Computerized Data Acquisition System for Minimizing Electricity Theft

Taha Shlibek[1,*] Abdalla I. Fadel [2,*] Manal Shlibek[3]

[1]Master student at Nanjing University of Aeronautics and Astronautics - shlibektaha@gmail.com

[2]Professor - School of Engineering and Applied Science/ Libyan Academy- fadel.abdalla@zu.edu.ly

[3]Engineer at GECOL- MSc- Electrical Engineering/ Libyan Academy- manal.shlibek@gmail.com

Abstract

Electrical power theft is the main cause of Non-Technical Losses (NTL) in the power grids. To solve this problem, the authors suggest an Arduino based system with the assistance of other devices to automatically detect and control power theft. In this paper power consumption data is classified into the no-load, normal load, and theft load data. The proposed system is termed as "Computerized Data Acquisition System for Minimizing Electricity Theft" as it receives the consumption data from the consumer side and then classifies this data based on how it is to be handled for data acquisition, monitoring, and detection. In no-load condition, the source side does not receive any electrical signal from the consumer side, indicating no consumption for that particular household. In normal load conditions, the consumer side sends current consumption information to the source end, and the source shall then compare it with the measured line current. if data is similar, no power theft signal is generated. Otherwise, a probable theft (an illegal connection of electricity) took place. This paper introduces a data acquisition strategy that is based on Arduino to convert an analog sensor data to a digital signal to be fed to the Arduino for further handling. The obtained results after testing the system proved to be accurate and reliable. Energy consumption was timely monitored and displayed on an LCD screen.

www.partner.org

Keywords: ARDUINO UNO, IDE, ACS712, DAQ, ADC, Master and Slave boards.

1. Introduction

The Non-Technical Losses (NTL) in the electrical power system (NTL) comprises illegal connections, meter tampering, billing errors, etc. out of which electricity theft through meter tampering and direct rigging from the transmission or the distribution line contributes a higher percentage of loss of electricity [1], [2]. Power theft is an international problem, in both developed and developing countries. In the United States of America, the NTL is estimated to be about 0.5% to 3.5% of the gross annual revenue [3]. Whereas, in the UK, the theft of electricity in England and Wales is 13% higher in 2019 than in 2018.

In developing countries like India, the loss of electricity due to theft is projected to about 20% to 30% of the overall loss in power utility [4]. There are currently about 52 million electricity energy meters that are interconnected in India and it is estimated the figure will reach 2 billion by 2030 [5]. In Sri Lanka, power stations face a loss of nearly 30% of its total distribution of electricity [6]. The evidence from a sample of 102 countries between 1980 and 2010 shows that theft is increasing in many parts of the world [7].

Zimbabwe's power utility does not have a real-time monitoring system that is able to detect the exact location where electricity theft is taking place. For this reason, the power utility has found it challenging to address electricity theft in the country [8]. Same is happening in the Libyan's power grid, where the electricity theft is increasing day by day.

A lot of researchers have previously studied and investigated power theft, how it is detected and stopped using smart technology. The electricity theft varies from country to country. The losses in the USA, Russia, Brazil, and India were 6%, 10%, 16%, and 18%, respectively, of their total energy production [9]. A study was conducted in BC Hydro in Canada suggested that they had reduced their non-technical loss figure by 50% within a three-year period [10]. Another study undertaken by Imperial College London, on behalf of UK Power Networks, estimated that a 2.8% reduction in customer's energy consumption will lower associated network losses by 5.5% [11]. A work conducted at Aalto University in Finland concluded that using statistical methods to reconcile the various data sets could become problematic when there are "major losses" on the network between the various meters. Their conclusions point towards a need to calculate energy losses between meters to balance measurements [12].

In 2018, a study [13] released at Oxford University on implementing prepayment meter system to minimize the power theft. The study proved to be suitable for developed and developing countries. The study discusses improving electricity networks' infrastructure to be compatible with prepayment meter system without any inadequacies. The researchers used the empirical case of Mozambique's national electricity company and its transition to an electricity network mediated by prepayment technologies in the capital city Maputo. The study, however, suggests that a lot of maintenance and reparation need to be done to enhance the concept of the prevention of power theft. The study explores how utilities engage with informality to produce access to 'formal' electricity networks through everyday processes of maintenance and repair.

Researchers in India 2019 have designed and implemented a system in order to remotely locate energy meter monitoring with control and theft detection through GSM [14]. Researchers proposed a wireless system inside the smart meter. With their proposed system, they could automatically generate electricity bills. The power theft detection mechanism of their system and ours is almost the same. With the difference that while they use a smart energy meter; this paper discusses using a mechanical energy meter. Their proposed system is not compatible with the outdated power grid in Libya. Hence, power theft indicating system proposed in this paper is more suitable for the Libyan infrastructure. Once power theft gets detected, the system’s relay will stop it on the spot.

2. Data Acquisition Hardware based the Proposed System

The proposed system has three circuits, namely Master, Slave, and theft circuits. Both Master and Slave circuits are almost equal, both having Arduino, LCD, current sensor. With the only difference that the Master circuit has a relay, whereas, the Slave circuit has a load (such as lamp/s).

The Master circuit acquires the data from the Slave circuit (data acquisition) for monitoring. The parameters of the system can be viewed either in analog form or digital form. The proposed system divides consumption modes into three categories: no load consumption, normal load consumption, and power theft mode. If the data consumption is within normal range the electricity shall not be cut off.

Data acquisition (DAQ) hardware acts as the interface between a processor and signals. It digitizes analog signals so that a processor can interpret and manipulate them. The three main components of the DAQ components are included in the proposed system for measuring signals. These components are the ACS712 current sensor, the signal conditioning circuit (built-in an ACS712 sensor), the analog-to-digital converter (ADC), that is built-in an Arduino, the Arduino which is the processor, and cables.

