Internship Project
MOTORIZED MACHINE

PRESENTED BY

Anish Mayanache
Shraddha Karekar
Rohan Mangalore

CONTENTS:

ABSTRACT:

The objective of this project was to design and develop a current sensor device using the ACS712 integrated circuit. The ACS712 is a highly accurate and reliable current sensor chip capable of measuring both AC and DC currents. This report presents the methodology, design considerations, and implementation details of the current sensor device. The project began with a thorough review of the ACS712 datasheet and related literature to understand its operating principles and specifications. Based on this knowledge, a hardware design was developed, consisting of the ACS712 chip, necessary supporting components, and an Arduino microcontroller for data acquisition and processing.

The hardware prototype was assembled, ensuring proper connections and component placements. Calibration procedures were performed to determine the output sensitivity and offset of the ACS712 sensor, ensuring accurate current measurements. The Arduino microcontroller was programmed to read the analog output of the ACS712 and convert it into meaningful current values. To validate the performance of the current sensor device, a series of tests were conducted using different current sources. The measurements obtained from the current sensor device were compared against the reference values to assess the accuracy and reliability of the sensor.

INTRODUCTION:

A current sensor device made using ACS712 is a highly accurate and reliable solution for measuring electrical currents. The ACS712 sensor utilizes the Hall effect to detect the magnetic field generated by the current passing through a conductor. It provides precise measurements of both AC and DC currents.The ACS712 sensor features different models with varying current ranges, allowing users to select the appropriate sensor based on their specific application needs. It offers analog output, typically in the form of a voltage signal, which is proportional to the current being measured. This allows for precise and consistent current measurements.

COMPONENTS REQUIRED:

Name	Quantity	Component
Arduino UNO	1	Arduino Uno R3
KEYPAD	1	Keypad 4x4
LCD display	1	LCD 16 x 2
Light Bulbs	3	Light bulbs
LED lights	1 each	Red, Green, Yellow Light
Resistors	3	1 kΩ Resistor
DPST	3	DPST switch
Battery	1 each	12V, 9V Battery
Current Sensor	1	ACS712
Jump Wires	As req.	Nil

COMPONENT DESCRIPTION:

1. Arduino UNO:

The Arduino Uno is a widely-used microcontroller board that serves as the cornerstone of the Arduino platform. It features an ATmega328P microcontroller, operating at 16 MHz, and offers a range of digital and analog input/output pins for connecting to various sensors, actuators, and other electronic components. The board can be powered via USB or an external power source and supports programming through the Arduino IDE, which simplifies code development and uploading. The Arduino Uno is known for its versatility, ease of use, and extensive community support, making it a popular choice for prototyping and DIY electronics projects.

