Initially, the proposed topic was a trash bin that uses PIC Microcontroller/Arduino technology to sort and manage waste efficiently. However, after further discussion, the idea was changed to a digitized coin bank. The new concept aims to leverage PIC Microcontroller/Arduino technology to create a modern coin bank that can store and manage coins. By using Arduino, the project will be able to incorporate features such as display screens and sensors to enhance the user experience. This shift in focus demonstrates the flexibility and adaptability of the team, and their ability to generate innovative ideas while leveraging existing technologies.

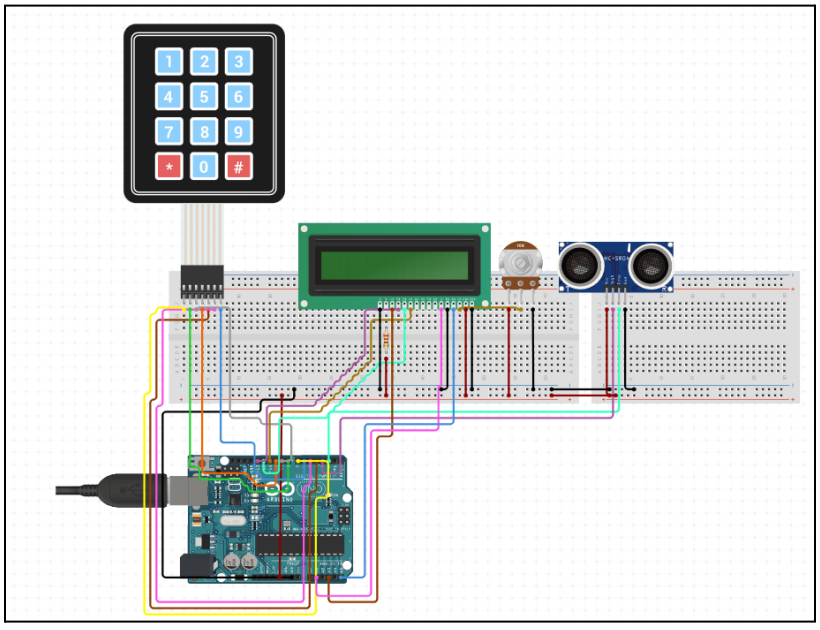
Overall, the digitized coin bank is a promising project that has the potential to provide a new and exciting way to manage and store loose change.

Section 2. Proposed Designs and Program

Design 1: Interfacing Ultrasonic Sensor, LCD, 4 x 4 Matrix Keypad with Arduino UNO

Component specifications:

- Arduino UNO
- 1602A LCD Module
- US-100 Ultrasonic Sensor
- 220Ω Resistor
- 10KΩ Potentiometer