1. Calculating ac Current by the ACS712- 30A Sensor

In this paper, we measure the voltage output from the ACS712 current sensor. This voltage output represents the value of the load that we are trying to identify its legality. As the ACS712 current sensor converts physical parameters (electricity values) to electrical signals to be acquired by signal conditioning circuitry to convert, in turn, signals into a form that can be converted to digital so that it can be sent to the Analog-to-digital converter that is built in the Arduino. The converter converts conditioned sensor signals to digital values. Thus, the proposed system can be considered a data acquisition system.

The sensor used is ACS712-30A, measuring 30 and -30 ampere current and output sensitivity is 66mv/A. Current could be measured by using below sensitivity equations:

- $\mathrm{adc}_{\mathrm{value}} = \mathrm{analogRead(current\ pin)};$ For reading the value from the analog pin.
- $\mathrm{adc}_{\mathrm{Voltage}} = (\frac{\mathrm{adc}_{\mathrm{value}}}{1024} * 5000);$ For getting milli volts.
- $\mathrm{Current\ Value} = (\frac{\mathrm{Voltage - offsetvoltage}}{\mathrm{Sensitivity}});$ For getting the desired current value.

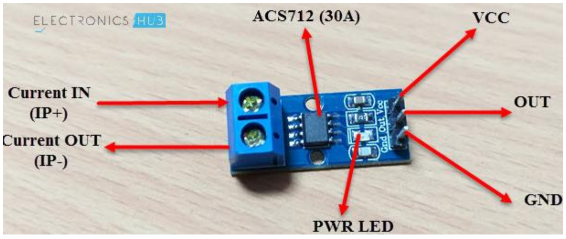


Figure 1. Displays the ACS712-30A module.

2. Analog-to-digital converter (ADC)

The Arduino-ADC takes the input voltage signal and converts it into a digital value. The resolution of Arduino Uno ADC is 10 bit. Meaning that it has the ability to detect 1,024 (2^{10}) discrete analogue signals and integer values from (0-1023). Consequently; input voltages between 0 and 5 volts equal integer values between 0 and 1023=2^10 or between 1 and 1024=(2^{10}). (5/1024)= 0.00488V= 4.88mV or 4.9mV per unit.

The function analogRead(pin) is for Arduino ADC is used to read analogue values from the specified analog pin and the resolution is 10 bits, meaning that the ADC has the ability to detect 1,024 (2^10) discrete analogue levels.

If nothing is connected to analogue input channel then the analogRead () function returns the noisy fluctuating value.

2.1 Calculating current from output voltage of the ACS712 by ADC

In the proposed system, calculations are done in the following order to obtain the current from the sensor according to following points:

- The output sensitivity (mVperAmp) is 66mv/A means the ADC chip is 10-bit (0-1023). Accordingly, the ADC‘s output values from 0 to 1023 will convert into values from 0 to 512 volt in the form of analog or convert into values from 0 to 5 volt in case of digital form.
- If a power supply powers an ACS712 sensor chip with 5V, then, Vcc= 5V. This sensor has an analog output voltage (offset voltage) = 512v, this analog output voltage of the ACS712 will

- be digitized through the ADC chip to be equal to Vcc x 0.5 = 5/0.5= 2.5v. In this case, there is no current passing through a sensor. The offset voltage of the sensor should be subtracted from the measured voltage.
- Theoretically, when there is no current flowing through the sensor, the offset voltage of the ACS712-30A sensor would be 512V in the analog form (ac), which corresponds to 2.5 V in the digital form (dc). But practically, it was noticed 510.5 equals to 2.5v. Accordingly, we wrote the voltage_offset=510.5 instead of 512 to minimize measurement errors.
- The output voltage of this sensor is analog, thus, to read it, one can directly measure the output voltage using an Arduino through Analog Read pin (ADC pin).

2.2 Measuring the AC-Current

By using analogRead function stored in Arduino library we could measure the output voltage of the current sensor.

For measuring a group of samples, the measured digital values will be stored into the code with variable we named 'Adcvalue' after creating a loop into the code for a number of samples, shown as follows:

$$\{\quad \text{for (int j} = 0;\ \text{j} < 5;\ \text{j} + +)$$
$$\{\text{Adcvalue} += \text{analogRead(A0)};\}$$
$$\text{float adc}_{\text{average}} = \frac{\text{float(Adcvalue)}}{5} \quad \}$$

In this for loop we calculate five samples, once the Arduino gets 5 values (samples) from the current sensor, the average value is calculated in order to eliminate ACS712 measure fault as follows: ADC_average= Adcvalue/5.

It could be any number of samples, for example, for calculating an average of 10 samples instead of 5 samples, the code line should be changed to:

$$\{\quad \text{for (int j} = 0;\ \text{j} < 10;\ \text{j} + +)\ \{\text{Adcvalue} += \text{analogRead(A0)};\}$$
$$\text{float adc}_{\text{average}} = \frac{\text{float(Adcvalue)}}{10} \quad \}$$

Based on the sensitivity of ACS712-30A is 0.066, then, the coefficient of ACS712-30A is:

Coefficient = [5/(1024*0.066)] = 5/1024/0.066 (1)

Where, Iac = ADC * coefficient (2)

Squaring all parts of equation (2), we get (3):

$$(\mathrm{Iac})^2 = \{(\mathrm{ADC})^2 * (\mathrm{coefficient})^2\} \quad (3)$$

Inserting (1) into (3) we get (4), as follows:

$$(\mathrm{Iac})^2 = (\mathrm{ADC})^2 * ((5.0/1024.0)/0.066)^2 \quad (4)$$

In Arduino C, there is an Arduino function that calculates the value of a number raised to a power. This function is: pow(,).

