**Topic: Data Logging System Using Arduino Uno**

**(CURRENT AND POWER MEASUREMENT OF HOUSEHOLD FRIDGE DATA LOGGING SYSTEM)**

**Introduction:**

Using an Arduino UNO, this assignment illustrates the current and power consumption of the typical household refrigerator. The appliance, which is vital, consumes power constantly. Therefore, it is beneficial and a good assignment to evaluate the current and power used by a typical household refrigerator in order to find ways to lower its consumption level and conserve energy.

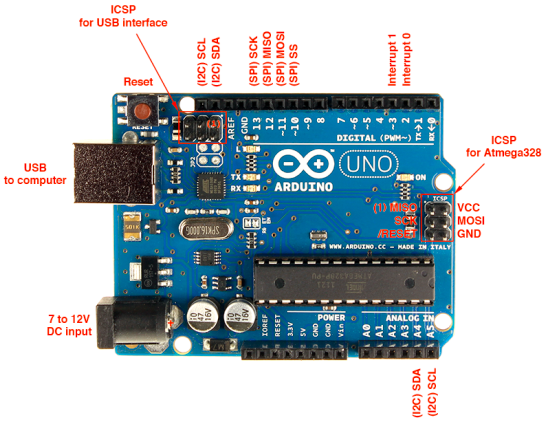
The following goals are part of this assignment:

* To gather sensor data
* To save it on an SD card
* To evaluate the data
* To provide some suggestions/advice on how to reduce it

**Bill of Materials:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **BOM level** | **Part name** | **Description** | **Quantity** | **Cost** |
| 1 | CURRENT AND POWER MEASUREMENT OF HOUSEHOLD FRIDGE DATA LOGGING SYSTEM | | | |
| 2 | Arduino Uno Rev3 | https://eph.com.pk/product/arduino-uno-r3-with-usb-cable/ | 1 | 2000 |
| 3 | CT Sensor ACS712 30A | https://images.app.goo.gl/7JVoST5DhPwy8Hgb9 | 1 | 450 |
| DS3231 RTC Module | https://epro.pk/product/ds3231-rtc-real-time-clock-in-pakistan/ | 1 | 900 |
| Micro SD Card Module | https://www.daraz.pk/products/microsd-card-module-for-arduino-i235815995.html | 1 | 180 |
| 4 | USB Cable Adapter (for Arduino) | https://www.amazon.com/Adapter-Arduino-Elliptical-Recumbent-Positive/dp/B07ZM46WKF | 1 | 300 |
| SD Card (SanDisk Ultra microSDXC UHS-1 16GB) | https://images.app.goo.gl/8wJooapdbi4AFwBK8 | 1 | 800 |
| RTC Cell | https://images.app.goo.gl/teUgJfZ9HRS5sY4W7 | 1 | 300 |
| 5 | Extension Cable (Female-Male) | https://images.app.goo.gl/4f8qJwKkoipEBQbCA | 1 | 300 |
| Wire Cutter and Stripper | https://images.app.goo.gl/ZixNc27nAmEd2aUb6 | 1 | 600 |
| Jumper wires (Male-Female and Female-Male) | https://www.embededstudio.com/product/jumper-wires-male-female/ | 8 | 140 |

Total Price = 5970

**Arduino UNO:**

Arduino has 14 digital output pins. Pin 0 (RX) and Pin 1 (TX) are used for sending and receiving TTL serial data (RX and TX). Pins 2 and 3 can be used to trigger an interrupt on a low value (0), a rising or falling edge. Pins 3, 5, 6, 9, 10, and 11 are PWM pins. The pins 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK) are for SPI communication using the SPI library. Digital pin 13 is connected to a built-in LED. Arduino UNO has six analog pins, denoted by the letters A0 through A5, has 10 bits of resolution, or 1024 values. Pins A4 (SDA pin) and A5 (SCL pin) are used for TWI communication using the Wire library. Pin AREF and RESET are used for reference voltage for analog pins and to reset microcontroller board manually respectively. (ElectroSchematics.com, 2014)

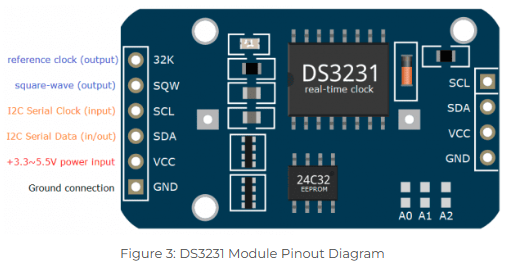
**USB Cable Adapter (for Arduino):**



(Amazon.com: 9V 1A AC DC 100V-240V Power Supply Adapter Cord for Arduino ..., n.d.)

