UNIVERSITY OF BATANGAS – LIPA CAMPUS COLLEGE OF ENGINEERING AND ARCHITECTURE COMPUTER ENGINEERING

LIQUID CRYSTAL DISPLAY IN ARDUINO

Experiment #2

MICROPROCESSOR SYSTEMS

NAME: TIAMSIM EDSEL, CASTILLO JOAQUIN, LASAT ALAN

STUDENT NUMBER:

DATE OF SUBMISSION:

April 13 2023

ENGR. CHARLES RAY JUANILLAS
PROFESSOR

Activity Questions:

1. Based on this activity, why do you need to use different library if your LCD Hardware have i2C display?

   Because this Library might not be compatible with existing sketches so I used different library that extremely similar to LiquidCrystal library that allows to control I2C displays with functions.

2. What is the purpose of the potentiometer?

   A potentiometer is a resistor with a manually adjustable resistance that controls the flow of electric current. It's used to adjust the LCD's contrast.

3. Can you display different symbols aside from the basic letter and characters?

   Yes, we can.

Results and Discussion:

1. Study the Figure 2.1. The figure shows the different pin available for all HD44780 types of LCD. This will be connected to the Arduino as shown in Figure 2.2 schematic diagram.

2. Study the Figure 2.2. Follow the connection between the LCD and Arduino.

3. Study the Figure 2.3. The figure shows the actual diagram of the connection of Arduino and the LCD.

4. If you the LCD has an i2C interface, there is a different wiring diagram which is shown in Figure 2.4.

5. If you the LCD has an i2C interface, there is a different wiring diagram which is shown in Figure 2.4. i2C Interface makes the Arduino and LCD connection simple.
   If using this kind of interface, make sure to download and include the "*LiquidCrystal I2C*" library in the Manage Libraries section of Arduino IDE.

6. Display the following to the LCD using your Arduino:

   a. Display the text "Hello World Arduino".

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
void setup()
{
```

```
lcd.begin();
lcd.setCursor(3,0);
lcd.print("Hello World");
}
void loop() {
lcd.setCursor(5,1);
lcd.print("Arduino");
}
```

b. Display your name in the first row and the course and section in the second row.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
void setup()
{
lcd.begin();
lcd.setCursor(1,0);
lcd.print("Edsel Lasat Castillo");
}
void loop() {
lcd.setCursor(5,1);
lcd.print("CPE3A");
}
```

c. Display your *Hugot Line* using the function scrollDisplayLeft() or scrollDisplayRight().

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 24, 2);
void setup() {
lcd.begin();
}
void loop() {
scroll_left();
delay(500);
}
//Scrolls message from left to right
void scroll_left(){
lcd.begin();
lcd.print("Mahal na mahal kita, bat moko iniwan ");
delay(400);
for (int positionCounter = 0; positionCounter < 24; positionCounter++) {
// scroll one position left:
```

```
lcd.scrollDisplayLeft();
delay(300);
}
delay(300);
{
}
}
```

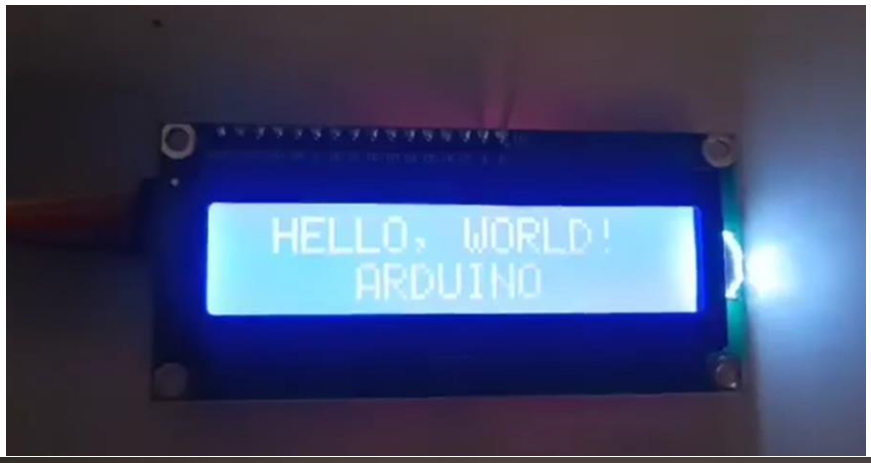
7. Create another Arduino program. This time, you will use button to change the display you created from Procedure 6. Every time the button was pressed, the text display will be changed.

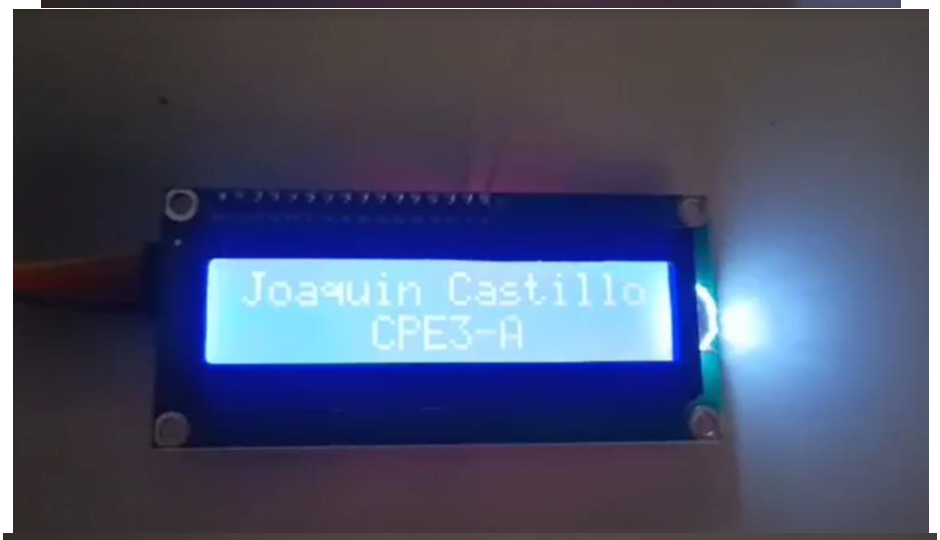
```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
int buttonpin = 9;
int passFlag = 0;