2. ACS712 sensor:

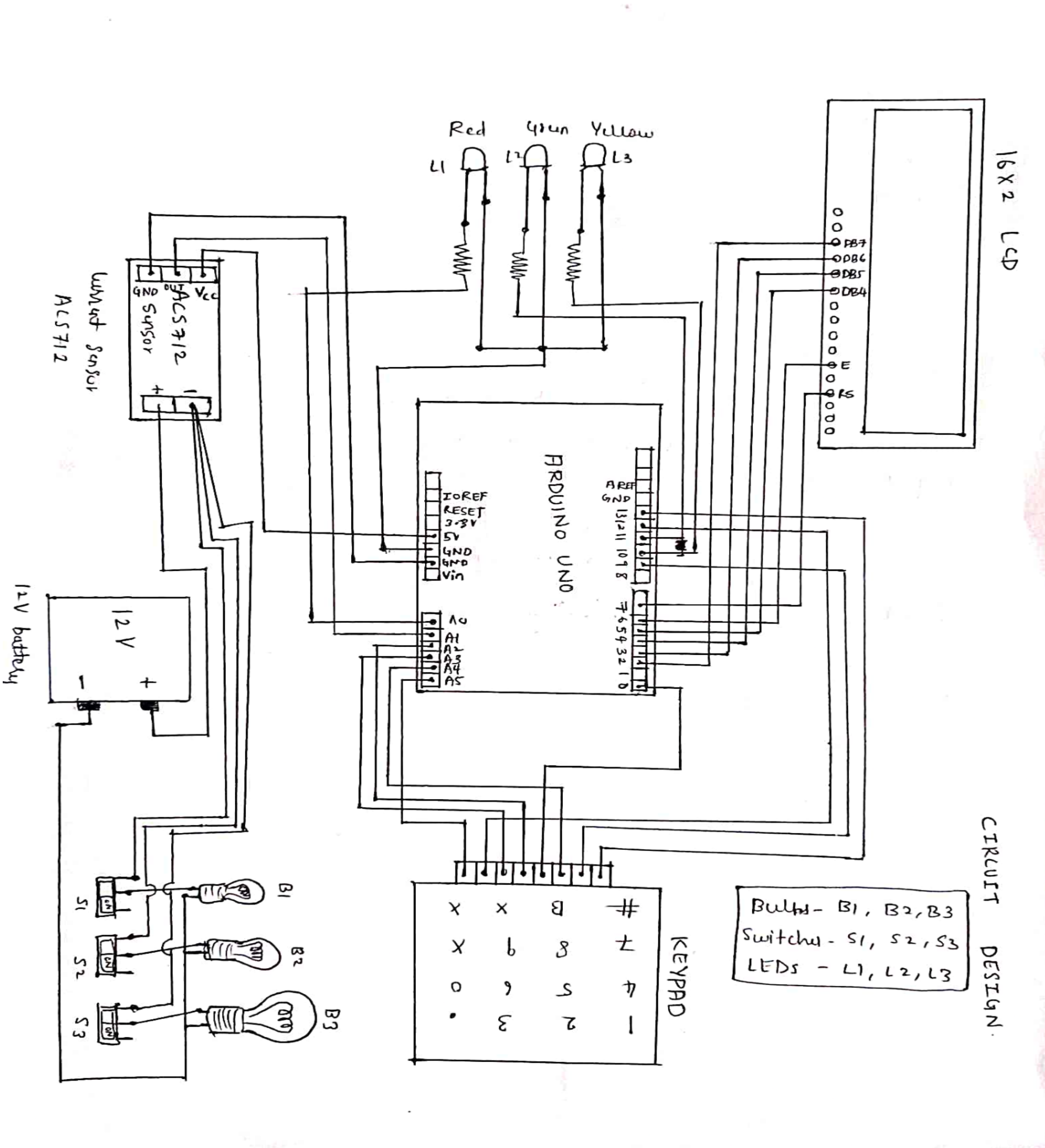
The ACS712 sensor is a highly accurate and versatile current sensing device. It utilizes the Hall effect to measure both AC and DC currents. With different models available for various current ranges, it provides precise measurements tailored to specific application needs. The sensor offers an analog output voltage that changes proportionally with the current passing through the conductor, allowing for accurate current measurement. It operates within a supply voltage range of 4.5V to 5.5V and can be easily integrated with microcontrollers. The ACS712 sensor is widely used in applications such as power monitoring, energy management systems, motor control, robotics, and industrial automation, making it an essential tool for accurate current measurement and control.