Therefore, $\mathrm{X}^2 = \mathrm{pow(base, exponent)} = \mathrm{pow(X, 2)}$

By applying the concept of $\mathrm{X}^2 = \mathrm{pow(X, 2)}$, into (4) we get (5), as follows:

$$(\mathrm{I_{ac}})^2 = \mathrm{pow(ADC, 2)} * \mathrm{pow}\left(\frac{\frac{5.0}{1024.0}}{0.066}, 2\right) \quad (5)$$

Where,

$$\mathrm{pow(ADC, 2)} = \mathrm{pow(adc_{result}, 2)} \quad (5.1)$$

$$\text{and} \quad \mathrm{pow(adc_{result}, 2)} = \mathrm{pow(adc_{average} - voltage_{offset}, 2)} \quad (5.2)$$

$$\text{and} \quad \mathrm{adc_{result}} {+}{=} \mathrm{pow(adc_{average} - voltage_{offset}, 2)} \quad (5.3)$$

Rearranging equation (5) we get (6):

$$(\mathrm{I_{ac}})^2 = \mathrm{adc\_result} * \mathrm{pow((5.0/1024.0)/0.066, 2)} \quad (6)$$

$$\text{Taking} \quad \mathrm{I_{rms}} = \sqrt{(\mathrm{I_{ac}})^2} = \sqrt{((\mathrm{adc_{result}}) * \mathrm{pow}\left(\frac{\frac{5.0}{1024.0}}{0.066}, 2\right))} \quad (7)$$

$$\text{Considering } \mathrm{I_{ac,rms}} = \frac{\mathrm{I_{rms}}}{\text{number of reading}} \quad (8)$$

$$\text{Hence we get,} \quad \mathrm{I_{ac,rms}} = \frac{\sqrt{((\mathrm{adc_{result}}) * \mathrm{pow}\left(\frac{\frac{5.0}{1024.0}}{0.066}, 2\right))}}{\text{number of reading}} \quad (9)$$

Equation (9) needs to be written into the source code to find out the RMS current value that is calculated by the current sensor ACS712-30A.

To calculate the RMS current, one needs to take a large number of readings at regular intervals. In our code, we consider 300 readings, but it could be any number between 200 and 500. Then all readings are added to a running total and then by taking the square root we get the total RMS value. When the total RMS value is captured, it must be divided by the number of readings, as a result, the RMS current value is obtained. The ACS712 sensor is connected with an Arduino for both Master and Slave boards shown in figure 2.

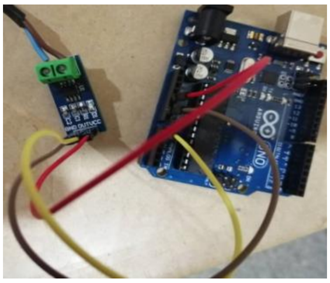


Figure 2. Connecting the ACS712 with the UNO in the System.

Code 1 shows steps taken to interface an Arduion with a ACS712-30A .

```
unsigned long adc_result=0;
int number_of_readings_from_adc=300;
float voltage_offset=510.5;
for(int i=0; i<number_of_readings_from_adc; i++){
  unsigned long Adcvalue=0;
  for(int j=0; j<5; j++){ Adcvalue+=analogRead(A0); }
  float adc_average = float(Adcvalue)/5;
  adc_result += pow(adc_average-voltage_offset,2);}
source_current = sqrt(float(adc_result)*pow(5.0/1024.0/0.066,2)/number_of_readings_from_adc);
```

Code 1. Interfacing an Arduino with ACS712-30A.

Running this code, the Arduino board can calculate the values from the current sensor, regardless of whether it is a Master board or Slave.

If the board is Master, then the calculated current from the sensor shall be:

```
source_current = sqrt(float(adc_result)*pow(5.0/1024.0/0.066,2)/number_of_readings_from_adc);
```

Code 2. Calculating the Source Current.

If the board is Slave, then the calculated current from the sensor shall be:

```
consumer_current=sqrt(float(adc_result)*pow(5.0/1024.0/0.066,2)/number_of_readings_from_adc);
```

Code 3. Calculating the Consumer Current.

3. Connecting the Master Board with the Slave Board

As the proposed system is set to get consumption data and prevent power theft. A set of Slave boards is connected to communicate with each Master board that is connected to the substation's energy meter. Each Slave board contains a primary Arduino UNO processor. Arduino UNO has a memory for temporarily storing data that is processed by the microcontroller during program execution and the Slave board's microcontroller transmits this processed data to the Master board's microcontroller which in turn has a second processor (second Arduino Uno). The data memory that is built in the second Arduino can also store a big number of consumption readings that come from Slave boards. Then the Master's microcontroller transmits the processed data to the utility's control room.

Connection between the Slave and Master boards is done using wires, and hence we need to write functions into the code to program I2C bus that connects two Arduinos together.

In the presented code (see Appendix), we have applied the six commands below for sending the consumption data from the Slave to the Master after the Master sends its request to the Slave board, as follows:

1. Wire.begin(2): meaning that we initiated the Wire library and joined the I2C bus as a master or slave with address 2, it is not necessary number 2, you can use any other number, however; Slave address should be changed to that number too. Same is applied to the Master code.
2. The Master board shall request data (5 bytes) from the Slave board using: Wire.requestFrom(2, 5); meaning that the Master Arduino requests five bytes from the Slave Arduino whose address is 2. Number 5 means five digits are to be measured. For instance, 32.17, a Slave must transmit ASCII symbols that equivalent "3", "2", ".", "1" and "7" characters to the Master.
3.

As the code consists of two loops, the first of which is the setup loop where the functions are executed once, and second is the void loop. Code 4 explains steps 1 and 2 that the Master and Slave boards do in the setup loop.

```
//for I2C
#include <Wire.h>

// for the Slave Board
void setup()
// join i2c bus with address 2
{Wire.begin(2);
//register event
Wire.onRequest(requestEvent); }
```

```
//for I2C
#include <Wire.h>

// for the Master Board
void setup()
//for I2C
{Wire.begin(); }
```

Code 4 shows Process occur in the setup loop.

4. These bytes are needed to be retrieved by the Master upon receiving them, Retrieving bytes can be done using Wire.available() function in the Master code.
5. The Master will then read these bytes using Wire.read()" command.
6. In the Slave code, we used the Wire.write(string) function and the string that we sent was a series of five bytes.