Used to provide power supply to Arduino,

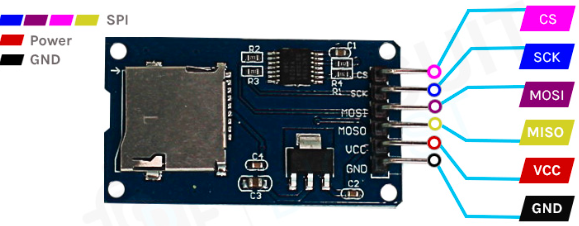
**DS3231 RTC Module**:



(How to Use a Real-time Clock Module with the Arduino, 2021)

In this, pins 32K, SQW, SCL, SDA, VCC, GND are used for 32 KHz oscillator, square wave signal, I2C serial clock. I2C serial signal, power supply (5/3/3 V) and ground respectively. (How to Use a Real-time Clock Module with the Arduino, 2021)

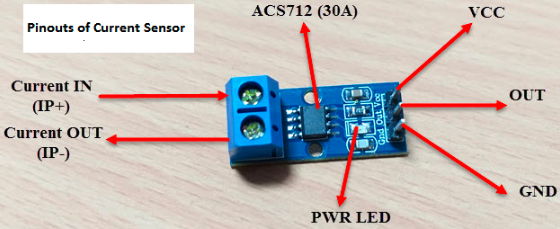
**Micro SD Card Module:**



(How Does a Micro SD Card Module Work and How to Interface it with Arduino?, n.d.)

In this, pins GND, VCC, MISO (Master in Slave Out), MOSI (Master out Slave In), SCK (Serial Clock), CS (Chip Select) are used for GND, power supply (5/3.3 V), SPI data output, SPI data input, data synchronization pulse generated by the Arduino, to enable or disable the module respectively.

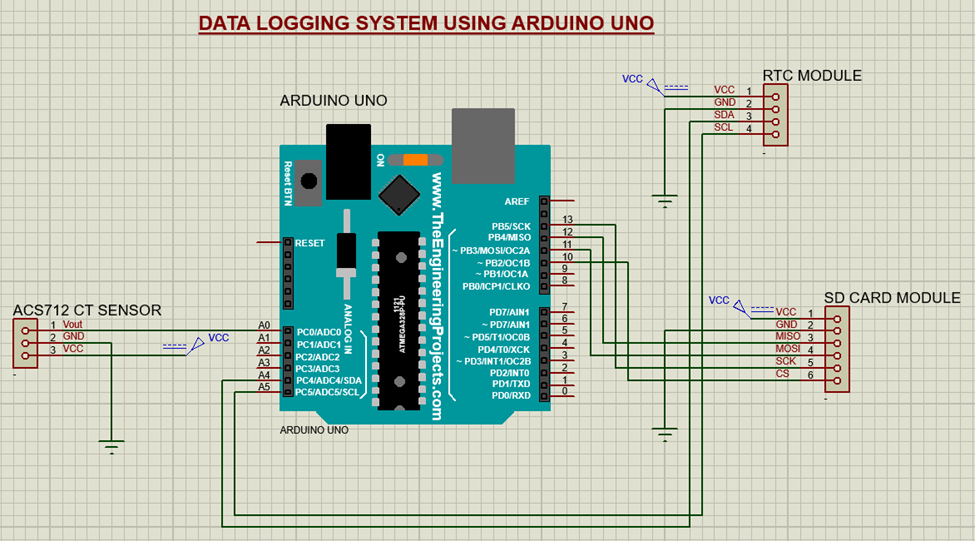
**CT Sensor ACS712 30A:**



(Robu.in | Indian Online Store | RC Hobby | Robotics, 2021)

In this, pins IP+ and IP- are used to connect the wire in series of which we want to measure the current. And pins, VCC, OUT, GND with the power supply, output (analog pin of Arduino) and with GND.