3. Display :

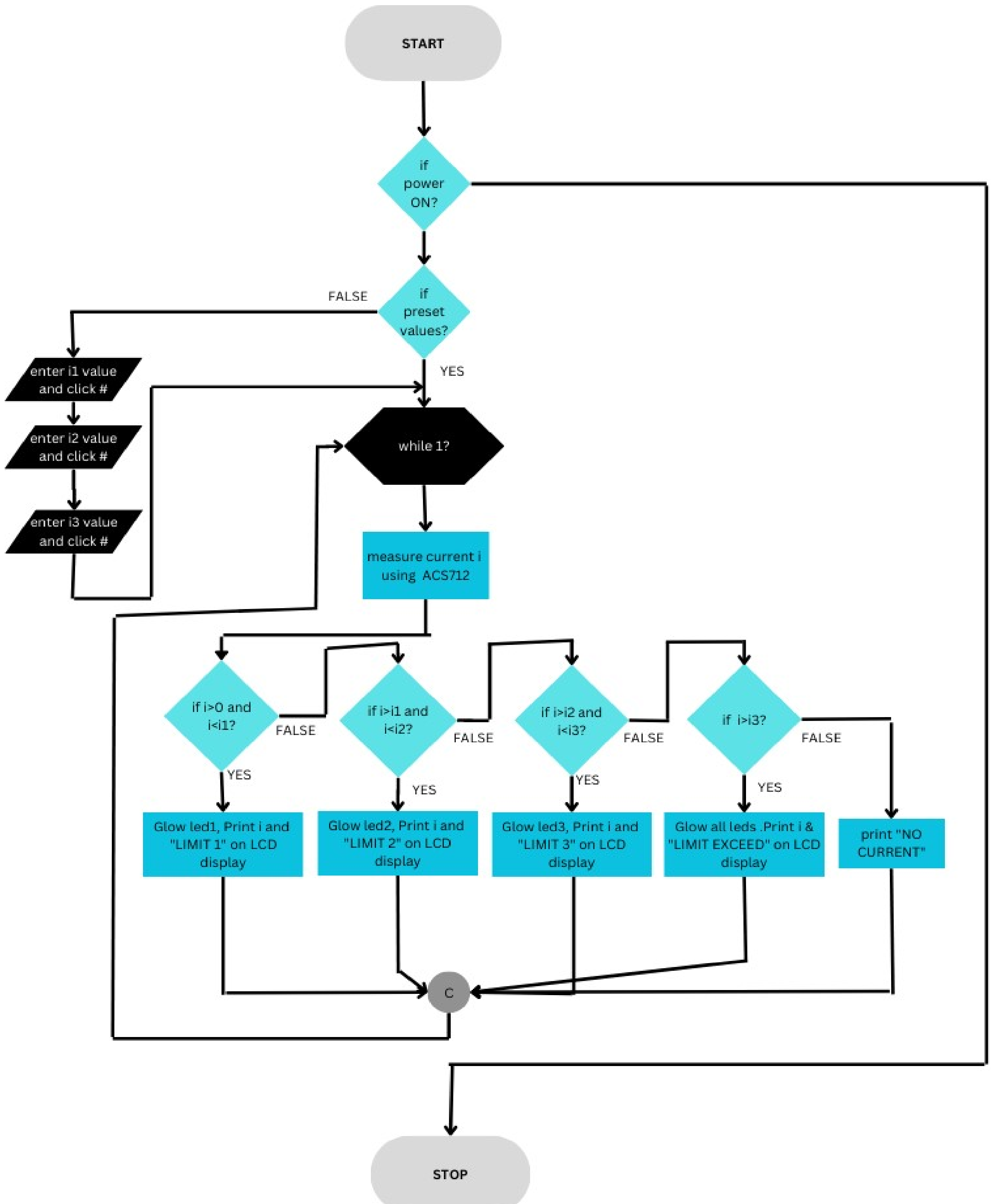
Displays are widely used in various electronic devices, ranging from simple digital clocks to sophisticated smartphones and televisions. They come in different technologies, each offering unique characteristics and capabilities.

One common type of display is the Liquid Crystal Display (LCD), which consists of a grid of tiny liquid crystal cells that can change their optical properties when an electric current is applied.