7- The Slave Arduino must transmit a byte to the Master Arduino as ASCII symbols or as an ASCII character which is a 7-bit character code. Where every single bit represents a unique character in electronic devices. As long as the string is a sequence of characters or symbols that are chosen from the alphabet set, then we get the string from writing this function: { consumer_current_str+=(char)Wire.read();}

Code 5 explains steps 3 to 6

```
// for the Slave Board
void loop()
{ if(abs(millis()-time_old)>200)
  { time_old=millis(); } }

void requestEvent() {
String str = (String)consumer_current;
for(byte i=0; i<5; i++){
Wire.write(str[i]); } }
```

```
// for the Master Board
void loop() {
  if(!theft){
Wire.requestFrom(2, 5);

if(Wire.available())
{ consumer_current_str="";
// slave may send less than requested
while (Wire.available())
// receive a byte as character
{consumer_current_str+=(char)Wire.read(); }
consumer_current=consumer_current_str.toFloat();}
```

Code 5 shows Process occur in the void loop.

However, in the Master code (Wire.available()) command was used, this command returns the numbers of bytes available to be read with the Wire.read() function. After making sure that every byte was transmitted from a slave device to a master board by using Wire.requestFrom() function, the Master will then read these bytes through the Wire.read() function.

consumer_current_str+=(char)Wire.read(); this function means the Slave Arduino shall transmit a byte to the Master Arduino as ASCII symbols or as a character. ASCII is a 7-bit character code where every single bit represents a unique character in electronic devices. Then convert each character to a string.

```
void setup() {
    Wire.begin(2);
    Wire.onRequest(requestEvent); }
void loop() {
 }
 void requestEvent() {
 //sending consumer current to source part arduino
  String str = (String)consumer_current;
  for(byte i=0; i<5; i++){
    Wire.write(str[i]);}
```

Code 6. Sending Five Bytes from the Slave Board.

In order to display values of the consumer current which was calculated via the Slave board, and values of the total consumption, that calculated via the Master board, we need to first convert a string form to a float number so that it can be displayed on LCDs. Consequently, the "consumer _current_str.toFloat()" function was used.

The float number is then stored into a variable named "consumer_current". The system's code was written as shown in Code 7.

```
void setup() {
    Wire.begin(); }
void loop() {
 Wire.requestFrom(2, 5);
if(Wire.available()){
consumer_current_str="";
while (Wire.available()) {
consumer_current_str+=(char)Wire.read();}
consumer_current = consumer_current_str.toFloat();}
```

Code 7. Converting from a String Form to the Float Number

The Slave Arduino board may send a number less than the specified bytes as requested. In other words, if the requested number is less than 10.00, for example, 9.65 this number has 4 bytes which are less than 5 bytes as it is requested. In this case, the Master Arduino should make sure it receives the exact number of bytes as it requests from the Slave Arduino. This can be done with the following command:

```
if(Wire.available()){
  consumer_current_str="";
```

Code 8. Checking the Number of Bytes

After ensuring that the Master Arduino has received theexact number of bytes, the Master controller exact number of bytes, the Master controller will retrieve bytes as characters and then will read them. We can see the command "Wire.available" is repeated in a while loop as illustrated in Code 9 so that it can read all the bytes.

```
if(Wire.available()){
  consumer_current_str="";
// receive each byte as character, retrieve them as bytes and
//reads a byte that was transmitted from the slave board
// to the Master after a call to requestFrom()
  while (Wire.available())
  { consumer_current_str+=(char)Wire.read(); }
  consumer_current = consumer_current_str.toFloat(); }
```

Code 9. Retrieving and Reading Bytes as Characters.

If the board is Slave, then the Slave controller will send bytes to the Master controller by calling requestEvent function, as illustrated in Code 10.

```
void requestEvent() {
  String str = (String)consumer_current;
  for(byte i=0; i<5; i++){
    Wire.write(str[i]); }}
```

Code 10. Calling requestEvent Function.

If it is desired to send five numbers, for example, 56.17 A. The Slave board must send the ASCII symbols the equivalent characters "5", "6", ".", "1" and "7". Accordingly, there will be five symbols or 5 bytes or 5 numbers, and hence for Slave code we have: (byte i=0; i<5; i++).

In void setup() function, we called the Wire.onRequest(requestEvent) function for sending five bytes. And at the end of the code we emphasized the role of requestEvent() function.

To connect two Arduino boards together with three wires, A4, A5, and GND pins of the Slave

Arduino board must be connected with A4, A5, and GND pins of the Master Arduino board respectively as shown in figure 3.

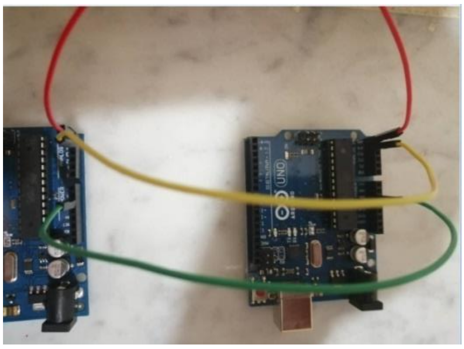


Figure 3. Connections between Master and Slave Boards.

As the data acquisition system typically converts analog signals into digital values for processing.

The data acquisition procedure in the proposed system is the process of getting a set of signals that measure real consumption of electricity and converting these signals into digital values that can be manipulated by a microcontroller that built-in an Arduino.

4. Construction of the Hardware System

The block diagram below (figure 4) shows the construction of the suggested system.