Code in Arduino IDE

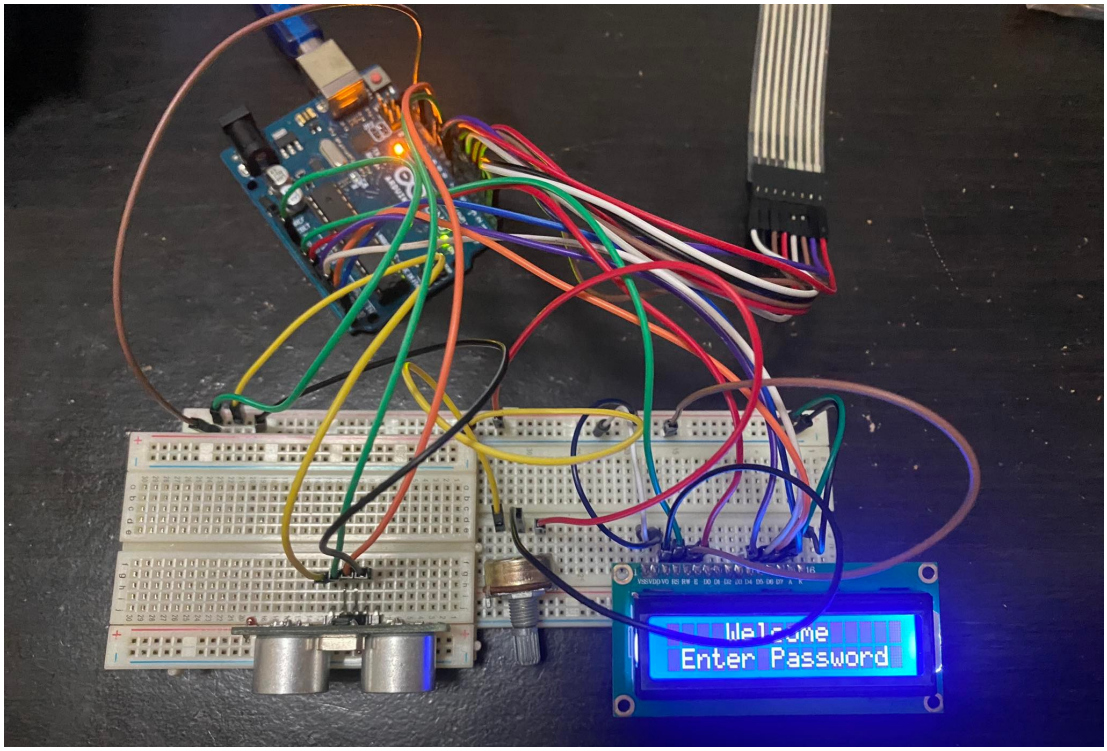
```
#include <LiquidCrystal.h>
#include <Keypad.h>
#define trigPin 9
#define echoPin 10

char* password ="1245";
int pozisyon = 0;

const byte rows = 4;
const byte cols = 4;

char keyMap [rows] [cols] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};

byte rowPins [rows] = {1, 2, 3, 4};
byte colPins [cols] = {5, 6, 7, 8};

Keypad myKeypad = Keypad( makeKeymap(keyMap), rowPins, colPins, rows, cols);

LiquidCrystal lcd (A0, A1, A2, A3, A4, A5);

void setup() {
  lcd.begin(16, 2);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}

void loop() {
```

```
  char whichKey = myKeypad.getKey(); //define which key is pressed with getKey

  lcd.setCursor(0, 0);
  lcd.print("    Welcome");
  lcd.setCursor(0, 1);
  lcd.print(" Enter Password");

  if (whichKey == '*' || whichKey == '#' || whichKey == 'A' || whichKey == 'B'
|| whichKey == 'C' || whichKey == 'D') {
    pozisyon = 0;

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("  Invalid Key!");
    delay(1000);
    lcd.clear();
  }

  if (whichKey == password[pozisyon]) {
    pozisyon++;
  }

  if (pozisyon == 4) {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("*** Verified ***");
    delay(3000);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("    Digitized");
    lcd.setCursor(0, 1);
    lcd.print("    Coin Bank");
    delay(3000);
```

```
    lcd.clear();
    while (1) { // infinite loop to continuously measure distance and display
it
      digitalWrite(trigPin, LOW);
      delayMicroseconds(2);
      digitalWrite(trigPin, HIGH);
      delayMicroseconds(10);
      digitalWrite(trigPin, LOW);
      long duration = pulseIn(echoPin, HIGH);
      int distance = duration * 0.034 / 2;
      lcd.setCursor(0, 0);
      if(distance > 15){
        lcd.setCursor(0,0);
        lcd.print("    LEVEL 1");
        lcd.setCursor(0,1);
        lcd.print("    SAVE UP");
        delay(3000);
        lcd.clear();
      }
      else if(distance > 10 && distance <= 15){
        lcd.setCursor(0,0);
        lcd.print("    LEVEL 2");
        lcd.setCursor(0,1);
        lcd.print("  KEEP IT UP!");
        delay(3000);
        lcd.clear();
      }
      else if(distance > 5 && distance <= 10){
        lcd.setCursor(0,0);
        lcd.print("    LEVEL 3");
        lcd.setCursor(0,1);
        lcd.print(" ALMOST THERE!");
        delay(3000);
```

```
        lcd.clear();
      }
      else if(distance > 0 && distance <= 5){
        lcd.setCursor(0,0);
        lcd.print("    LEVEL 4");
        lcd.setCursor(0,1);
        lcd.print("   GOOD JOB!");
        delay(3000);
        lcd.clear();

      }
      delay(1000);
    }
  }
}
```