CIRCUIT DIAGRAM:



FLOWCHART:



START
if power ON?
FALSE
if preset values?
YES
enter i1 value and click #
enter i2 value and click #
enter i3 value and click #
while 1?
measure current i using ACS712
if i>0 and i<i1?
FALSE
YES
if i>i1 and i<i2?
FALSE
YES
if i>i2 and i<i3?
FALSE
YES
if i>i3?
FALSE
YES
Glow led1, Print i and "LIMIT 1" on LCD display
Glow led2, Print i and "LIMIT 2" on LCD display
Glow led3, Print i and "LIMIT 3" on LCD display
Glow all leds .Print i & "LIMIT EXCEED" on LCD display
print "NO CURRENT"
C
STOP

ARDUINO CODE:

```
/*This is an arduino code to execute a current sensing device using ACS712
interfaced with Arduino UNO and an LCD display.*/

#include <Keypad.h>         // Header file for a 4x4 Keypad
#include<EEPROM.h>          // Header file for accessing EEPROM memory from arduino
#include<LiquidCrystal.h>   // Header file for interfacing 16x2 LCD display

#define led1 10
#define led2 11
#define led3 A0

const byte ROWS = 4; //four rows
const byte COLS = 4; //four columns

// define the symbols on the buttons of the keypads
char keys[ROWS][COLS] = {
  {'1','4','7','#'},
  {'2','5','8','B'},
  {'3','6','9','C'},
  {'.','0','X','D'}
};

String num1,num2,num3;  /*String variables to store 3 current limits from the Keypad*/
double n1,n2,n3;        /*Double type variables to store the corresponding current limits*/
double x,y,z;
double num;

byte rowPins[ROWS] = {13, 9, A4, 0}; //connect to the row pinouts of the keypad
byte colPins[COLS] = {A2, A3, 12, A5}; //connect to the column pinouts of the keypad

/*Defining the Keypad*/
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );

LiquidCrystal lcd(7, 6, 5, 4, 3, 2);

const int currentPin = A1;
int sensitivity = 66;
int adcValue= 0;  /*Value at analog pin*/
int offsetVoltage = 2500;
double adcVoltage = 0;   /*Variable to store voltage value*/
double currentValue = 0;  /*Variable to store current value*/

int currentNum = 1;    /* No. of the current limit which is input from keypad*/
```

```
void presetvalue(void); /*Function to store the already stored values in EEPROM*/

void compare(void);     /*Function to compare the instantaneous value of current
                        to the limits*/
char key1;


void setup()
{
  // Setup code here, to run once:
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
  pinMode(led3, OUTPUT);
  lcd.begin(16, 2);       //initialize the LCD
  Serial.begin(9600);
  lcd.print(" Current Sensor ");
  lcd.setCursor(0,1);
  lcd.print("  with Arduino  ");
  delay(2000);
  lcd.clear();
  lcd.print("PRESET VALUES?");
  lcd.setCursor(0,1);
  lcd.print("CLICK B");
}

void loop()
{
  // Main code here, to run repeatedly:

  char key = keypad.getKey();    /*Input a key from Keypad*/
  if (key)
  {
    if(key >= '0' && key <= '9' || key == '.')  /*To print limits containing numbers 0 to 9
                                                and decimal point(.) */
    {
        lcd.clear();
        if(currentNum==1)                         /*For current limit 1*/
        {
        //if a number key or the decimal point key is pressed
          num1 += key;
        }
        else if(currentNum == 2)                  /*For current limit 2*/
        {
```

```
      num2 += key;
     }
     else if(currentNum == 3)                    /*For current limit 3*/
      num3 += key;
   }

   else if(key == 'B')       /*If B is clicked, the preset values are triggered*/
   {
     presetvalue();
   }

   else if( key == '#')     /*To jump to next current limit and after inputting all 3 values,
                            click # to trigger these values in the device*/
   {
     if(currentNum <= 3)
      currentNum++;
     else
     {
      n1=num1.toDouble();    /*Converting string num1 to double n1*/
      n2=num2.toDouble();    /*Converting string num2 to double n2*/
      n3=num3.toDouble();    /*Converting string num3 to double n3*/