const unsigned long eventInterval = 300;
unsigned long previousTime = 0;

// Set the LCD address to 0x27 for a 16 chars and 2 line display
LiquidCrystal_I2C lcd(0x27, 16, 2);


int initial = 0;
int oldstate = 0;
int buttonstate = 0;

void setup() {
  // put your setup code here, to run once:

  pinMode(buttonpin, INPUT);


  // initialize the LCD
  lcd.begin(16,2);

  // Turn on the blacklight and print a message.
  lcd.backlight();
}

void loop() {
digitalWrite(5,HIGH);
  unsigned long currentTime = millis();

  buttonstate = digitalRead(buttonpin);
```

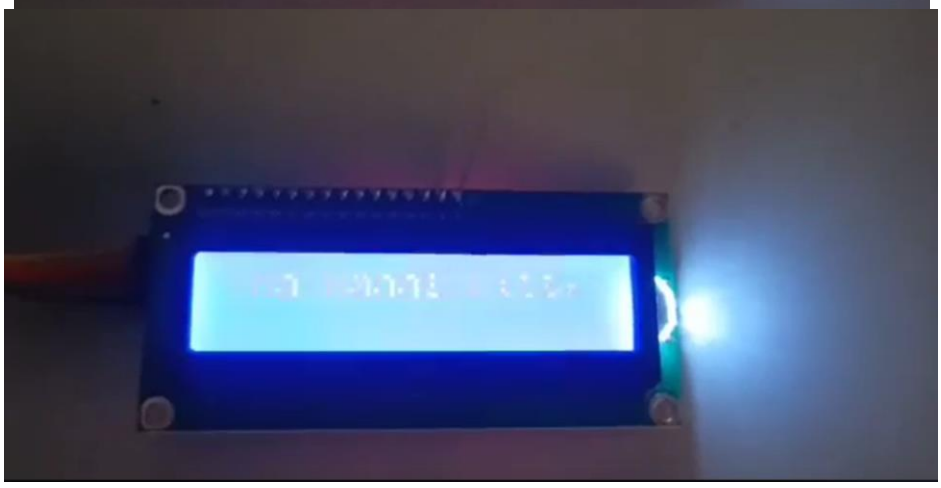
```
if (buttonstate == HIGH) {


  delay(50);
  buttonstate = digitalRead(buttonpin);
  if (buttonstate == LOW) {

    initial = oldstate + 1;
    passFlag = 0;
  }


}

else {
  delay(100);
}

switch (initial) {

  case 1:
    if (passFlag == 0) {
      hello();
      passFlag++;
       passFlag;              // Increment passFlag
    }
    oldstate = initial;
    break;

  case 2:

    if (passFlag == 0) {
      hi();
      passFlag++;
      passFlag;               // Increment passFlag
    }
    oldstate = initial;
    break;


  case 3:

    if (passFlag == 0) {
      hola();
      passFlag++;
       passFlag;                 // Increment passFlag
```

```
      }
      while(digitalRead(buttonpin) != HIGH){
        currentTime = millis();
        if (currentTime - previousTime >= eventInterval){
          lcd.scrollDisplayLeft();
          previousTime = currentTime;
        }
      }
      oldstate = initial;
      break;

    default:
      lcd.clear();
      oldstate = 0;
      break;
  }
}

void hello() {
  lcd.clear();
  lcd.print(" HELLO, WORLD!");
  lcd.setCursor(0, 1);
  lcd.print("    ARDUINO");
  lcd.setCursor(17, 1);
}

void hi() {
  lcd.clear();
  lcd.print("Joaquin Castillo");
  lcd.setCursor(0, 1);
  lcd.print("     CPE3-A");
  lcd.setCursor(17, 1);
}

void hola() {
  lcd.clear();
  lcd.print("Mahal na mahal kita, bat moko iniwan");

}
```

Documentation:







Conclusion:

We learned how to connect an LCD to an Arduino and use the LCD screen to show text characters. As an example, we connected an I2C LCD to an Arduino using a circuit and code. We used an LCD to display a character that we needed to portray, and it was a lot of fun displaying words for this project.

https://drive.google.com/drive/u/0/folders/12_oecHhvZQO9vvgzIPMmY-m6S9v57ygO?fbclid=IwAR2tzyFWV6vcbimNN1lVujlPmBnm-AW21NKf0K559rUe3hzhpdWoiKsATVQ