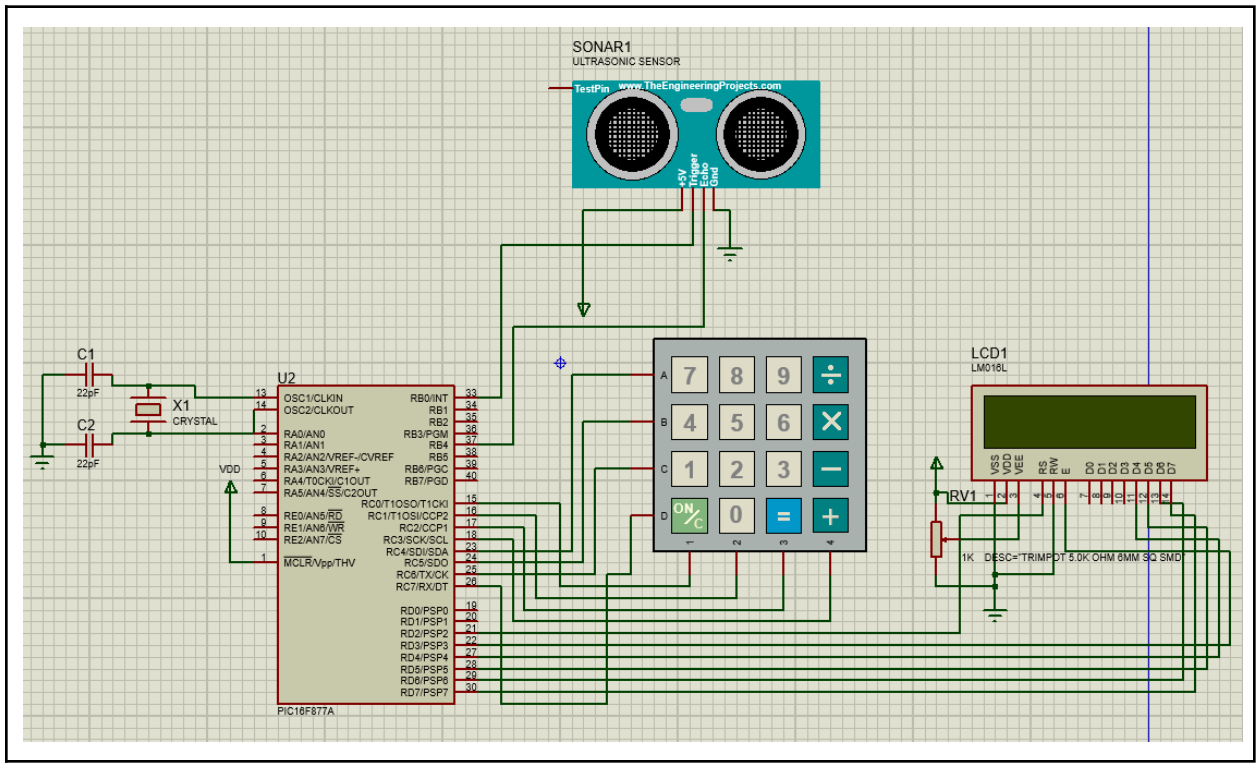
Actual prototype



Design 2: Interfacing Ultrasonic Sensor, LCD, 4 x 4 Matrix Keypad with PIC16F877A

Component Specification:

- PIC16F877A
- 4x4 Matrix KEYPAD
- 1602A LCD Module
- US-100 Ultrasonic Sensor
- Crystal Oscillator
- 10KΩ Potentiometer
- 22pF Capacitor



Code in MikroC Pro for PIC

```
// LCD module connections
sbit LCD_RS at RD2_bit;
sbit LCD_EN at RD3_bit;
sbit LCD_D4 at RD4_bit;
sbit LCD_D5 at RD5_bit;
sbit LCD_D6 at RD6_bit;
sbit LCD_D7 at RD7_bit;

sbit LCD_RS_Direction at TRISD2_bit;
sbit LCD_EN_Direction at TRISD3_bit;
sbit LCD_D4_Direction at TRISD4_bit;
sbit LCD_D5_Direction at TRISD5_bit;
sbit LCD_D6_Direction at TRISD6_bit;
sbit LCD_D7_Direction at TRISD7_bit;
// End LCD module connections

// 4x4 Keypad
char keypadPort at PORTC;
unsigned short kp, cnt, oldstate=0;
//End 4x4 Keypad

char* password = "1245";
int positionCnt = 0;

void main()
{
  //LCD
  int a;
  int distance;
  char txt[7];
  Lcd_Init();
  Lcd_Cmd(_LCD_CLEAR);          // Clear display
  Lcd_Cmd(_LCD_CURSOR_OFF);     // Cursor off

  TRISB = 0b00010000;           //RB4 as Input PIN (ECHO)

  Lcd_Out(1,1,"Digitized");
  Lcd_Out(2,1,"Coin Bank");
  //Lcd_Out(3,2," Electronics With ");
  //Lcd_Out(4,6," Shreyash");

  Delay_ms(500);
```

```
Lcd_Cmd(_LCD_CLEAR);
Lcd_Cmd(_LCD_CURSOR_OFF);     // Cursor off

while(1){
  LCD_Out(1,1,"Welcome!");
  LCD_Out(2,1,"Enter Password");

  //Keypad
  cnt = 0;
  Keypad_Init();
  do{
    kp = 0;
    do
      kp = Keypad_key_Click();
    while(!kp);
    switch (kp){
      case 1: kp = 49; break; //1
      case 2: kp = 50; break; //2
      case 3: kp = 51; break; //3
      case 4: kp = 65; break; //A
      case 5: kp = 52; break; //4
      case 6: kp = 53; break; //5
      case 7: kp = 54; break; //6
      case 8: kp = 66; break; //B
      case 9: kp = 55; break; //7
      case 10: kp = 56; break; //8
      case 11: kp = 57; break; //9
      case 12: kp = 67; break; //C
      case 13: kp = 42; break; //*
      case 14: kp = 48; break; //0
      case 15: kp = 35; break; //#
      case 16: kp = 68; break; //D
    }
    if(kp == '*' || kp == '#' || kp == 'A' || kp == 'A' || kp == 'B' || kp == 'C' || kp == 'D'){
      positionCnt = 0;
      Lcd_Cmd(_LCD_CLEAR);
      LCD_Out(1,1, "INVALID PASSCODE");
      LCD_Out(2,1, "Please try again...");
      Delay_ms(1000);
      Lcd_Cmd(_LCD_CLEAR);
    }
    if(kp == password[positionCnt]){
      positionCnt++;
    }
```

```
    if(positionCnt == 4){
      Lcd_Cmd(_LCD_CLEAR);
      LCD_Out(1,1, "****VERIFIED****");
      Delay_ms(3000);
      Lcd_Cmd(_LCD_CLEAR);

      //After Verified
      while(1){
        TMR1H = 0;              //Sets the Initial Value of Timer
        TMR1L = 0;              //Sets the Initial Value of Timer

        PORTB.F0 = 1;           //TRIGGER HIGH
        Delay_us(10);           //10uS Delay
        PORTB.F0 = 0;           //TRIGGER LOW

        while(!PORTB.F4);       //Waiting for Echo
        T1CON.F0 = 1;           //Timer Starts
        while(PORTB.F4);        //Waiting for Echo goes LOW
        T1CON.F0 = 0;           //Timer Stops

        a = (TMR1L | (TMR1H<<8));   //Reads Timer Value
        distance = a * 0.034 / 2; //Converts Time to Distance
        if(distance > 15){          //Check whether the result is valid or not
            LCD_Out(1,1, "LEVEL 1");
            LCD_Out(2,1, "SAVE UP!");
            Delay_ms(3000);
            Lcd_Cmd(_LCD_CLEAR);
        }
        else if(distance > 10 && distance <= 15){         //Check whether the result is valid or
not
            LCD_Out(1,1, "LEVEL 2");
            LCD_Out(2,1, "KEEP IT UP!");
            Delay_ms(3000);
            Lcd_Cmd(_LCD_CLEAR);
        }
        else if(distance > 5 && distance <= 10){          //Check whether the result is valid or
not
            LCD_Out(1,1, "LEVEL 3");
            LCD_Out(2,1, "GOOD JOB!");
            Delay_ms(3000);
            Lcd_Cmd(_LCD_CLEAR);
        }
        else if(distance > 0 && distance <= 5){         //Check whether the result is valid or not
            LCD_Out(1,1, "LEVEL 4");
```

```
                LCD_Out(2,1, "GOOD JOB!");
                Delay_ms(3000);
                Lcd_Cmd(_LCD_CLEAR);
            }
            Delay_ms(1000);
        }
      }
    } while (1);
  }
}
```

Section 3. Proposed Testing Methodology and Data Gathering Structure

Experimental Procedure

1. Place a number of coins in the coin bank and document the initial measurement of distance from the ultrasonic sensor to the top of the coin stack.
2. Enter the system's password and authenticate access using the keypad.
3. Confirm that the LCD display accurately displays the security authentication and distance level.
4. Add more coins to the coin bank and document the new measurement of the distance between the ultrasonic sensor and the top of the coin stack.
5. Repeat step 4 multiple times, adding varying quantities of coins each time, and documenting the distance measurements each time.
6. To ascertain the ultrasonic sensor's precision, compare the measured distances to the actual distances.
7. Check that the keypad and LCD display are operating properly.

Data Analysis

1. For each trial, calculate the difference between the measured and actual distances.
    a. distance > 15
    b. distance > 10 && distance <= 15
    c. distance > 5 && distance <= 10
    d. distance > 0 && distance <= 5
2. Determine if the ultrasonic sensor is precise enough for the intended application by comparing the mean deviation to an acceptable margin of error.
3. Check that the keypad and LCD display are operating properly.
    a. distance > 15
        i. The LCD will display “Level 1” and “SAVE UP”
    b. distance > 10 && distance <= 15

i. The LCD will display “Level 2” and “KEEP IT UP!”

c. distance > 5 && distance <= 10
   i. The LCD will display “Level 3” and “ALMOST THERE!”

d. distance > 0 && distance <= 5
   i. The LCD will display “Level 4” and “GOOD JOB!”

Section 4. Schedule of Activities in terms of Day Basis

Activity	**Project Work Days**						
	1	**2**	**3**	**4**	**5**	**6**	**7**
Proposal	Done						
Design Schematic (Arduino UNO)		Done					
Program (Arduino IDE)		Done					
Fabrication (Arduino IDE)		Done					
Design Schematic (PIC16F877A)			In Progress				
Program (PIC16F877A)			In Progress				
Fabrication (PIC16F877A)			In Progress				
Test Evaluation (Arduino UNO)		Done					

Test Evaluation (PIC16F877A)			In Progress				
Result Analysis (Arduino UNO)		Done					
Result Analysis (PIC16F877A)			In Progress				
Report Writing				Pending	Pending	Pending	
Submission of Project							Pending