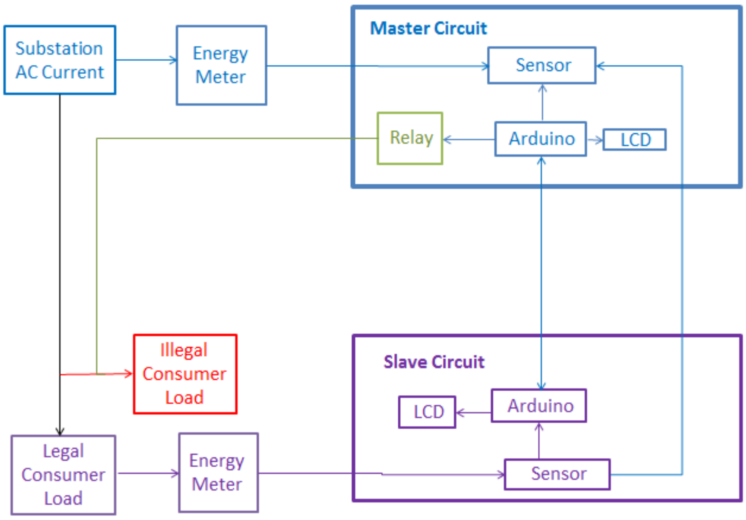


Figure 4. System block diagram.

All three circuitries are apparent in the figure below. The Slave circuit, which consists of four items which are an Arduino UNO, LCD 1602, ACS712-30A, and a lamp(s) to represent a load. Whereas, the Master circuit that consists of four components namely; an Arduino, LCD 1604, ACS712-30A, and a relay.

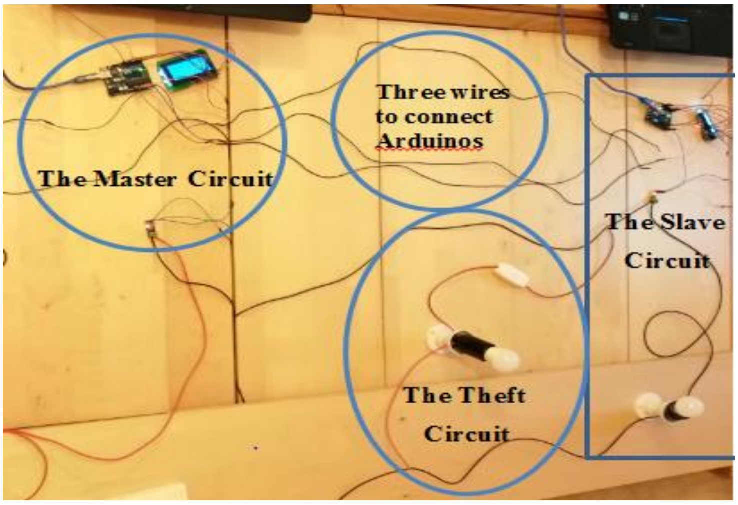


Figure 5. System Overview Diagram.

The proposed system has been modeled to detect and control power theft. As the Slave board is able to detect exactly where the probable theft has occurred. Any difference noticed between the Master and Slave boards indicates illegal usage.

5. System Flow Chart

The proposed system consists of two different wire circuits: Master and Slave circuits. Each circuit has its own functionality that differs from the other in tasks. The system, thus, requires two different codes programmed using C/ C++ languages which are compatible with Arduino based on wire connection that shown in Appendices.

6. Evaluation and Results

The monitoring system has been projected in real time. Using the design discussed above, we could successfully address consumers power theft problem in a completely automated, cost-effective and most importantly reliable way. The hardware project is focused on three arguments namely: no load, normal load, and power theft.

The hardware mainly consists of Arduino, current sensors, relay, and LCDs. Two current sensors were connected together, one sensor is connected to the substation's energy meter and the other is connected to a home's energy meter. These two sensors are used to compare the currents at the two sides. The current varies continuously in a transmission line. All the time two sensors compare currents at both ends which are then shown in the LCD screen. If any theft gets detected in the line between these sensors, current values mismatch occurs. The information from two sensors is passed to Arduinos through wires. The Master Arduino receives the signal and displays it on the LCD screen. The Master circuit is integrated to the meter's substation so as to remotely cut off the power supply from a house or any consumer who trying to steal the power. Figure 9 shows the results from the system for power theft case.

Figure 9. Results from the System for Power Theft Case.

The Monitoring system has been implemented and the desired results have been obtained. Figure 10 shows other examples of power theft conditions

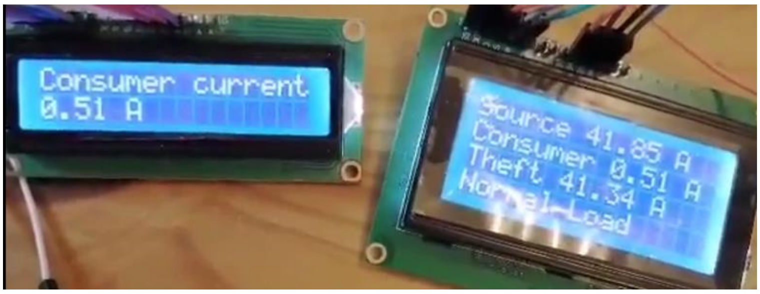


Figure 10. Shows the deference between Master and Slave Readings.

The Master and Slave boards are configured to start automatically as part of the booting process and run on time as long as the system is powered on. When a Slave board is switched on, a consumed

power data is stored into the temporary data memory. When a call is received from the Master board, each byte of data is sent as a character and accordingly a reaction will be taken by the Master board. This procedure is configured to work throughout the day to immediately inform the utility in case of illegal use. Data is then used by the Master code for further processing.

7. Conclusion

In the paper, the authors introduce a data acquisition system for the power system Automation. The system was projected for detecting illegal signals on the electricity network. The process starts with the aggregation of the data at a home side and ends with the actuation of the proposed model. Data gathered from various Slave boards that is stored in the controller's memory of the Master board. This data is stored in the Master Arduino and it is used for identifying consumption categories. The data set is divided into two data sets, one contains legal consumption data (where the data is within normal average load), and the other data set contains illegal consumption data (the power theft dataset) that will be used for preventing the theft. If the total losses are greater than expected value then the relay, at the point the theft has occurred, will electronically open the circuit. Accordingly, the electricity will be cut off from that point. As the process is continuous, the data is recorded for different instants of time and timely sent to the station. If normal conditions are met, the system works normally without any disturbance that involved cutting-off or reducing the quantity of electricity. Data was acquired from a load to the Slave current sensor first, then, from the Slave sensor to the Slave Arduino and lastly, from the Slave Arduino to the Master Arduino. Accordingly, the utilization of the DAQ concept based on Arduino for the control of electricity consumption over a network is achieved to stabilize the power grid.