**Schematic:**



Proteus software was used to create the schematic for the data logging system using an Arduino UNO. The schematic includes three modules: an RTC, an SD card, and an ACS712 CT sensor. Each module is represented by a connector with the number of pins that the module actually possesses. Hardware connections are made according to this schematic.

**Construction:**

Connections of DS3231 RTC Module with Arduino:

Connect pins VCC, GND, SDA and SCL of RTC module with 5V, GND, A4 (SDA) and A5 (SCL) of Arduino UNO respectively.

Connections of Micro SD card Module with Arduino:

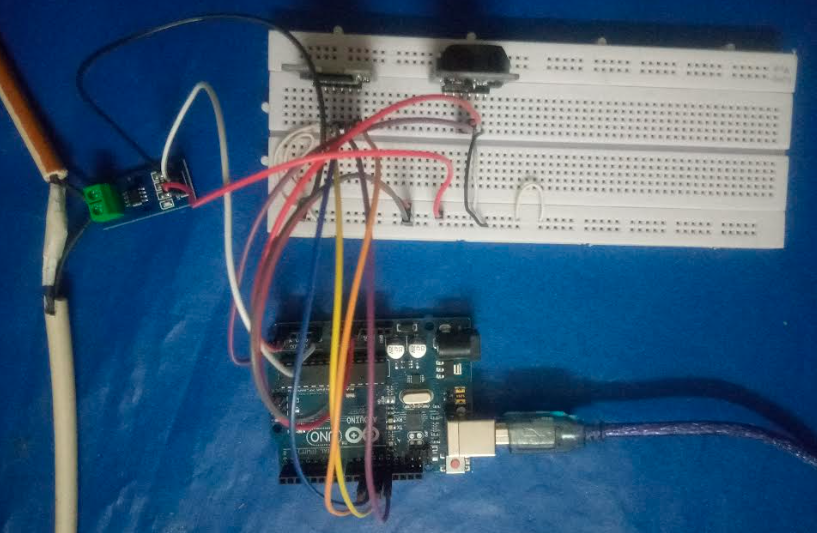
Connect pins VCC, GND, MISO, MOSI, SCK and CS of micro SD card module with 5V, GND, 12, 11, 13 and 10 of Arduino respectively.

Connections of Micro CT sensor with Arduino:

Connect pins Vout, GND and VCC with A0, GND and 5V of Arduino respectively.

Insert the SD card in SD card module and RTC cell in RTC card module.

Connect CT sensor in series with the wire of particular appliance using IP+ & IP- pins.

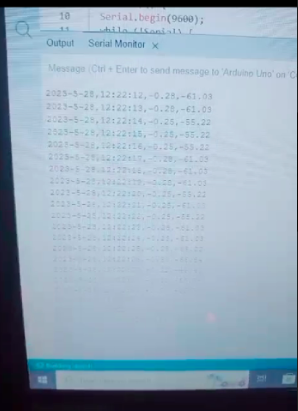
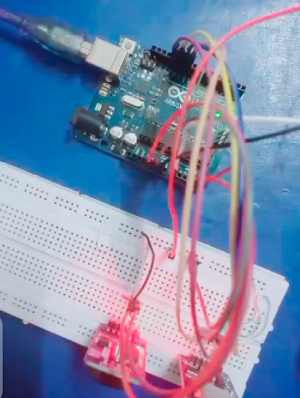
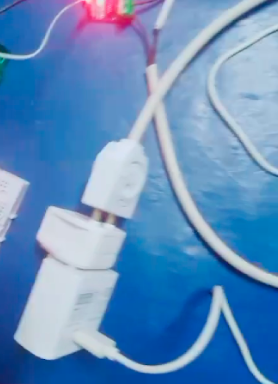


**Working:**

Arduino UNO (microcontroller board) is the brain of this data logging system. The current date and time are tracked by the DS3231 RTC module. I2C communication (SDA and SCL pins) is used to connect it to the Arduino Uno. The data (current & power consumption) is stored in the MicroSD Card module. It uses SPI communication (MISO, MOSI, SCK, and CS pins) to connect to the Arduino Uno. The current is measured by ACS712 30A current sensor. The live wire of the refrigerator's power source is connected to the CT Sensor. The analogue output it produces, which is connected to analogue pin A1 of the Arduino Uno, determines the current. Arduino UNO supplies power to RTC module, MicroSD Card module, and CT sensor. The code reads the value (current) through CT sensor, then convert it into power by multiplying current with 220V (fixed/ reference voltage) and logs the data (current and power) to csv file in SD card through SD card module. The code also opens the data logging file, initializes the RTC module and SD card, and creates the data with a timestamp (date and time) before logging it to the file.

The power consumption information is stored every hour (1200 seconds) at 20-minute intervals. The data file contain the values for current, voltage, date, time, and power. The code will log data only for 24 hours. After 24 hours, the data logging file will closed and circuit will not log any more data.

The Arduino was directly connected by DC barrel jack with home supply using adapter.



**Code:**

#include <Wire.h>

#include <RTClib.h>

#include <SD.h>

RTC\_DS3231 rtc;

File dataFile;

const int ACS712\_PIN = A0;

const int mVperAmp = 66;

void setup() {

  Serial.begin(9600);

  while (!Serial) {

    ;

  }

  if (!SD.begin(10)) {

    Serial.println("SD CARD IS NOT INITIALIZED!");

    return;

  }

  dataFile = SD.open("data.csv", FILE\_WRITE);

  if (!dataFile) {

    Serial.println("ERROR! .CSV FILE IS NOT OPENED");

    return;

  }

  if (!rtc.begin()) {

    Serial.println("COULD'NT FIND RTC");

    while (1);

  }

  if (rtc.lostPower()) {

    Serial.println("RTC LOST POWER, SETTING THE TIME...");

    rtc.adjust(DateTime(F(\_\_DATE\_\_), F(\_\_TIME\_\_)));

  }

  dataFile.println("Date,Time,Current,Power");

  dataFile.flush();

}

void loop() {

  DateTime now = rtc.now();

  int readValue;

  int maxValue = 0;

  int minValue = 1024;

  uint32\_t start\_time = millis();

  while ((millis() - start\_time) < 1000)

  {

    readValue = analogRead(ACS712\_PIN);

    if (readValue > maxValue)

    {

      maxValue = readValue;

    }

    if (readValue < minValue)

    {

      minValue = readValue;

    }

  }

  float Voltage = ((maxValue - minValue) \* 5.0) / 1024.0;

  float VRMS = (Voltage / 2.0) \* 0.707;

  float AmpsRMS = (VRMS \* 1000) / mVperAmp;

  float current = AmpsRMS-0.15;

  float power = current \* 220.0;

  String dataString = String(now.year()) + "-" + String(now.month()) + "-" + String(now.day()) + "," +

                      String(now.hour()) + ":" + String(now.minute()) + ":" + String(now.second()) + "," +

                      String(current) + "," + String(power);

  dataFile.println(dataString);

  dataFile.flush();

  Serial.println(dataString);

  delay(60000);

  DateTime nextDay = now + TimeSpan(24 \* 60 \* 60);

  if (nextDay <= rtc.now()) {

    dataFile.close();

    while (1);

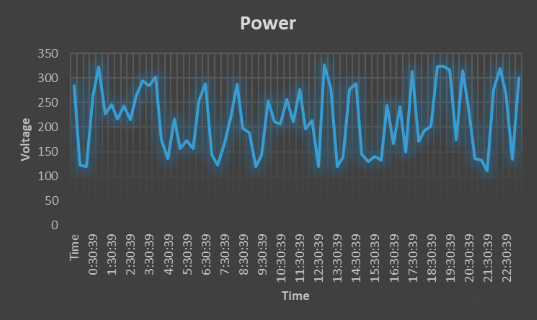
  }

}

Firstly, the libraries are called. Then, in the setup() function, the code initializes the serial communication, checks if the SD card is successfully initialized, opens the data file on the SD card, initializes communication with the RTC module, and sets the time if the RTC has lost power. It also writes a header line to the data file. After that in void loop() function starts by obtaining the current date and time from the RTC module. Then, it reads the current sensor values for 1 minute, tracking the maximum and minimum values. The voltage across the sensor is calculated, and from that, the RMS voltage and RMS current are derived. The current and power values are then calculated based on the RMS current and assuming a constant voltage of 220V. Next, the data is formatted into a string that includes the date, time, current value, and power value. This data string is written to the data file and flushed to ensure it is saved immediately. Additionally, the data is printed to the serial monitor for monitoring purposes. A delay of 1 second is added to create a 1-second sampling interval. The sketch checks if 24 hours have passed by comparing the current time with the next day. If 24 hours have passed, the data file is closed, and the program stops running. The loop() function continues to repeat, collecting current and power data every 1 second and saving it to the SD card along with the timestamp. After 24 hours, the data file is closed, and the program stops.

If you want to change the time of readings, change the delay.

**Result:**



The csv file that was created has shown that the power and current has fluctuations, varies throughout 24 hrs. The peaks in graph shows that the greater current is drawn, more power is consumed and compressor is on at that time. Also,

Average current = sum of current readings / 73

= 84.305/73

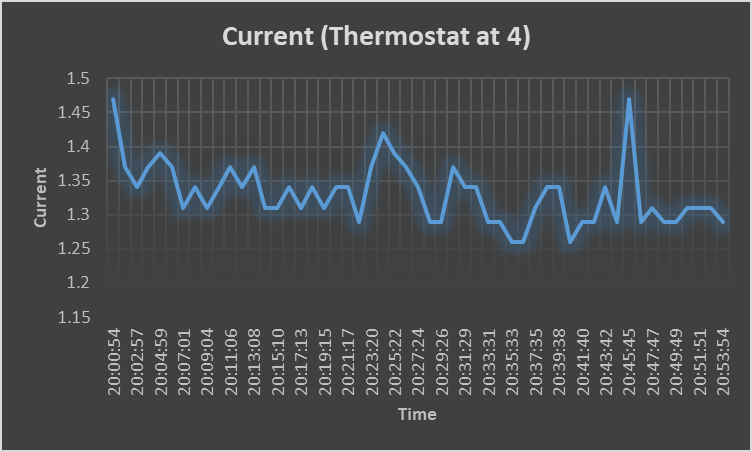
= 1.154A

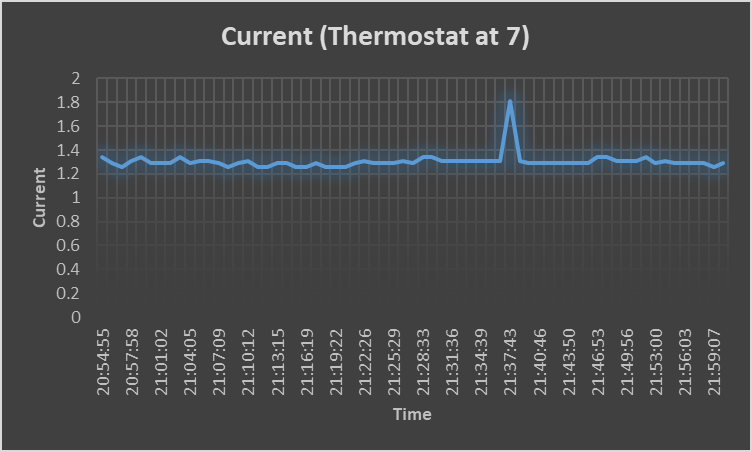
Average power = sum of power readings / 73

= 19073.48 /73

= 261.28W

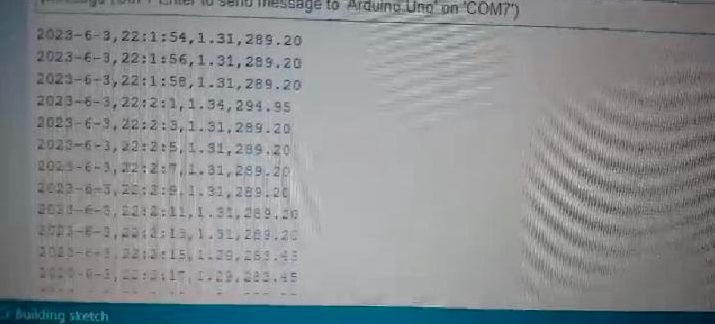
When we increase the value of thermostat, more current will draw, more power will be consumed and compressor will on for more time, and vice versa. This can be shown in graph.





The excel file is attached.

**Accuracy proof:**





https://drive.google.com/file/d/1O7pZpDhgjXKF1ttev67ct\_1PDn\_oxdiu/view?usp=drive\_link

In this video you can see that the time and date is correct and the readings measured by the sensor and the clamp meter are equal. Drive link of video is attached.

**Power solutions:**

Power solutions for field deployable systems are:

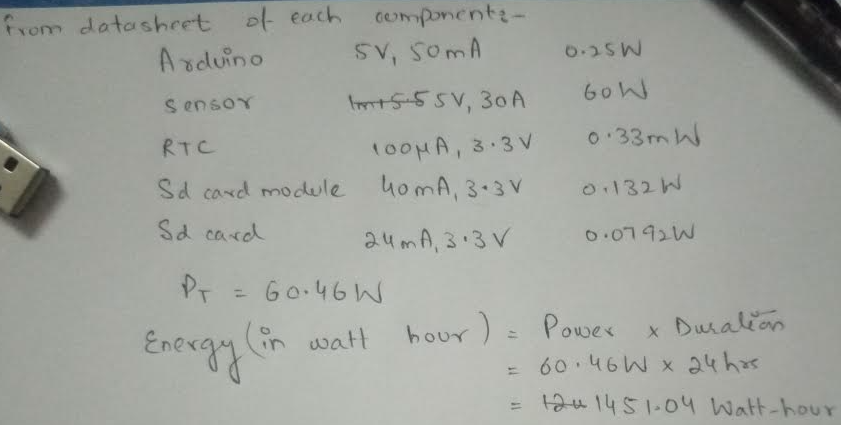
* Connect with household electricity

But in case of load shedding, these can be used:

* Rechargeable Batteries
* Power bank
* Solar panel

In our data logging system, we don’t require any source during load shedding because we’ve to measure the current and power of fridge, and during load shedding if our appliance is off, then there’s no need to connect any extra power source. So, we connected our system with the household electricity.

Otherwise in other data logging field deployable system for 24hrs or more, we can use battery/power supply by calculation the total power consumption of your system (circuit). Like this:



Like this you can estimate the energy consumption per hour. Now, you can calculate battery capacity, value.

Power bank and solar panel can also be used.

**Conclusion:**

In this assignment, the objectives of this assignment are successfully fulfilled. We logged the data in SD card successfully. We learned a lot in this assignment. The data (current drawn & power consumed by household fridge) that we’ve obtained shows that the power consumed and current drawn by household fridge. After analyzing the data of .csv file, we can conclude the average power and average current consumed by household fridge that’s 261.28W and 1.154A respectively.

**Bibliography:**

*Amazon.com: 9V 1A AC DC 100V-240V Power Supply Adapter Cord for Arduino ...* (n.d.). Retrieved from Amazon: https://www.amazon.com/Adapter-Arduino-Elliptical-Recumbent-Positive/dp/B07ZM46WKF

*ElectroSchematics.com*. (2014, January 02). Retrieved from Arduino Uno Pinout: https://www.electroschematics.com/arduino-uno-pinout/

*How Does a Micro SD Card Module Work and How to Interface it with Arduino?* (n.d.). Retrieved from Arduino Micro SD Card Module Tutorial - How SD Card Module Works and How to use it with Arduino: https://circuitdigest.com/microcontroller-projects/interfacing-micro-sd-card-module-with-arduino

*How to Use a Real-time Clock Module with the Arduino*. (2021, November 19). Retrieved from Circuit Basics: https://www.circuitbasics.com/how-to-use-a-real-time-clock-module-with-the-arduino/

*Robu.in | Indian Online Store | RC Hobby | Robotics*. (2021, February 03). Retrieved from Interfacing ACS712 Current Sensor with Arduino - Step by Step Guide to Measure Current - Robu.in: Indian Online Store: RC Hobby: Robotics: https://robu.in/interfacing-acs712-current-sensor-with-arduino-step-by-step-guide-to-measure-current/