      EEPROM.write(0,n1);    /*Storing n1 in EEPROM at location 0 in the memory*/
      EEPROM.write(10,n2);   /*Storing n2 in EEPROM at location 10 in the memory*/
      EEPROM.write(20,n3);   /*Storing n3 in EEPROM at location 20 in the memory*/
      x=EEPROM.read(0);      /*Reading n1 from memory*/
      y=EEPROM.read(10);     /*Reading n2 from memory*/
      z=EEPROM.read(20);     /*Reading n3 from memory*/
      lcd.print("I1:");
      lcd.print(n1);
      lcd.setCursor(0,1);
      lcd.print("I2:");
      lcd.print(n2);
      lcd.setCursor(8,1);
      lcd.print("I3:");
      lcd.print(n3);
      delay(500);
      compare();
     }

   }
  }
}
```

```
void presetvalue()
{
    /*Reading and displaying the preset values on LCD*/
    x=EEPROM.read(0);
    y=EEPROM.read(10);
    z=EEPROM.read(20);
    lcd.clear();
    lcd.print("PRESET VALUES:");
    lcd.clear();
    lcd.print("I1:");
    lcd.print(x);
    delay(2000);
    lcd.setCursor(0,1);
    lcd.print("I2:");
    lcd.print(y);
    delay(2000);
    lcd.setCursor(8,1);
    lcd.print("I3:");
    lcd.print(z);
    delay(2000);
    compare();
}

void compare()
{
  while(1)
  {

    adcValue = analogRead(currentPin);  /*Value at analog pin connected to the sensor*/
    adcVoltage = (adcValue / 1024.0) * 5000;  /*Voltage Value*/
    currentValue = ((adcVoltage - offsetVoltage) / sensitivity); /*Current Value*/
    lcd.clear();
    lcd.print("I=");
    lcd.print(currentValue);
    delay(1000);
    num=currentValue;
    if(num>0 && num<x)    /*LIMIT 1 condition*/
    {

      digitalWrite(led1,LOW);
      digitalWrite(led2,LOW);
      digitalWrite(led3,HIGH);
    }
    else if(num>x && num<y)   /*LIMIT 2 condition*/
```

```
      {

         digitalWrite(led3,LOW);
         digitalWrite(led2,LOW);
         digitalWrite(led1,HIGH);
      }
      else if(num>y && num<z)   /*LIMIT 3 condition*/
      {

           digitalWrite(led1,LOW);
           digitalWrite(led3,LOW);
           digitalWrite(led2,HIGH);
      }

      else if(num>z)
      {
         digitalWrite(led2,HIGH);
         digitalWrite(led1,HIGH);
         digitalWrite(led3,HIGH);
      }
        else         /*Zero current flowing through the circuit*/
      {
         lcd.setCursor(0,1);
         lcd.print(" NO CURRENT");
         digitalWrite(led1,LOW);
         digitalWrite(led2,LOW);
         digitalWrite(led3,LOW);
      }
    }
}
```

SCOPE FOR IMPROVEMENT:

While the current sensor device developed using the ACS712 for measuring current through a motor achieved accurate and reliable results, there are several areas where further improvements can be made.

Noise Filtering: One potential area for improvement is implementing enhanced noise filtering techniques. Motors often generate electrical noise and transient currents that can introduce inaccuracies in the current measurements.

Temperature Compensation: The ACS712's performance can be influenced by temperature variations. To improve accuracy, a temperature compensation mechanism can be incorporated into the device.

Real-Time Data Logging: The current sensor device can be enhanced by incorporating a real-time data logging feature. This would enable the device to store and record current measurements over a specific time period, allowing for subsequent analysis and monitoring of the motor's performance.

Expanded Current Range: While the ACS712 offers a wide current measurement range, specific motor applications may require even higher current capacities. To address this, a future improvement could involve selecting a different current sensor chip with an extended range or developing a circuitry capable of amplifying and measuring higher currents accurately.

By focusing on these areas for improvement, the current sensor device made using the ACS712 can be further enhanced to provide even more precise, reliable, and versatile current measurements for motor applications.