8. References

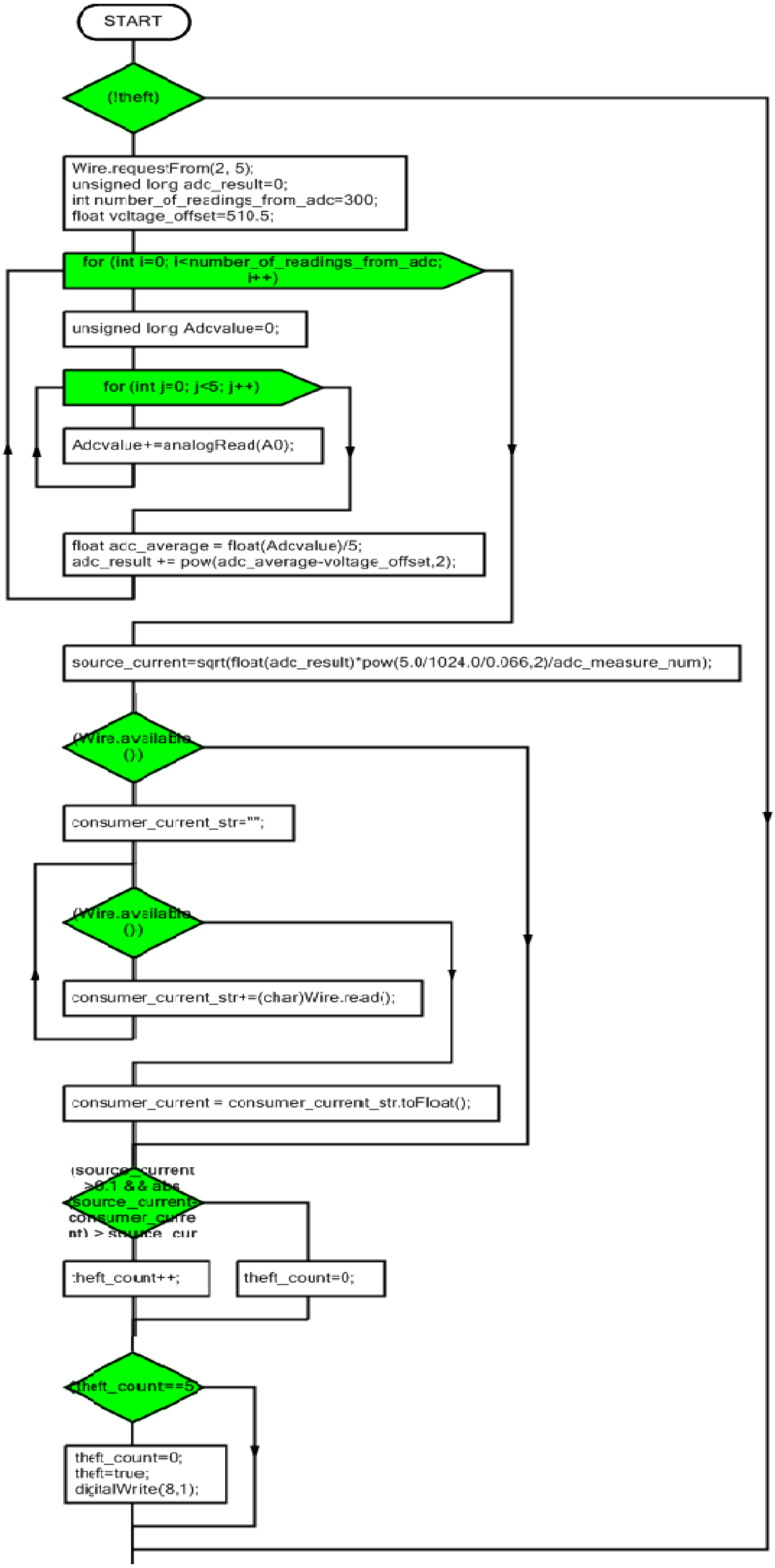
1. Nagi. J, Yap. K, Tiong. S, Ahmed. S, Malik Mohamad, “Non Technical Loss Detection for Metered consumers using support vector machines”, IEEE transactions on Power Delivery, Vol.25, No.2, Pg. 1162-1171, April 2010.
2. Pabla. A, “Electric Power Distribution”, Pg. 213-215, Fifth Edition, Eleventh Reprint, Tata McGraw Hill Publications, 2008.
3. Prakash. R, Jebaseeli. E, Sindhu. Y, "Power Theft Identification Using GSM Technology”, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (An ISO 3297: 2007 Certified Organization) Vol. 6, Issue 3, November 2017, IJAREEIE www.ijareeie.com, Department of Electrical and Electronics Engineering, Sathyabama University, India.
4. International Journal Of Engineering Sciences& Research Technology (IJESRT 2016), "Power Theft Prevention System Via Remote Monitoring". Bigyan Basnet, Dileep Kumar Appana*, Arjun Prasad Kandel, Narayan Poudel, Department of Electrical Engineering, Advanced College of Engineering, Nepal. * Lead Project Engineer, Enersys Energy Solutions Hyderabad, India. Department of Electrical Engineering, Advanced College of Engineering, Nepal.
5. Jamil. F,”Comparison of Electricity Supply and Tariff Rates in South Asian Countries” Available at: http://www.efsl.lk/reports/electricity_supply_south_asian_ countries.pdf[Accessed 18thJuly 2014], 2011.

