();
 delay(5000);

//BATARY VOLTAGE CALCULATING
   val =analogRead(batMonPin);    // read the voltage on the divider

    pinVoltage  =  val  *  0.00488;//  5V / 1024 -> BECAUSE MİCROPROCESSOR H

   batteryVoltage = pinVoltage * ratio;

  Serial.print("Battary Voltage: ");
  Serial.println(batteryVoltage);
      delay(1000);
//------------------------------------------------------------------------
//MPPT VOLTAGE CALCULATED------------------------------------------------

   val1 =analogRead(mpptmon);
  pinVoltage1 = val1 * 0.00488;//  5V / 1024 -> BECAUSE MİCROPROCESSOR HAS 1
  mpptVoltage = pinVoltage1 * ratio1;

  Serial.print("Voltage: ");
  Serial.println(mpptVoltage);
//-------------------------- BATARY PERCENTAGE FOR LCD VİEWİNG---------------
bat=batteryVoltage-10;
 ort=(bat*100)/4.4;
  lcd.clear();
  lcd.setCursor(13, 1);
  if(ort<=0) {
    lcd.print("%0");
    lcd.print(ort);
  }
  else {
    lcd.print("%");
    lcd.print(ort);
  }
//-----------------------
  Serial.println();
  //-------DS18B20 TEMP. SENSOR VALUES GETTİNG

  sensors.requestTemperatures(); // Send the command to get temperatures
  lcd.setCursor(0, 1);
```

```
    lcd.print("Sicaklik=");
    int ds18b20_temp = sensors.getTempCByIndex(0); // GETTİNG TEMP. VALUE İN DS1
    lcd.print(ds18b20_temp);

    //------------------CALCULATINGCURRENT---------------------------------------

  sensorValue=analogRead(analogInPin);
    outputValue = ( ((long)sensorValue * 5000 / 1024) - 500 ) * 1000 / 133;// (


        Serial.print("sensor = " );
        Serial.print(sensorValue);

        Serial.print("\t Current (ma) = ");
        Serial.println(outputValue);

    //------------------ GETTİNG İNTERNAL TEMP. VALUE FROM LM35-----------------

     tempC  =analogRead(tempPin);           //read the value from the sensor
    tempC=(((tempC / 1023.0) * 5.0) * 100.0);//tempC = (5.0 * tempC * 100.0)/102
   tempC=tempC-273;

     Serial.print("Ic Sicaklik Degeri = ");
    Serial.print((byte)tempC);             //send the data to the computer
    Serial.println(" Derece ");
   //FAN  CONTROLLİNG  PART
    if(tempC<28)  //İF TEMP. OVER 28 CELCİUS THAN OPEN FAN
      digitalWrite(33, LOW);

    else
      digitalWrite(33, HIGH);



    //--------GETTİNG LİGHT VALUE FROM LDR------------------------------

    int LDRReading =analogRead(LDR_Pin);  // GETTİNG LDR VALUE
      ldrvout=LDRReading*0.00488;
     lux=(2500/ldrvout-500)/10;// CONVERTİNG LUX VALUE

    Serial.print("Lux =");
    Serial.println(outputValue);

    lcd.setCursor(0, 0);
    lcd.print("Isik=");
    lcd.print(LDRReading);

    // ------ SD CARD WRİTİNG -----------------------------------------------
    String dataString ="";
     dataString +=String(ds18b20_temp);
```

```
        dataString +="     ";
         dataString +=String(LDRReading);
        dataString +="     ";
         dataString +=String(outputValue);
        dataString +="     ";
        //  dataString += String(mpptVoltage);
        dataString +="     ";
        //  dataString += String(batteryVoltage);


        File dataFile =SD.open("datalog.txt", FILE_WRITE);

        // if the file is available, write to it:
        if (dataFile) {
          dataFile.println(dataString);
          dataFile.close();
          // print to the serial port too:
          Serial.println(dataString);
          }
        // if the file isn't open, pop up an error:
        else {
          Serial.println("error opening datalog.txt");
         }
       // ----------sd card writing over ------------------------------------------

        delay(300000); // setup datalogging time range (5min. = 300.000ms that mean

      }

      // ------------LOOP FUNCTİON OVER -------------------------------------------

      //------------- GETTİNG TİME VALUE FROM DS1307 ------------------------------

      byte decToBcd(byte val) {
        return ((val/10*16) + (val%10));
      }
      byte bcdToDec(byte val) {
        return ((val/16*10) + (val%16));
      }
      // DS1307 USİNG I2C PROTOCOL FOR COMMUNİCATİON THAT MEANS SDA AND SCL PİNS USE

      byte readByte() {
        while (!Serial.available()) delay(10);
        byte reading = 0;
        byte incomingByte =Serial.read();
        while (incomingByte !='\n') {
          if (incomingByte >='0' && incomingByte <='9')
            reading = reading * 10 + (incomingByte -'0');
          else;
```

```
    incomingByte =Serial.read();
  }
  Serial.flush();
  return reading;
}

//-----------------GETTİNG TİME VALUE FUNCTİON OVER ---------------------

//-----------------TİMEPRİNTİNGFUNCTİON--------------------------------

void printTime() {
  char buffer[3];
  const char* AMPM = 0;
 readTime();

  Serial.print(monthday); lcd.print(monthday);
  Serial.print("."); lcd.print(".");

  if(month<=9) {
    Serial.print("0");
    lcd.print("0");
    Serial.print(month);
    lcd.print(month);
  }

  else {
    Serial.print(month); lcd.print(month);
  }
  Serial.print("");
  Serial.print(".20"); lcd.print(".20");
  Serial.print(year); lcd.print(year);
  Serial.print(" "); lcd.print(" ");
  lcd.setCursor(0, 1);
  if (hour<=9) {
    Serial.print("0");lcd.print("0");
    Serial.print(hour);lcd.print(hour);
  }

  else
    Serial.print(hour); lcd.print(hour);
  Serial.print(":");lcd.print(":");
   sprintf(buffer,"%02d",  minute);
  Serial.print(buffer);

  lcd.print(minute);
  Serial.println(AMPM);

}
```

```
// ---------------PRİNTİNG FUNCTION İS OVER---------------------------

//----------------READFUNCTİON--------------------------------------
void readTime() {
  Wire.beginTransmission(DS1307);
  Wire.write(byte(0));
  Wire.endTransmission();
  Wire.requestFrom(DS1307, 7);
  second = bcdToDec(Wire.read());
  minute = bcdToDec(Wire.read());
  hour = bcdToDec(Wire.read());
  weekday = bcdToDec(Wire.read());
  monthday = bcdToDec(Wire.read());
  month = bcdToDec(Wire.read());
  year = bcdToDec(Wire.read());
}

//----------------READ FUNCTİON OVER --------------------------------

//DS1307 RTC (Real Time Clock chip programming before that program just once
//afterthat this program just doing to get date value than reading and printi
```