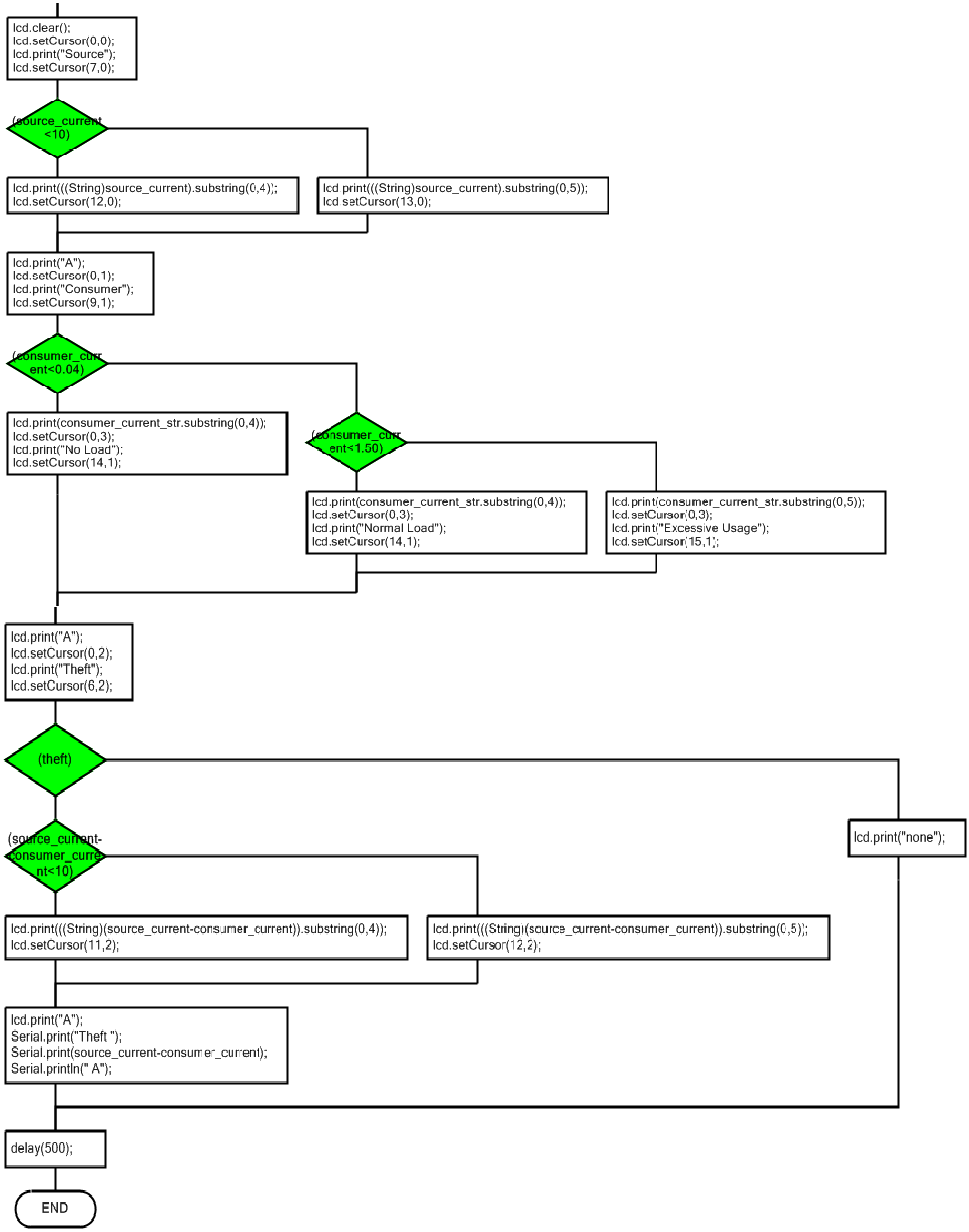
6.

7. Smith. T, ”Electricity Theft: a Comparative Analysis, Energy Policy”. 32(18), pp. 2067– 2076, 2008.
8. Electricity sector India information was taken from: https://www.ibef.org/industry/power-sector-india.aspx
9. Musungwini. S, "A Framework for Monitoring Electricity theft in Zimbabwe using Mobile Technologies", 2016 Midlands State University Gweru, Zimbabwe.
10. Bank.W, Electric Power Transmission and Distribution Losses (% of output); IEA: Paris, France, 2016
11. Smart Meters and Losses: Best Practice Review. Registered in England and Wales. Registered No. 3870728. Registered Office: Newington House, 237 Southwark Bridge Road, London.
12. UK Power Networks (Operations) Limited. Registered in England and Wales, London.
13. https://www.euramet.org/Media/docs/Repository/A169/ENG04_smart_grids/KorhonenAkseli.pdf
14. Acuto. M, Dinardi. C and Marx. C, Electricity Services for Informality and the Work of Infrastructure Maintenance and Repair in an African City Idalina Baptista.
15. Alsaba. S, Rajderkar. T, Raisoni. G, “Design and Implementation of Remotely Located Energy Meter Monitoring With Control and Theft Detection through GSM”, International Conference on Innovations in Engineering, Technology, Science & Management, (ICI-ETSM-2019), 09, Issue 5. Institute of Technology (JIT) is governed by Samridhi Sarwajanik Charitable Trust (SSCT), College of Engineering , Department of Electrical Engineering, Nagpur, India, (May. 2019), www.iosrjen.org.

8. Appendices

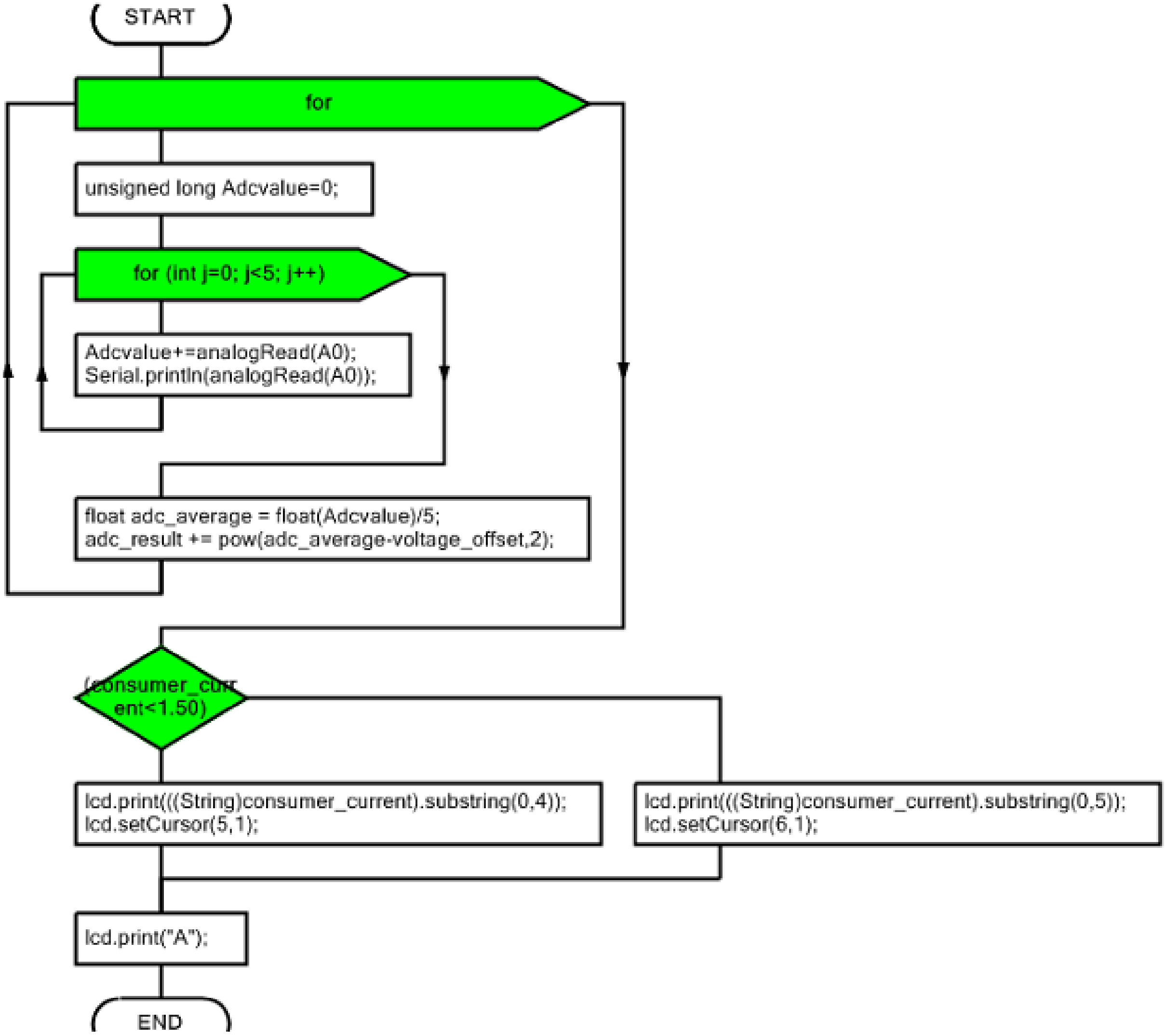
Appendix A: Wire Master Code:


START
(!theft)
Wire.requestFrom(2, 5);
unsigned long adc_result=0;
int number_of_readings_from_adc=300;
float voltage_offset=510.5;
for (int i=0; i<number_of_readings_from_adc; i++)
unsigned long Adcvalue=0;
for (int j=0; j<5; j++)
Adcvalue+=analogRead(A0);
float adc_average = float(Adcvalue)/5;
adc_result += pow(adc_average-voltage_offset,2);
source_current=sqrt(float(adc_result)*pow(5.0/1024.0/0.066,2)/adc_measure_num);
(Wire.available())
consumer_current_str="";
(Wire.available())
consumer_current_str+=(char)Wire.read();
consumer_current = consumer_current_str.toFloat();
theft_count++;
theft_count=0;
(theft_count==5)
theft_count=0;
theft=true;
digitalWrite(8,1);


lcd.clear();
lcd.setCursor(0,0);
lcd.print("Source");
lcd.setCursor(7,0);
(source_current <10)
lcd.print(((String)source_current).substring(0,4));
lcd.setCursor(12,0);
lcd.print(((String)source_current).substring(0,5));
lcd.setCursor(13,0);
lcd.print("A");
lcd.setCursor(0,1);
lcd.print("Consumer");
lcd.setCursor(9,1);
(consumer_current<0.04)
lcd.print(consumer_current_str.substring(0,4));
lcd.setCursor(0,3);
lcd.print("No Load");
lcd.setCursor(14,1);
(consumer_current<1.50)
lcd.print(consumer_current_str.substring(0,4));
lcd.setCursor(0,3);
lcd.print("Normal Load");
lcd.setCursor(14,1);
lcd.print(consumer_current_str.substring(0,5));
lcd.setCursor(0,3);
lcd.print("Excessive Usage");
lcd.setCursor(15,1);
lcd.print("A");
lcd.setCursor(0,2);
lcd.print("Theft");
lcd.setCursor(6,2);
(theft)
lcd.print("none");
(source_current-consumer_current<10)
lcd.print(((String)(source_current-consumer_current)).substring(0,4));
lcd.setCursor(11,2);
lcd.print(((String)(source_current-consumer_current)).substring(0,5));
lcd.setCursor(12,2);
lcd.print("A");
Serial.print("Theft ");
Serial.print(source_current-consumer_current);
Serial.println(" A");
delay(500);
END

B). Wire Slave Flow-chart

The Slave wire code will be uploaded to the Slave Arduino is shown in figure 7.



.

C). Calling the requestEvent Function:

For void requestEvent() section, the flow-chart is shown as follows: