Portfolio - Practical - Report

ENG20009 – Engineering Technology Inquiry Project

Dang Khoa Le

ID: 103844421

Table of Contents

Abstract:

The following laboratory portfolio practicals have chosen Option 2 - Simulation - Arduino Mega (on WOKWI). Followings are the report showing pseudo codes or flowcharts related to each weeks' practicals.

A. Lab 2 Report Pseudo and Flowchart:

1. Pass and Pass Plus tasks

Pass: Using 4 push buttons to control the behaviour of 3 LEDs on the bar graph or buzzer:

- The first push button should toggle the first LED/buzzer on and off.

Pass Plus: Continue from Pass question above with the following tasks:

- The second button should increase the speed of the second LED's blinking.
- The third button should decrease the speed of the second LED's blinking.
- The fourth button should toggle the brightness of the third LED between high and low.

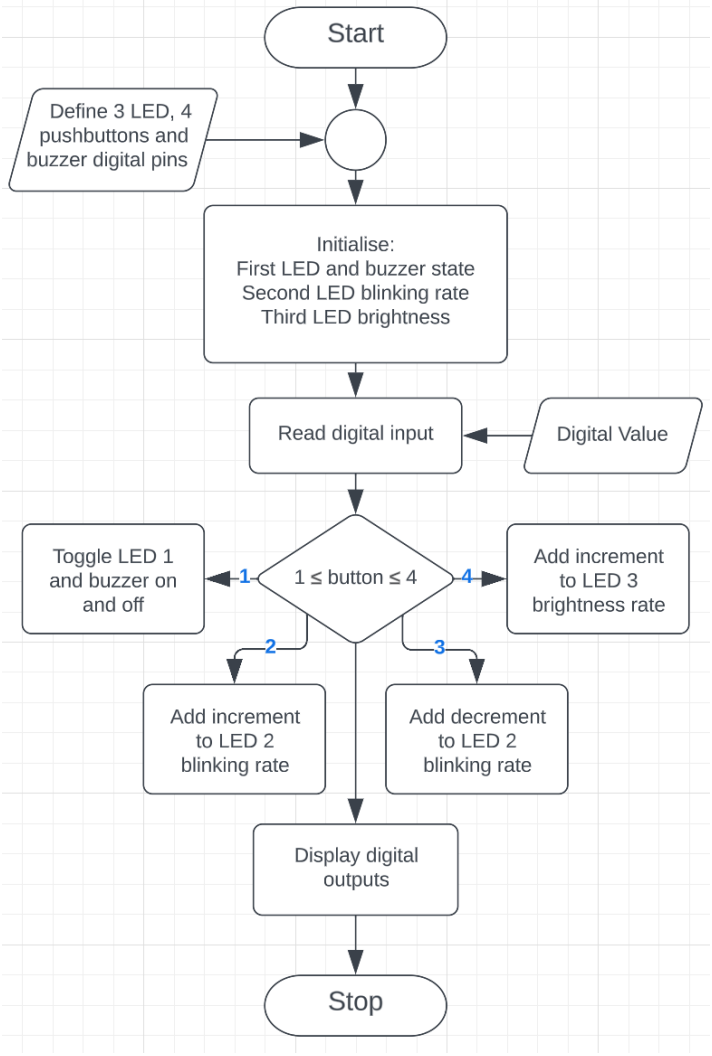
Arduino Sketch:

```
const int led1Pin = 13;
const int led2Pin = 12;
const int led3Pin = 11;
const int buzzerPin = 7;
```

```
const int button1Pin = 18;
const int button2Pin = 19;
const int button3Pin = 20;
const int button4Pin = 21;

bool led1On = false;
int led2Speed = 500;
int led3Brightness = 255;

void setup() {
  pinMode(led1Pin, OUTPUT);
  pinMode(led2Pin, OUTPUT);
  pinMode(led3Pin, OUTPUT);
  pinMode(buzzerPin, OUTPUT);

  pinMode(button1Pin, INPUT_PULLUP);
  pinMode(button2Pin, INPUT_PULLUP);
  pinMode(button3Pin, INPUT_PULLUP);
  pinMode(button4Pin, INPUT_PULLUP);
}

void loop() {
  // Toggle the First Led and Buzzer
  if (digitalRead(button1Pin) == LOW) {
    led1On = !led1On;
    digitalWrite(led1Pin, led1On ? HIGH : LOW);
    digitalWrite(buzzerPin, led1On ? HIGH : LOW);
    delay(100);
  }

  // Increase the Second Led speed
  if (digitalRead(button2Pin) == LOW) {
    led2Speed = max(50, led2Speed - 50);
    delay(100);
  }

  // Decrease the Second Led speed
  if (digitalRead(button3Pin) == LOW) {
    led2Speed = min(1000, led2Speed + 50);
    delay(100);
  }

  // Toggle the Third Led brightness
  if (digitalRead(button4Pin) == LOW) {
    led3Brightness = led3Brightness == 255 ? 50 : 255;
    delay(100);
  }

  // Blink Led at the current speed
  digitalWrite(led2Pin, HIGH);
```

```
  delay(led2Speed / 2);
  digitalWrite(led2Pin, LOW);
  delay(led2Speed / 1);

  // Set LED3 brightness
  analogWrite(led3Pin, led3Brightness);
}
```

Flowchart:



2. Credit, Distinction and High Distinction tasks

Credit: Using a dip switch, create a solution to display the following characters on the 8x8 LED matrix. Make a proper connection for 8×8 LED Matrix with resistors to the microcontroller. The LED matrix have to be constructed manually by using 64 units of LED as 8×8 LED Matrix. Characters to be displayed 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

Distinction: Continuing from Credit Task for designing LED matrix with a new feature. Use a pushbutton, when it is pressed, the character will move from left to right.

High Distinction: Continue from Distinction Task for designing LED matrix with additional feature to be added with the same pushbutton to change up to 3 modes for speed of the moving character including slow, medium and fast.

Arduino Sketch:

```
#include "ScrollingText8x8Display.h"
ScrollingText8x8Display render;

const int switch1Pin = 22;
const int switch2Pin = 23;
const int switch3Pin = 24;
const int switch4Pin = 25;
const int switch5Pin = 26;

const int button1Pin = 42;
const int button2Pin = 43;
const int button3Pin = 44;

void setup()
{
  pinMode(switch1Pin, INPUT_PULLUP);
  pinMode(switch2Pin, INPUT_PULLUP);
  pinMode(switch3Pin, INPUT_PULLUP);
  pinMode(switch4Pin, INPUT_PULLUP);
  pinMode(switch5Pin, INPUT_PULLUP);

  pinMode(button1Pin, INPUT_PULLUP);
  pinMode(button2Pin, INPUT_PULLUP);
  pinMode(button3Pin, INPUT_PULLUP);

  byte displayRowPins[] = { 21, 20, 19, 18, 17, 16, 15, 14 };
  byte displayColumnPins[] = { 13, 12, 11, 10, 9, 8, 7, 6 };

  ScrollingDirection scrollingDirection = LEFT_TO_RIGHT;
  CharacterOrientation characterOrientation = BOTTOM;
  render.init(displayRowPins, displayColumnPins, scrollingDirection,
characterOrientation);
}

void loop()
{
  bool switch1On = digitalRead(switch1Pin) == LOW;
  bool switch2On = digitalRead(switch2Pin) == LOW;
  bool switch3On = digitalRead(switch3Pin) == LOW;
  bool switch4On = digitalRead(switch4Pin) == LOW;
  bool switch5On = digitalRead(switch5Pin) == LOW;

  bool button1On = digitalRead(button1Pin) == LOW;
  bool button2On = digitalRead(button2Pin) == LOW;
  bool button3On = digitalRead(button3Pin) == LOW;

//using binary to hexadecimal conversion to operate letter 0-F
//case 0 switch5
if (digitalRead(switch5Pin) == LOW) {
```

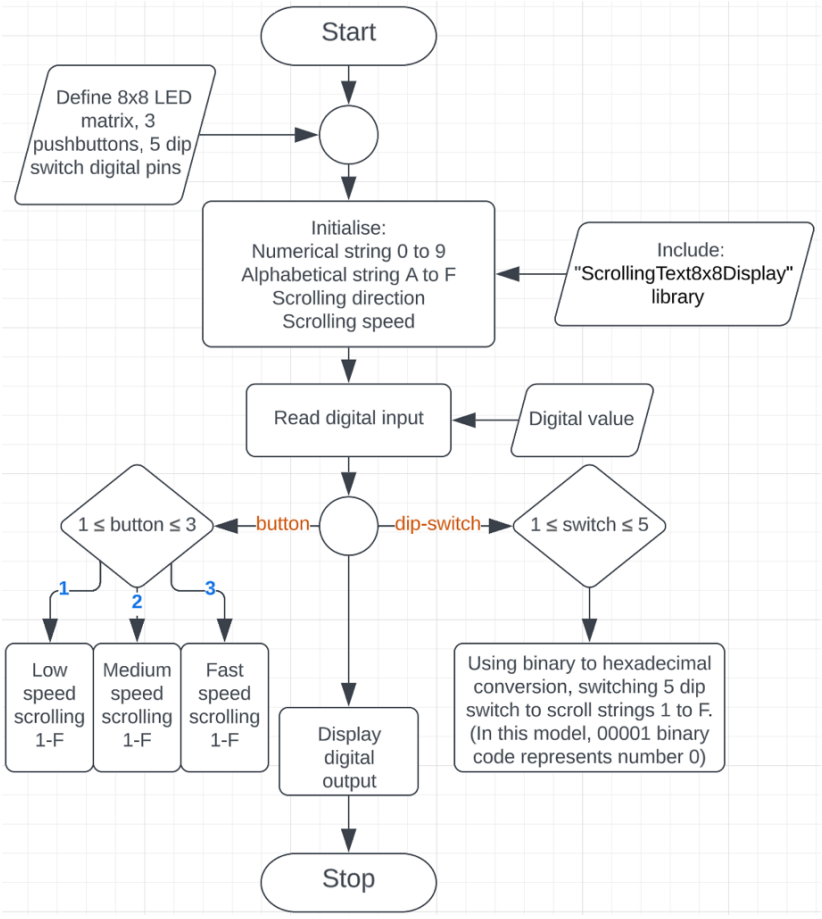
```
float scrollingSpeed = 80;
render.displayText("0", scrollingSpeed);
}

//case 1 0001
if (digitalRead(switch1Pin) == HIGH && digitalRead(switch2Pin) == HIGH &&
digitalRead(switch3Pin) == HIGH && digitalRead(switch4Pin) == LOW) {
float scrollingSpeed = 80;
render.displayText("1", scrollingSpeed);
}

//case 2 0010
if (digitalRead(switch1Pin) == HIGH && digitalRead(switch2Pin) == HIGH &&
digitalRead(switch3Pin) == LOW && digitalRead(switch4Pin) == HIGH) {
float scrollingSpeed = 80;
render.displayText("2", scrollingSpeed);
}

//case 3 0011
if (digitalRead(switch1Pin) == HIGH && digitalRead(switch2Pin) == HIGH &&
digitalRead(switch3Pin) == LOW && digitalRead(switch4Pin) == LOW) {
float scrollingSpeed = 80;
render.displayText("3", scrollingSpeed);
}

//case 4 0100
if (digitalRead(switch1Pin) == HIGH && digitalRead(switch2Pin) == LOW &&
digitalRead(switch3Pin) == HIGH && digitalRead(switch4Pin) == HIGH) {
float scrollingSpeed = 80;
render.displayText("4", scrollingSpeed);
}

//case 5 0101
if (digitalRead(switch1Pin) == HIGH && digitalRead(switch2Pin) == LOW &&
digitalRead(switch3Pin) == HIGH && digitalRead(switch4Pin) == LOW) {
float scrollingSpeed = 80;
render.displayText("5", scrollingSpeed);
}

//case 6 0110
if (digitalRead(switch1Pin) == HIGH && digitalRead(switch2Pin) == LOW &&
digitalRead(switch3Pin) == LOW && digitalRead(switch4Pin) == HIGH) {
float scrollingSpeed = 80;
render.displayText("6", scrollingSpeed);
}

//case 7 0111
if (digitalRead(switch1Pin) == HIGH && digitalRead(switch2Pin) == LOW &&
digitalRead(switch3Pin) == LOW && digitalRead(switch4Pin) == LOW) {
float scrollingSpeed = 80;
render.displayText("7", scrollingSpeed);
```

```
}

//case 8 1000
if (digitalRead(switch1Pin) == LOW && digitalRead(switch2Pin) == HIGH &&
digitalRead(switch3Pin) == HIGH && digitalRead(switch4Pin) == HIGH) {
float scrollingSpeed = 80;
render.displayText("8", scrollingSpeed);
}

//case 9 1001
if (digitalRead(switch1Pin) == LOW && digitalRead(switch2Pin) == HIGH &&
digitalRead(switch3Pin) == HIGH && digitalRead(switch4Pin) == LOW) {
float scrollingSpeed = 80;
render.displayText("9", scrollingSpeed);
}

//case A 1010
if (digitalRead(switch1Pin) == LOW && digitalRead(switch2Pin) == HIGH &&
digitalRead(switch3Pin) == LOW && digitalRead(switch4Pin) == HIGH) {
float scrollingSpeed = 80;
render.displayText("A", scrollingSpeed);
}

//case B 1011
if (digitalRead(switch1Pin) == LOW && digitalRead(switch2Pin) == HIGH &&
digitalRead(switch3Pin) == LOW && digitalRead(switch4Pin) == LOW) {
float scrollingSpeed = 80;
render.displayText("B", scrollingSpeed);
}

//case C 1100
if (digitalRead(switch1Pin) == LOW && digitalRead(switch2Pin) == LOW &&
digitalRead(switch3Pin) == HIGH && digitalRead(switch4Pin) == HIGH) {
float scrollingSpeed = 80;
render.displayText("C", scrollingSpeed);
}

//case D 1101
if (digitalRead(switch1Pin) == LOW && digitalRead(switch2Pin) == LOW &&
digitalRead(switch3Pin) == HIGH && digitalRead(switch4Pin) == LOW) {
float scrollingSpeed = 80;
render.displayText("D", scrollingSpeed);
}

//case E 1110
if (digitalRead(switch1Pin) == LOW && digitalRead(switch2Pin) == LOW &&
digitalRead(switch3Pin) == LOW && digitalRead(switch4Pin) == HIGH) {
float scrollingSpeed = 80;
render.displayText("E", scrollingSpeed);
}
```

```
//case F 1111
if (digitalRead(switch1Pin) == LOW && digitalRead(switch2Pin) == LOW &&
digitalRead(switch3Pin) == LOW && digitalRead(switch4Pin) == LOW)  {
float scrollingSpeed = 80;
render.displayText("F", scrollingSpeed);
}

//Switch off all dip-switch
//button 1 red for low speed scrolling
if (digitalRead(button1Pin) == LOW) {
float scrollingSpeed = 20;
render.displayText("0", scrollingSpeed);
render.displayText("1", scrollingSpeed);
render.displayText("2", scrollingSpeed);
render.displayText("3", scrollingSpeed);
render.displayText("4", scrollingSpeed);
render.displayText("5", scrollingSpeed);
render.displayText("6", scrollingSpeed);
render.displayText("7", scrollingSpeed);
render.displayText("8", scrollingSpeed);
render.displayText("9", scrollingSpeed);
render.displayText("A", scrollingSpeed);
render.displayText("B", scrollingSpeed);
render.displayText("C", scrollingSpeed);
render.displayText("D", scrollingSpeed);
render.displayText("E", scrollingSpeed);
render.displayText("F", scrollingSpeed);
delay(100);
}

//button 2 yellow for medium speed scrolling
if (digitalRead(button2Pin) == LOW) {
float scrollingSpeed = 50;
render.displayText("0", scrollingSpeed);
render.displayText("1", scrollingSpeed);
render.displayText("2", scrollingSpeed);
render.displayText("3", scrollingSpeed);
render.displayText("4", scrollingSpeed);
render.displayText("5", scrollingSpeed);
render.displayText("6", scrollingSpeed);
render.displayText("7", scrollingSpeed);
render.displayText("8", scrollingSpeed);
render.displayText("9", scrollingSpeed);
render.displayText("A", scrollingSpeed);
render.displayText("B", scrollingSpeed);
render.displayText("C", scrollingSpeed);
render.displayText("D", scrollingSpeed);
render.displayText("E", scrollingSpeed);
render.displayText("F", scrollingSpeed);
delay(100);
}
```

```
//button 3 green for fast speed scrolling
if (digitalRead(button3Pin) == LOW) {
float scrollingSpeed = 90;
render.displayText("0", scrollingSpeed);
render.displayText("1", scrollingSpeed);
render.displayText("2", scrollingSpeed);
render.displayText("3", scrollingSpeed);
render.displayText("4", scrollingSpeed);
render.displayText("5", scrollingSpeed);
render.displayText("6", scrollingSpeed);
render.displayText("7", scrollingSpeed);
render.displayText("8", scrollingSpeed);
render.displayText("9", scrollingSpeed);
render.displayText("A", scrollingSpeed);
render.displayText("B", scrollingSpeed);
render.displayText("C", scrollingSpeed);
render.displayText("D", scrollingSpeed);
render.displayText("E", scrollingSpeed);
render.displayText("F", scrollingSpeed);
delay(100);
}
}
```

Flowchart:



B. Lab 3 Report Pseudo and Flowchart:

1. Pass and Pass Plus tasks

Pass: Create menu using the UART and display in the serial monitor which has selection at the following:

- The first menu should toggle the first buzzer on and off.

Pass Plus: Continue from the Pass question above with more selection below:

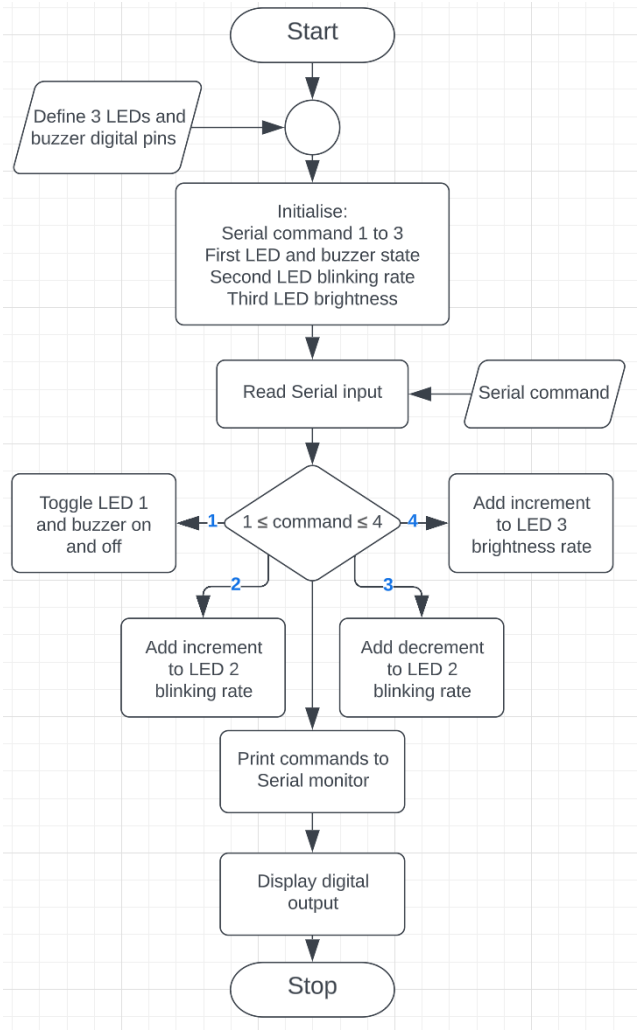
- The second menu should increase the speed of the second LED's blinking.
- The third menu should decrease the speed of the second LED's blinking.
- The fourth menu should toggle the brightness of the third LED between high and low.

Arduino Sketch:

```
const int buzzerPin = 13;
const int led1Pin = 3;
const int led2Pin = 2;
const int led3Pin = 1;

int speed = 1000;
int brightness = 0;
bool buzzerOn = false;
bool led1On = false;

void setup() {
  Serial.begin(9600);
  pinMode(buzzerPin, OUTPUT);
  pinMode(led1Pin, OUTPUT);
  pinMode(led2Pin, OUTPUT);
  pinMode(led3Pin, OUTPUT);
  Serial.println("Menu:\n1 - Toggle Led and Buzzer\n2 - Increase Blinking Speed\n3
- Decrease Blinking Speed\n4 - Set LED3 Brightness Level");
}

void loop() {
  if (Serial.available() > 0) {
    char command = Serial.read();
    // Menu option 1
    if (command == '1') {
      buzzerOn = !buzzerOn;
      led1On = !led1On;
      if (led1On) {
        digitalWrite(led1Pin, HIGH);
        Serial.println("Toggle the First Led and Buzzer on");
      } else {
        digitalWrite(led1Pin, LOW);
        Serial.println("Toggle the First Led and Buzzer off");
      }
      if (buzzerOn) {
        digitalWrite(buzzerPin, HIGH);
      } else {
```

```
        digitalWrite(buzzerPin, LOW);
      }
    // Menu option 2
    } else if (command == '2') {
      speed -= 100;
      if (speed < 100) {
        speed = 100;
      }
      Serial.print("Increase the Second Led Speed to: ");
      Serial.println(speed);
    // Menu option 3
    } else if (command == '3') {
      speed += 100;
      if (speed > 2000) {
        speed = 2000;
      }
      Serial.print("Decrease the Second Led Speed to: ");
      Serial.println(speed);
    // Menu option 4
    } else if (command == '4') {
      brightness += 25;
      if (brightness > 255) {
        brightness = 0;
      analogWrite(led3Pin, brightness);
      }
      Serial.print("The Third Led Brightness is: ");
      Serial.println(brightness);
    }
  }
// LED blinking rate
  static unsigned long previousMillis = 0;
  unsigned long currentMillis = millis();
  if (currentMillis - previousMillis >= speed) {
    previousMillis = currentMillis;
    digitalWrite(led2Pin, !digitalRead(led2Pin));
  }
}
```

Flowchart:



2. Credit task

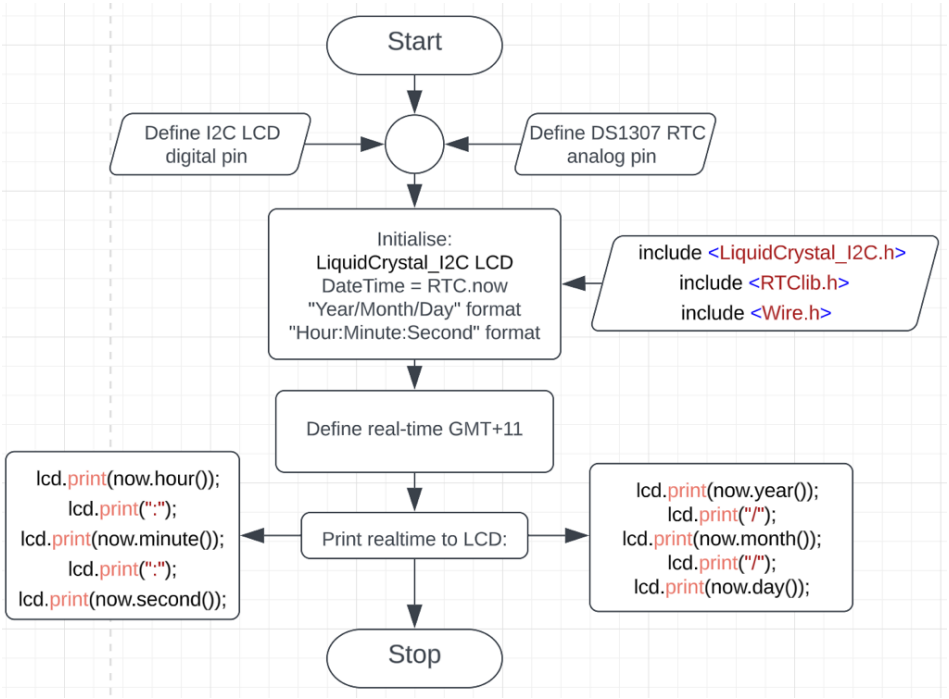
Credit: Using the RTC and GLCD (SSD1306 OLED Display or TFT-LCD display). Create a digital clock by reading the data from RTC and displaying it on LCD.

Arduino Sketch:

```
#include <LiquidCrystal_I2C.h>
#include <RTClib.h>
#include <Wire.h>

#define SERIAL_OPTION 0

LiquidCrystal_I2C lcd(0x27,16,2);
RTC_DS1307 RTC;

void setup() {
  if (SERIAL_OPTION) Serial.begin(9600);
// Initialize LCD.
  lcd.init();
  lcd.backlight();
```

```
  RTC.begin();

// Adjusting to Melbourne GMT+11 time.
  RTC.adjust(DateTime(2023,03,21,10,39,30));
}

void loop() {
  // Get the current date and time from the RTC.
  DateTime now=RTC.now();

  if (SERIAL_OPTION) {
    Serial.print(now.year()); Serial.print("/");
    Serial.print(now.month()); Serial.print("/");
    Serial.print(now.day()); Serial.print(" ");

    Serial.print(now.hour()); Serial.print(":");
    Serial.print(now.minute()); Serial.print(":");
    Serial.print(now.second()); Serial.print("\n");
  }

  // Format the date and time and print:
  lcd.setCursor(0,0); lcd.print("DATE: ");
  lcd.print(now.year()); lcd.print("/"); lcd.print(now.month()); lcd.print("/");
lcd.print(now.day());
  lcd.setCursor(0,1); lcd.print("TIME: ");
  lcd.print(now.hour()); lcd.print(":"); lcd.print(now.minute()); lcd.print(":");
lcd.print(now.second());

  // Wait for 1 second before updating the display.
  delay(1000);
}
```

Flowchart:



3. Distinction

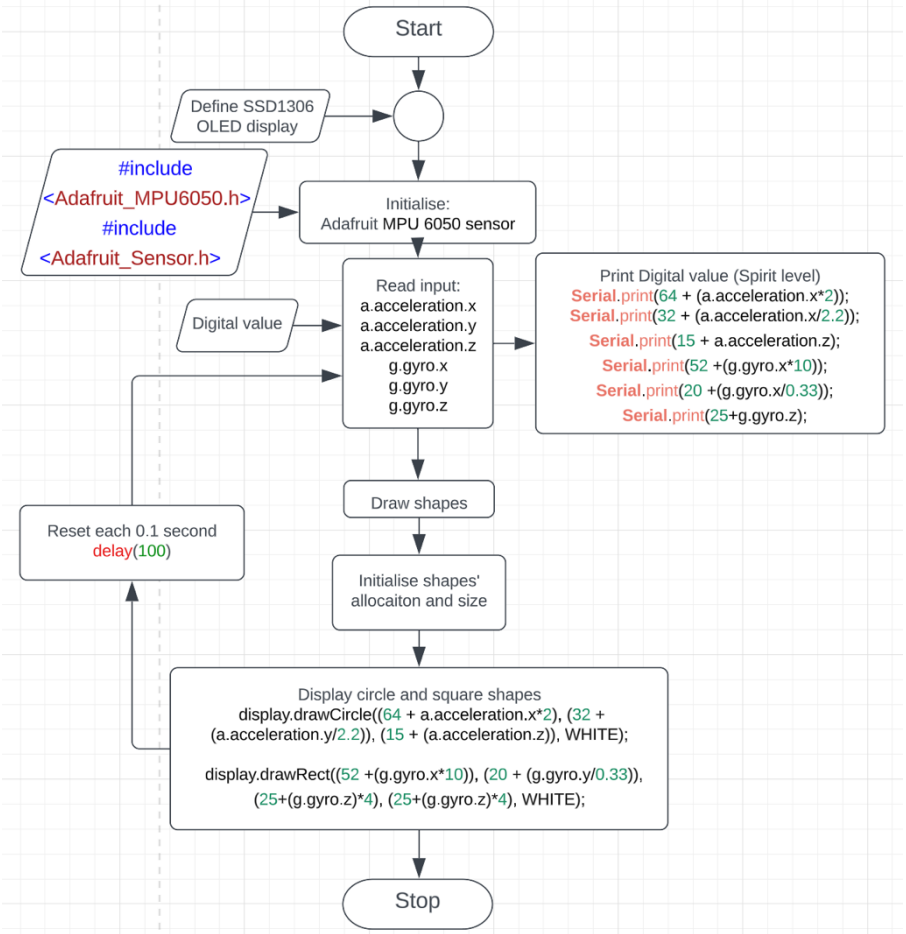
Distinction: Using the accelerometer of the IMU (MPU6050) create a spirit level that displays the current angle away from level on the LCD display (SSD1306 OLED Display or TFT-LCD display) and provide a graphic that will assist with levelling the board.

Arduino Sketch:

```
//Access: https://wokwi.com/projects/360766818972243969

#include <Wire.h>
#include <Adafruit_SSD1306.h>
#include <Adafruit_GFX.h>
#include <Adafruit_MPU6050.h>
#include <Adafruit_Sensor.h>


#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
Adafruit_MPU6050 mpu;

void setup() {
  Serial.begin(9600);
  Wire.begin();

  if (!mpu.begin()) {
    Serial.println("Failed to find MPU6050 chip");
```

```
    while (1) {
      delay(10);
    }
  }

  mpu.setAccelerometerRange(MPU6050_RANGE_16_G);
  mpu.setGyroRange(MPU6050_RANGE_250_DEG);
  mpu.setFilterBandwidth(MPU6050_BAND_21_HZ);

  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);

}


void loop() {
  circle();
  sensors_event_t a, g, temp;
  mpu.getEvent(&a, &g, &temp);
  Serial.print(64 + (a.acceleration.x*2));
  Serial.print(",");
  Serial.print(32 + (a.acceleration.x/2.2));
  Serial.print(",");
  Serial.print(15 + a.acceleration.z);
  Serial.print(", ");
  Serial.print(52 +(g.gyro.x*10));
  Serial.print(",");
  Serial.print(20 +(g.gyro.x/0.33));
  Serial.print(",");
  Serial.print(25+g.gyro.z);
  Serial.println("");

    display.clearDisplay();
    delay(100);
    display.display();
    display.clearDisplay();
}

void circle(){
  sensors_event_t a, g, temp;
  mpu.getEvent(&a, &g, &temp);

  display.drawCircle((64 + a.acceleration.x*2), (32 + (a.acceleration.y/2.2)), (15
+ (a.acceleration.z)), WHITE);
  display.drawRect((52 +(g.gyro.x*10)), (20 + (g.gyro.y/0.33)), (25+(g.gyro.z)*4),
(25+(g.gyro.z)*4), WHITE);

  display.display();
}
```

Flowchart:



C. Lab 4 Report Pseudo and Flowchart:

1. Pass task

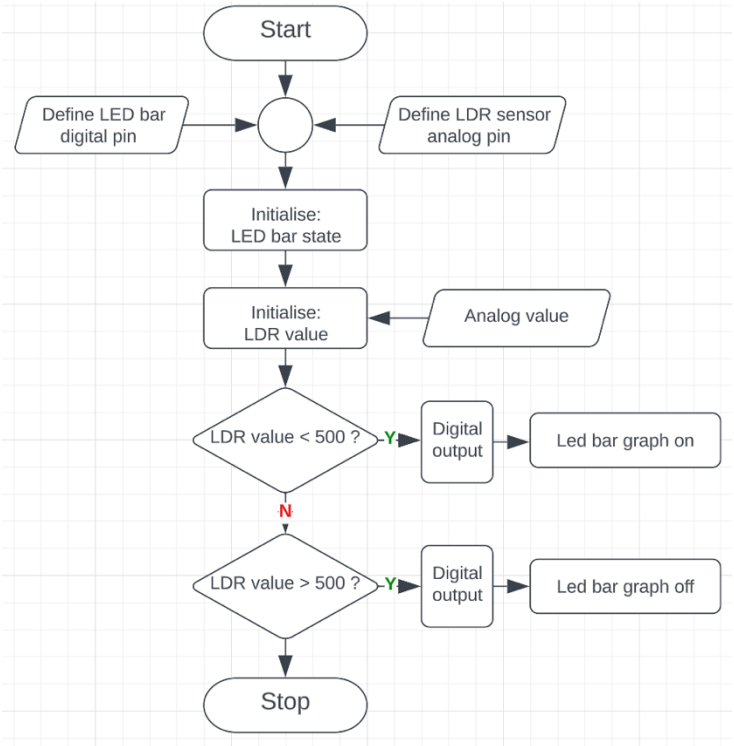
Pass: Create a night-activated LDR sensor to turn on the LED bar during night time and turn off LED bar during day time. The first menu should toggle the first buzzer on and off.

Arduino Sketch:

```
const int LDRPin = A0;     // LDR pin
const int LEDPin = 23;      // LED bar graph pin

void setup() {
  pinMode(LEDPin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  int LDRValue = analogRead(LDRPin);
  Serial.print("LDR value: ");
  Serial.println(LDRValue);
  // If it's dark, LED on
  if (LDRValue < 500) {
    digitalWrite(LEDPin, HIGH);
  }
```

```
  // If it's light, LED off
  else {
    digitalWrite(LEDPin, LOW);
  }
  delay(1000);  }
```

Flowchart:



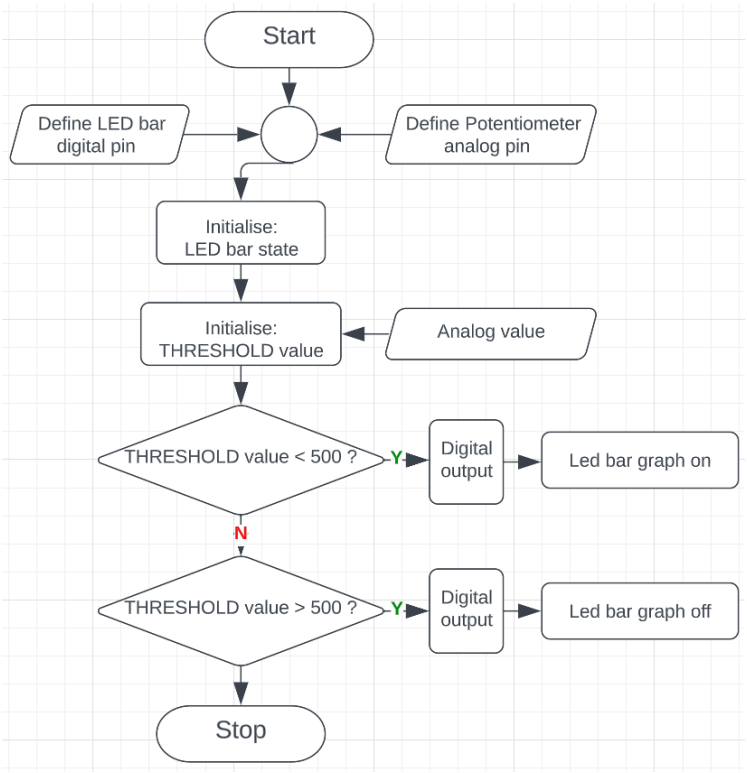
2. Pass Plus task

Pass Plus: Instead of LDR, use potentiometer to adjust the brightness of the LED bar.

Arduino Sketch:

```
const int LED_PIN = 23; // LED bar graph pin
const int LDR_PIN = A0; // Potentiometer pin
const int THRESHOLD = 500; // threshold value for light intensity

void setup() {
  pinMode(LED_PIN, OUTPUT);
  pinMode(LDR_PIN, INPUT);
}

void loop() {
  int lightIntensity = analogRead(LDR_PIN);
// if light intensity is below the threshold, turn LED
  if (lightIntensity < THRESHOLD) {
    digitalWrite(LED_PIN, HIGH);
// if light intensity is above the threshold, turn off LED
  } else {
    digitalWrite(LED_PIN, LOW);
  }}
```

Flowchart:



3. Credit task

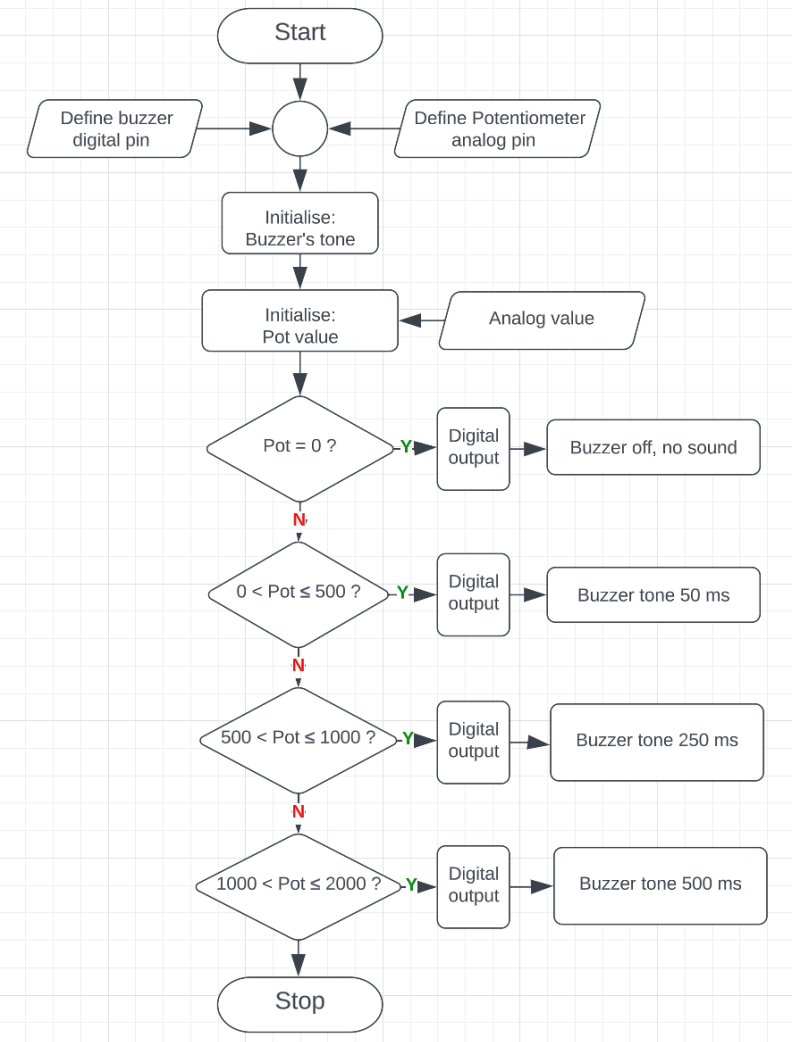
Credit: Write a program that controls the volume of noise from the speaker using input from the potentiometer

Arduino Sketch:

```
int potPin = A0;  // potentiometer pin
int buzzerPin = 13;  // buzzer pin
int volume = 0;   // initial volume level

void setup() {
  pinMode(buzzerPin, OUTPUT);
}

void loop() {
  volume = analogRead(potPin);
  volume = map(volume, 0, 1023, 0, 3);

  switch (volume) {
    // turn off the buzzer
    case 0:
      noTone(buzzerPin);
      break;
    // play a low tone for 50 ms
    case 1:
      tone(buzzerPin, 500, 50);
      break;
```

```
    // play a medium tone for 250 ms
    case 2:
      tone(buzzerPin, 1000, 250);
      break;
    // play a high tone for 500 ms
    case 3:
      tone(buzzerPin, 2000, 500);
      break;
  }

  delay(10);  }
```

Flowchart:



4. Distinction task

Distinction: Using a microphone to create a simple access control. The passcode/password is "123". This passcode need to be activated via voice over the mic. If the passcode is entered correctly, shows graphical symbol or image for correct authentication, otherwise shows incorrect image in the display.

Arduino Sketch:

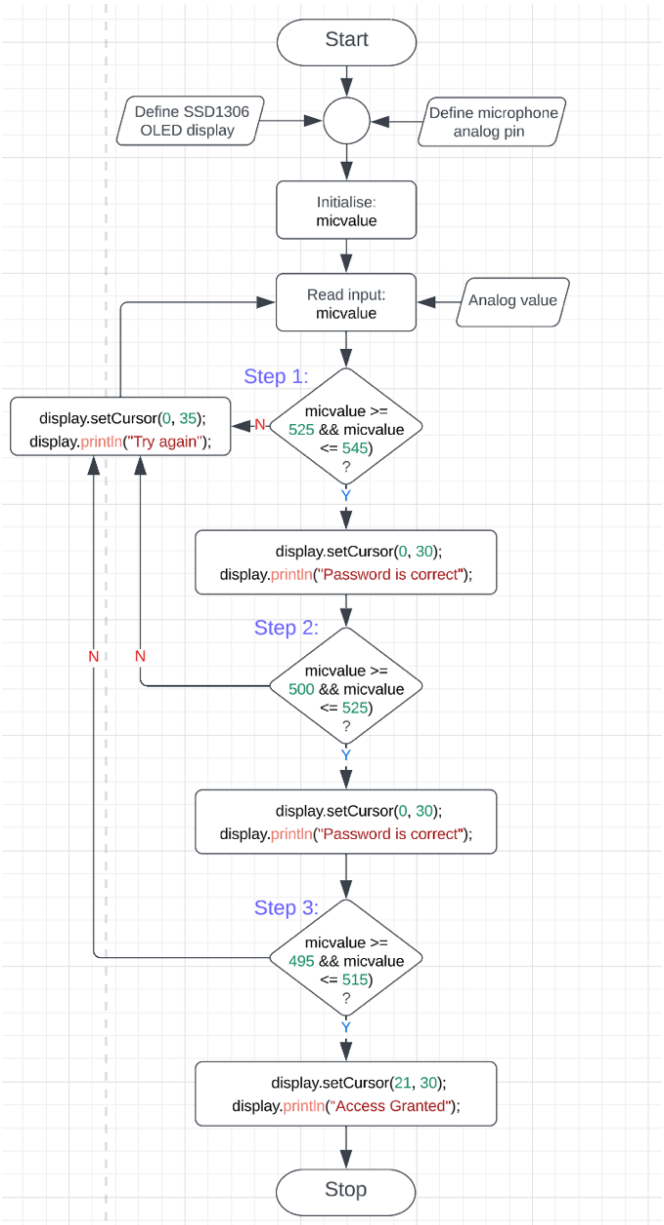
```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

void setup() {
  Serial.begin(115200);
  pinMode(13, OUTPUT);
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  display.clearDisplay();
  display.display();
}

void loop() {
  // Step 1: Display "Please say '1 2 3' for Identity Verification"
  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.setCursor(0, 15);
  display.println("Voices Confirmation");
  display.setCursor(0, 25);
  display.println("for Password and");
  display.setCursor(0, 35);
  display.println("Identity Verification");
  display.display();
  delay(3000);
  display.clearDisplay();

  // Step 2: Display "Step 1 Voice confirmation"
  display.setCursor(0, 25);
  display.println("Step 1");
  display.setCursor(0, 35);
  display.println("Voice confirmation:");
  display.display();
  delay(4000);
  display.clearDisplay();

  // Step 3: Read mic value
  int micvalue = analogRead(A0);
  Serial.println(micvalue);
  delay(200);
  Serial.println(micvalue);
  delay(200);
  Serial.println(micvalue);
  delay(200);
  Serial.println(micvalue);
  delay(200);
  Serial.println(micvalue);
```

```
delay(200);
Serial.println(micvalue);

// Step 4: Check mic value
if (micvalue >= 525 && micvalue <= 545){
  // Password is correct
  display.clearDisplay();
  display.setCursor(0, 30);
  display.println("Password is correct");
  display.display();
  delay(2000);

  // Step 5: Display "Step 2 Voice confirmation"
display.clearDisplay();
display.setCursor(0, 25);
display.println("Step 2");
display.setCursor(0, 35);
display.println("Voice confirmation:");
display.display();
delay(4000);
display.clearDisplay();

  // Step 6: Read mic value again
micvalue = analogRead(A0);
Serial.println(micvalue);
delay(200);
Serial.println(micvalue);
delay(200);
Serial.println(micvalue);
delay(200);
Serial.println(micvalue);
delay(200);
Serial.println(micvalue);
delay(200);
Serial.println(micvalue);

  // Step 7: Check mic value again
  if (micvalue >= 500 && micvalue <= 525) {
    // Password is correct
    display.clearDisplay();
    display.setCursor(0, 30);
    display.println("Password is correct");
    display.display();
    delay(2000);

    // Step 8: Display "Step 3 Voice confirmation"
    display.clearDisplay();
display.setCursor(0, 25);
display.println("Step 3");
display.setCursor(0, 35);
display.println("Voice confirmation:");
```

```
display.display();
delay(4000);
display.clearDisplay();

    // Step 9: Read mic value again
micvalue = analogRead(A0);
Serial.println(micvalue);
delay(200);
Serial.println(micvalue);
delay(200);
Serial.println(micvalue);
delay(200);
Serial.println(micvalue);
delay(200);
Serial.println(micvalue);
delay(200);
Serial.println(micvalue);

    // Step 10: Check mic value again
    if (micvalue >= 495 && micvalue <= 515) {
      // Access granted
      display.clearDisplay();
      display.setCursor(21, 30);
      display.println("Access Granted");
      display.display();
      delay(10000);
      display.clearDisplay();

    } else {
      // Mic value is incorrect
      display.clearDisplay();
      display.setCursor(0, 35);
      display.println("Try again");
      display.display();
      delay(3000);
      display.clearDisplay();
    }
  } else {
    // Mic value is incorrect
    display.clearDisplay();
    display.setCursor(0, 35);
    display.println("Try again");
    display.display();
    delay(3000);
    display.clearDisplay();
  }
} else {
  // Mic value is incorrect
  display.clearDisplay();
  display.setCursor(0, 35);
  display.println("Try again");
```

```
    display.display();
      delay(3000);
      display.clearDisplay();
  }
}
```

Flowchart:



D. Lab 5 Report Pseudo and Flowchart

1. Pass and Pass Plus tasks

Pass: Store the following list in PROGEM, then print them from PROGEM onto the LCD screen (SSD1306 OLED Display or TFT-LCD display). Each item from the list should scroll from right to left.

- *Student ID*
- *Student name*

Pass Plus: Continue from the Pass question above with following list:

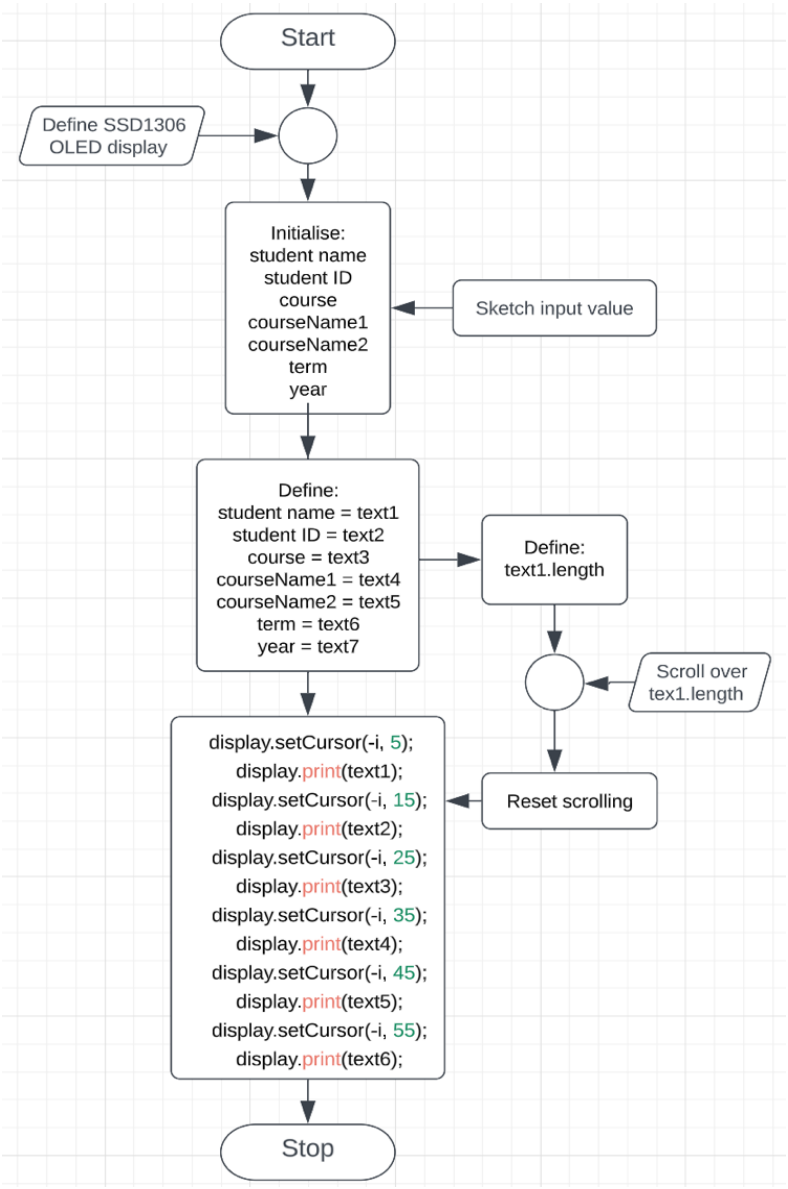
- ENG20009
- Engineering Technology Inquiry Project
- Semester 1
- 2023

Arduino Sketch:

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

void setup() {
  Serial.begin(9600);

  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println(F("SSD1306 allocation failed"));
    while (true);
  }

  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(WHITE);
}

void loop() {
  String studentID = "ID: 103844421";
  String studentName = "Dang Khoa Le";
  String course = "ENG20009";
  String course1Name = "Engineering Technology";
  String course2Name = "Inquiry Project";
  String term = "Semester 1";
  String year = "2023";

  String text1 = studentID;
  String text2 = studentName;
  String text3 = course;
  String text4 = course1Name;
  String text5 = course2Name;
  String text6 = term + ",  " + year + "  ";

  for (int i = 0; i <= text1.length() * 6; i++) {
    display.clearDisplay();
    display.setCursor(-i, 5);
    display.print(text1);
    display.setCursor(-i, 15);
```

```
    display.print(text2);
    display.setCursor(-i, 25);
    display.print(text3);
    display.setCursor(-i, 35);
    display.print(text4);
    display.setCursor(-i, 45);
    display.print(text5);
    display.setCursor(-i, 55);
    display.print(text6);
    display.display();
    delay(50);
  }}
```

Flowchart:



2. Credit task

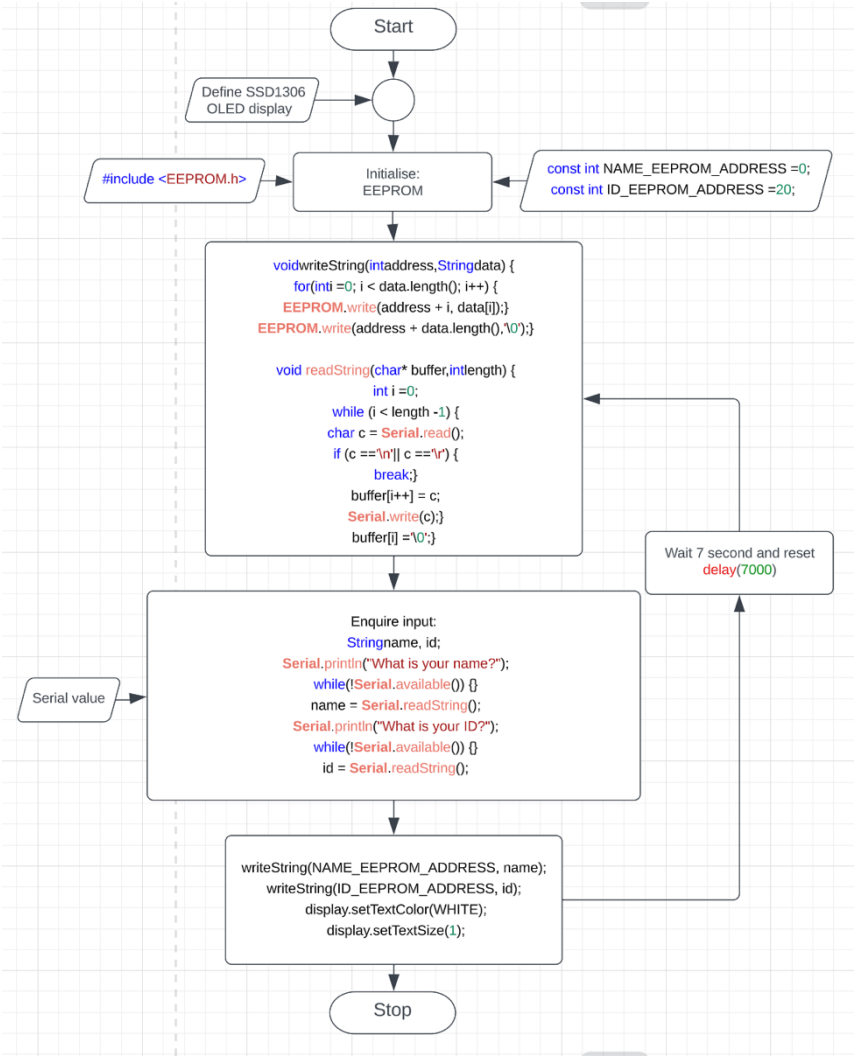
Credit: Connect to the EEPROM component. Write to the component your student ID and display it back on the LCD (SSD1306 OLED Display or TFT-LCD display) from the EEPROM.

Arduino Sketch:

```
#include <Wire.h>
#include <Adafruit_SSD1306.h>
#include <Adafruit_GFX.h>
#include <EEPROM.h>

#define OLED_RESET 4
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_ADDRESS 0x3C

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

const int NAME_EEPROM_ADDRESS = 0;
const int ID_EEPROM_ADDRESS = 20;


void setup() {
  Serial.begin(9600);
  Wire.begin();

  // Initialize OLED display
  if (!display.begin(SSD1306_SWITCHCAPVCC, OLED_ADDRESS)) {
    Serial.println("SSD1306 initialization failed");
    while (1);
  }

  // Clear the display
  display.clearDisplay();

  // Write default values to EEPROM
}

void loop() {
abc();
}

// Function to write a string to EEPROM
void writeString(int address, String data) {
  for (int i = 0; i < data.length(); i++) {
    EEPROM.write(address + i, data[i]);
  }
  EEPROM.write(address + data.length(), '\0');
}

// Function to read a string from EEPROM
void readString(char* buffer, int length) {
  int i = 0;
  while (i < length - 1) {
    char c = Serial.read();
```

```
    if (c == '\n' || c == '\r') {
      break;
    }
    buffer[i++] = c;
    Serial.write(c);
  }
  buffer[i] = '\0';
}

 void abc(){
   String name, id;
  // Prompt user to enter name and ID on serial monitor
  Serial.println("What is your name?");
  while (!Serial.available()) {}
  name = Serial.readString();

  Serial.println("What is your ID?");
  while (!Serial.available()) {}
  id = Serial.readString();

  // Write name and ID to EEPROM
  writeString(NAME_EEPROM_ADDRESS, name);
  writeString(ID_EEPROM_ADDRESS, id);
  display.setTextColor(WHITE);
  display.setTextSize(1);

  // Display name and ID on OLED
  display.clearDisplay();
  display.setCursor(0, 10);
  display.println("Your Name is:");
  display.setCursor(0, 20);
  display.println(name);
  Serial.println("Your Name is:");
  Serial.println(name);

  display.setCursor(0, 40);
  display.println("Your ID is :");
  display.setCursor(0, 50);
  display.println(id);
  Serial.println("Your ID is :");
  Serial.println(id);
  display.display();
  delay(7000);
  display.clearDisplay();
 }
```

Flowchart:



E. Lab 6 Report Pseudo and Flowchart

1. Pass and Pass Plus tasks

Pass: Using the interrupt hardware for pushbutton, display a non-alphanumeric symbol on the LCD LCD (SSD1306 OLED Display or TFT-LCD display).

Pass Plus: Continue Pass question above, using the interrupt hardware for pushbutton, display various symbols on the LCD, each time the button is pressed it should trigger an interrupt to change the symbol

Arduino Sketch:

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>


#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
```

```
const unsigned char symbol1Bitmap [1024] PROGMEM = {
    // 'logoBN128x64, 128x64px
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xc0, 0x03, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0xe0, 0x07, 0x80, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0xe0, 0x07, 0x80, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0xf0, 0x07, 0x80, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xf3, 0xcf, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0xff, 0xff, 0x80, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0xe0, 0xff, 0xc0, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0xc0, 0x7f, 0xe0, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0xe7, 0xc0, 0x7f, 0xe7, 0xc0, 0x00, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x3f, 0xff, 0xe0, 0xff, 0xff, 0xfc, 0x00, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x01, 0xf8, 0x0f, 0xff, 0xff, 0xf0, 0x1f, 0x80, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x03, 0xe0, 0x03, 0xff, 0xff, 0xc0, 0x03, 0xc0, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x07, 0x80, 0x00, 0xff, 0xff, 0x00, 0x41, 0xe0, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x07, 0x00, 0x00, 0x7f, 0xfe, 0x00, 0xe0, 0xf0, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x0e, 0x00, 0x00, 0x1f, 0xf8, 0x00, 0xe0, 0x70, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x0e, 0x00, 0x00, 0x0f, 0xf0, 0x00, 0xe0, 0x70, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x1c, 0x00, 0x00, 0x07, 0xe0, 0x00, 0xe0, 0x38, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x1c, 0x7f, 0xf0, 0x03, 0xe0, 0x00, 0xe0, 0x38, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x1c, 0x7f, 0xf0, 0x03, 0xc0, 0x1f, 0xff, 0x38, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x1c, 0x7f, 0xf0, 0x03, 0xe0, 0x1f, 0xff, 0x38, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x1c, 0x00, 0x00, 0x07, 0xe0, 0x00, 0x60, 0x38, 0x00,
0x00, 0x00, 0x00,
```

```
    0x00, 0x00, 0x00, 0x00, 0x0e, 0x00, 0x00, 0x0f, 0xf0, 0x00, 0xe0, 0x70, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x0e, 0x00, 0x00, 0x1f, 0xf8, 0x00, 0xe0, 0x70, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x07, 0x00, 0x00, 0x7c, 0x3e, 0x00, 0xe0, 0xf0, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x07, 0x80, 0x00, 0xf8, 0x1f, 0x00, 0x61, 0xe0, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x03, 0xc0, 0x03, 0xe0, 0x07, 0xc0, 0x03, 0xc0, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x01, 0xf8, 0x0f, 0xc0, 0x03, 0xf0, 0x1f, 0x80, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x3f, 0xff, 0xff, 0xff, 0xff, 0xfc, 0x00, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0xff, 0xff, 0xff, 0xff, 0xe0, 0x00, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3f, 0xff, 0xff, 0xfe, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3f, 0xff, 0xff, 0xfc, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x38, 0x00, 0x00, 0x1c, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3c, 0x00, 0x00, 0x1c, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x1c, 0x00, 0x00, 0x3c, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x1f, 0xff, 0xff, 0xf8, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x1f, 0xff, 0xff, 0xf8, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f, 0xff, 0xff, 0xf0, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f, 0xff, 0xff, 0xf0, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0xff, 0xff, 0xe0, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0xe0, 0x07, 0xc0, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0xf0, 0x0f, 0x80, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xfc, 0x3f, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x7f, 0xfe, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x1f, 0xf8, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f, 0xf0, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
```

```
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x80, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x10, 0x00, 0x08, 0x10, 0x00, 0x00, 0x80,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x08, 0x00, 0x00, 0x08, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x08, 0x00, 0x00, 0x08, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x46, 0x63, 0x31, 0x90, 0xff, 0x13, 0xc3, 0x8b, 0x91, 0xe5, 0xdc, 0x80,
0xf3, 0xbf, 0x00,
    0x00, 0x66, 0x73, 0x39, 0x90, 0xcd, 0x92, 0x24, 0x4c, 0x92, 0x26, 0x48, 0x81,
0x13, 0x27, 0x00,
    0x00, 0x2e, 0x57, 0x2b, 0x90, 0x88, 0x93, 0x04, 0x08, 0xd0, 0x24, 0x68, 0x81,
0x1a, 0x23, 0x00,
    0x00, 0x2a, 0xd5, 0x6a, 0xb0, 0x88, 0x91, 0xcc, 0x08, 0xd3, 0xe4, 0x68, 0x83,
0x1a, 0x63, 0x00,
    0x00, 0x2b, 0x95, 0xca, 0xe0, 0x88, 0x90, 0x64, 0x08, 0xd2, 0x24, 0x68, 0x81,
0x1a, 0x63, 0x00,
    0x00, 0x39, 0x9c, 0xce, 0x60, 0x88, 0x96, 0x24, 0x48, 0xd6, 0x64, 0x68, 0x81,
0x12, 0x27, 0x00,
    0x00, 0x11, 0x88, 0xc4, 0x6c, 0x88, 0x93, 0xc3, 0x88, 0xd3, 0xb4, 0x6e, 0x98,
0xf2, 0x3f, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x02, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x22, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x3c, 0x00
};

const unsigned char symbol2Bitmap [1024] PROGMEM = {
// 'monkey01
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff,
0xff, 0xff, 0xfe, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfc, 0x7f, 0x3f, 0xff,
0xff, 0xff,
```

```
0xff, 0xff, 0xfc, 0x7e, 0x3f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf8, 0x78, 0x7f, 0xff,
0xff, 0xff,
0xff, 0xff, 0xf8, 0x70, 0x7f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf8, 0x20, 0xff, 0xff,
0xff, 0xff,
0xff, 0xff, 0xf0, 0x00, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf0, 0x70, 0x00, 0x7f, 0xff,
0xff, 0xff,
0xff, 0xf0, 0x00, 0x00, 0x7f, 0xff, 0xff, 0xff, 0xff, 0xf8, 0x00, 0x00, 0x3c, 0x7f,
0xff, 0xff,
0xff, 0xfc, 0x00, 0x00, 0x00, 0x1f, 0xff, 0xff, 0xff, 0xfe, 0x00, 0x20, 0x07, 0x8f,
0xff, 0xff,
0xff, 0xff, 0x01, 0xf8, 0x0f, 0xc7, 0xff, 0xff, 0xff, 0xff, 0x83, 0xfe, 0x1f, 0xe7,
0xff, 0xff,
0xff, 0xff, 0x87, 0xff, 0x1f, 0xe7, 0xff, 0xff, 0xff, 0xff, 0x87, 0xff, 0x3f, 0xe6,
0x07, 0xff,
0xff, 0xcf, 0x87, 0xff, 0xf9, 0xf0, 0xe3, 0xff, 0xff, 0x03, 0x07, 0xff, 0xf1, 0xf1,
0xf8, 0xff,
0xfc, 0x38, 0x07, 0xe1, 0xe0, 0xf3, 0xfc, 0xff, 0xf8, 0x78, 0x07, 0xe0, 0xe0, 0xf7,
0xf8, 0x7f,
0xf8, 0xfe, 0x07, 0xe0, 0xe0, 0x7f, 0x10, 0x7f, 0xf1, 0xde, 0x07, 0xe0, 0x60, 0x7e,
0x06, 0x7f,
0xf3, 0x8f, 0x03, 0xf0, 0x70, 0xfd, 0xee, 0x7f, 0xf3, 0x3f, 0x03, 0xf0, 0x70, 0x7d,
0xee, 0x7f,
0xe6, 0x3e, 0x0f, 0xe0, 0x70, 0x7d, 0xfe, 0x7f, 0xe6, 0x1f, 0x0f, 0xf0, 0x70, 0xfd,
0xfe, 0x7f,
0xe6, 0xcf, 0x1f, 0xf0, 0x78, 0xff, 0xfc, 0x7f, 0xe6, 0xff, 0x3f, 0xf8, 0xfd, 0xff,
0xfc, 0xff,
0xe7, 0xff, 0x3f, 0xf9, 0xff, 0xff, 0xf8, 0xff, 0xe7, 0xff, 0x7f, 0xff, 0xff, 0xff,
0xf3, 0xff,
0xf7, 0xff, 0x7c, 0xff, 0xff, 0xff, 0xc7, 0xff, 0xf3, 0xff, 0x3e, 0x3f, 0xff, 0xff,
0x0f, 0xff,
0xf0, 0xfe, 0x3f, 0x07, 0xff, 0xfc, 0x3f, 0xff, 0xf8, 0x00, 0x3f, 0xc0, 0x38, 0x00,
0x7f, 0xff,
0xfc, 0x03, 0x1f, 0xfc, 0x00, 0x01, 0xff, 0xff, 0xff, 0xff, 0x9f, 0xff, 0x83, 0xe3,
0xff, 0xff,
0xff, 0xff, 0x8f, 0xff, 0xff, 0xf3, 0xff, 0xff, 0xff, 0xff, 0xc7, 0xff, 0xff, 0xf3,
0xff, 0xfb,
0xff, 0xff, 0xe3, 0xff, 0xff, 0xe7, 0xff, 0xc0, 0xff, 0xff, 0xe0, 0xff, 0xff, 0x87,
0xff, 0x86,
0xff, 0xff, 0xfc, 0x1f, 0xfe, 0x0f, 0xff, 0x9f, 0xff, 0xff, 0xff, 0x00, 0x30, 0x1f,
0xff, 0x3f,
0xff, 0xff, 0xf8, 0x00, 0x00, 0x0f, 0xff, 0x7f, 0xff, 0xfe, 0x00, 0x00, 0x00, 0x03,
0xff, 0x79,
0xff, 0xf8, 0x00, 0x00, 0x00, 0x03, 0xfe, 0x79, 0xff, 0x80, 0x00, 0x00, 0x00, 0x01,
0xfe, 0xf9,
0xff, 0x00, 0x00, 0x00, 0x00, 0x00, 0xfe, 0xe0, 0xff, 0x00, 0x00, 0x00, 0x00, 0x00,
0xfe, 0x78,
0xfe, 0x00, 0x00, 0x00, 0x00, 0x00, 0x7e, 0x7f, 0xfe, 0x00, 0x00, 0x00, 0x00, 0x00,
0x7c, 0x7f,
0xfc, 0x00, 0x00, 0x00, 0x00, 0x01, 0x3e, 0x7f, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x01,
0x3e, 0x1f,
```

```
0xf8, 0x01, 0xff, 0x80, 0x00, 0x01, 0x8e, 0x00, 0xf8, 0x01, 0xff, 0x00, 0x00, 0x00,
0x0e, 0x00,
0xf8, 0x03, 0xfe, 0x00, 0x00, 0x01, 0x86, 0x01, 0xf8, 0x03, 0xfe, 0x00, 0x00, 0x00,
0x80, 0x00,
0xfc, 0x03, 0xfe, 0x00, 0x00, 0x00, 0xc0, 0x00, 0xf8, 0x01, 0xfe, 0x00, 0x00, 0x00,
0xc0, 0x00,
0xf2, 0x60, 0x7e, 0x00, 0x00, 0x00, 0x40, 0x03, 0xcf, 0xfe, 0x3e, 0x00, 0x00, 0x00,
0x60, 0x07,
0xdf, 0xff, 0x1e, 0x00, 0x00, 0x00, 0x60, 0x07, 0x9f, 0xff, 0x1e, 0x00, 0x00, 0x00,
0x60, 0x07,
0xbf, 0xff, 0x8e, 0x00, 0x00, 0x00, 0x70, 0x0f, 0x3f, 0xff, 0x0e, 0x00, 0x00, 0x00,
0x7c, 0x7f,
0x3f, 0xff, 0x0e, 0x00, 0x00, 0x00, 0x3f, 0xff, 0xbf, 0xff, 0x27, 0x00, 0x00, 0x00,
0x7f, 0xff,
0x3f, 0xff, 0x27, 0x00, 0x00, 0x00, 0x7f, 0xff, 0xbf, 0xff, 0x27, 0x80, 0x00, 0x80,
0x7f, 0xff,
0xbf, 0xff, 0x87, 0x80, 0x00, 0xe0, 0x7f, 0xff, 0x9f, 0xff, 0x83, 0xe0, 0x00, 0x70,
0x7c, 0x3f,
0x9b, 0xff, 0xc3, 0xf0, 0x00, 0x18, 0xfc, 0xbf, 0x9b, 0xff, 0xe7, 0xf8, 0x00, 0x0e,
0xfd, 0xa3,
0x89, 0xbf, 0xe7, 0xfc, 0x00, 0x03, 0xfd, 0x81, 0xc0, 0x4f, 0xc7, 0xfe, 0x00, 0x03,
0xfb, 0x9d,
0xfe, 0x0c, 0x0f, 0xff, 0x00, 0x00, 0xfb, 0x98, 0xfe, 0x00, 0x1f, 0xff, 0x80, 0x01,
0xf3, 0xb8,
0xff, 0x70, 0x1f, 0xff, 0xc0, 0x00, 0xf3, 0xb9, 0xff, 0xff, 0xff, 0xff, 0xe0, 0x00,
0x77, 0x3b,
0xff, 0xff, 0xf0, 0xff, 0x30, 0x00, 0x77, 0x73, 0xff, 0xff, 0xe0, 0x1e, 0x18, 0x00,
0x37, 0x73,
0xff, 0xff, 0x88, 0x00, 0x00, 0x00, 0x36, 0x77, 0xff, 0xff, 0x38, 0x00, 0x02, 0x00,
0x06, 0xe7,
0xff, 0xff, 0x30, 0x00, 0x03, 0x80, 0x07, 0xff, 0xff, 0xfe, 0x70, 0x00, 0x01, 0xc0,
0x07, 0xfe,
0xff, 0xfe, 0x78, 0x00, 0x01, 0xc0, 0x0f, 0xfc, 0xff, 0xfe, 0x7c, 0x00, 0x03, 0xe0,
0x0f, 0xfc,
0xff, 0xfc, 0x7e, 0x00, 0x07, 0xf0, 0x0f, 0xf8, 0xff, 0xfc, 0xff, 0x00, 0x0f, 0xf8,
0x0f, 0xf8,
0xff, 0xfc, 0xff, 0xc0, 0x07, 0xf8, 0x1f, 0xf8, 0xff, 0xfc, 0xff, 0xf3, 0x87, 0xfc,
0x3f, 0xf8,
0xff, 0xfc, 0xff, 0xff, 0xc3, 0xfe, 0x3f, 0xf9, 0xff, 0xfc, 0x7f, 0xff, 0xf1, 0xff,
0x3f, 0xf1,
0xff, 0xfe, 0x7f, 0xfe, 0x79, 0xff, 0x7f, 0xf3, 0xff, 0xfe, 0x7f, 0xfe, 0x31, 0xff,
0x7f, 0xf3,
0xff, 0xfe, 0x3f, 0xff, 0x03, 0xff, 0x3f, 0xe7, 0xff, 0xff, 0x1f, 0xe7, 0x03, 0xff,
0xbf, 0xe7,
0xff, 0xff, 0x8f, 0xe3, 0x3f, 0xff, 0x9f, 0xcf, 0xff, 0xff, 0x87, 0xf3, 0x3f, 0xff,
0x8f, 0x8f,
0xff, 0xff, 0xc0, 0xf0, 0x7f, 0xff, 0xc0, 0x3f, 0xff, 0xff, 0xf0, 0xf0, 0xff, 0xff,
0xf0, 0x7f,
0xff, 0xff, 0xf8, 0x03, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfe, 0x03, 0xff, 0xff,
0xff, 0xff,
```

```
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff
};

const unsigned char symbol3Bitmap [1024] PROGMEM = {
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0xff, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0xf0, 0x00, 0x07, 0xc1, 0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x01, 0x8f, 0xf0, 0x1c, 0x00, 0x1c, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x01, 0x80, 0x7f, 0x70, 0x00, 0x06, 0x00, 0x03, 0x80, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0xc2, 0x07, 0xe0, 0x00, 0x03, 0xff, 0xff, 0xc0, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
```

```
  0x00, 0x60, 0x60, 0x00, 0x80, 0x00, 0x00, 0x00, 0x40, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x30, 0x1c, 0x00, 0x49, 0x00, 0x00, 0x70, 0xc0, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x18, 0x06, 0x00, 0x08, 0x00, 0x0f, 0x81, 0x80, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x08, 0x03, 0x01, 0x00, 0x7c, 0x30, 0x03, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x0c, 0x01, 0x3f, 0x00, 0x7e, 0x60, 0x06, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x06, 0x01, 0x7f, 0x80, 0xfe, 0x40, 0x0c, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x02, 0x01, 0x3f, 0x89, 0xfe, 0x40, 0x0c, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x03, 0x00, 0x3f, 0x91, 0xfe, 0x00, 0x18, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x01, 0xc0, 0x1f, 0x81, 0xf8, 0x00, 0x70, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x70, 0x00, 0x00, 0x00, 0x81, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x3c, 0x00, 0x00, 0x01, 0x9f, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x03, 0xc0, 0x00, 0x3f, 0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x01, 0x3f, 0xff, 0xf0, 0x78, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x01, 0x1e, 0xe0, 0x00, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x01, 0x80, 0x00, 0x01, 0x98, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0xc0, 0x00, 0x07, 0x8c, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0xe0, 0x00, 0x1d, 0x0c, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x01, 0x98, 0x00, 0xcd, 0x3c, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x03, 0x8f, 0x81, 0x99, 0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x03, 0x4b, 0x9f, 0x19, 0x30, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x03, 0x72, 0x68, 0x2c, 0x60, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x03, 0x92, 0x48, 0x4c, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x01, 0xd0, 0x48, 0xdf, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x03, 0x7c, 0x09, 0x9e, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x01, 0xf8, 0x0b, 0x1c, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
```

```
  0x00, 0x00, 0x00, 0x60, 0x88, 0x0c, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x20, 0x88, 0x0c, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x20, 0x88, 0x0e, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x70, 0x88, 0x0e, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x60, 0x88, 0x0e, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x60, 0x88, 0x0e, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x78, 0xc8, 0x1e, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x3f, 0xff, 0xfc, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00
};

const unsigned char MarilynMonroe [] PROGMEM = {
  0xff, 0xff, 0xff, 0xff, 0xff, 0xf8, 0x1f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xc0, 0x1f, 0xff, 0xff, 0xf0, 0x41, 0xff, 0xff,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0x80, 0x7f, 0xff, 0xff, 0xf8, 0x03, 0xff, 0xff,
0xff, 0xff, 0xff,
```

```
  0xff, 0xff, 0xff, 0xff, 0xff, 0xf9, 0xff, 0xff, 0xff, 0xe0, 0x07, 0xff, 0xff,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0x87, 0xff, 0xff, 0xff, 0xf8, 0x03, 0xff, 0xff,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0x07, 0xff, 0xff, 0xff, 0xf8, 0x01, 0xf1, 0xff,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0x9f, 0xff, 0xff, 0xff, 0xf8, 0x00, 0xf8, 0xff,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xbf, 0xff, 0xff, 0xff, 0xfc, 0x02, 0x78, 0x7f,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xfc, 0x3f, 0xff, 0xff, 0xfe, 0x03, 0x7c, 0x1f,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xf0, 0x07, 0xff, 0xff, 0xfe, 0x01, 0xfe, 0x1f,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xfd, 0xe0, 0x03, 0xff, 0xff, 0xfc, 0x00, 0xfe, 0x0f,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xfe, 0x87, 0xe0, 0xff, 0xff, 0xfc, 0x00, 0x06, 0x07,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xfc, 0x1f, 0xf9, 0xff, 0xff, 0xfc, 0x00, 0x02, 0x07,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xf8, 0x1f, 0xff, 0xff, 0xff, 0xfc, 0x00, 0xc3, 0xc3,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xf0, 0x3f, 0xff, 0xff, 0xe0, 0x0c, 0x00, 0xe7, 0x81,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xf0, 0x0f, 0xff, 0xff, 0xe0, 0x02, 0x00, 0x02, 0x00,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xf0, 0x0f, 0xff, 0xff, 0xe0, 0x01, 0x00, 0x00, 0x00,
0x3f, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0x80, 0x00, 0x3f, 0xff, 0xff, 0xe0, 0x00, 0x00, 0x1e,
0x3f, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xfc, 0x00, 0x00, 0x0f, 0xff, 0x3f, 0xf8, 0x00, 0x18, 0x7f,
0x1f, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xf8, 0x01, 0x80, 0x03, 0xfc, 0x3f, 0xfc, 0x00, 0x70, 0xfe,
0x1f, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xf0, 0x43, 0xff, 0xff, 0xf8, 0x7f, 0xf8, 0x00, 0x00, 0x7e,
0x1f, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xe0, 0x07, 0xff, 0xff, 0xf0, 0xff, 0xfc, 0x00, 0x00, 0x7c,
0x3f, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xe0, 0x0f, 0xff, 0xff, 0xf1, 0xef, 0xf8, 0x00, 0x01, 0xfc,
0x3f, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xe4, 0xff, 0xff, 0xff, 0xf3, 0x80, 0xa0, 0x00, 0x07, 0xfc,
0xaf, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xec, 0x5f, 0xff, 0xff, 0xe7, 0xf0, 0x00, 0x00, 0x03, 0xfe,
0xdf, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xee, 0x7f, 0xff, 0xff, 0xc7, 0xf8, 0x00, 0x00, 0x03, 0xff,
0xdf, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xfe, 0x7f, 0xff, 0xf7, 0xc7, 0xff, 0x06, 0x00, 0x03, 0xff,
0xbf, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xfe, 0x5f, 0xff, 0xc7, 0x07, 0xff, 0x80, 0x00, 0x07, 0xdb,
0xbf, 0xff, 0xff,
```

```
  0xff, 0xff, 0xff, 0xee, 0xff, 0xff, 0x80, 0x03, 0xff, 0xc0, 0x00, 0x03, 0xc3,
0x0f, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xfe, 0xff, 0xff, 0x98, 0x03, 0xff, 0xf8, 0x00, 0x07, 0xe0,
0x0f, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xef, 0xff, 0xff, 0xf8, 0x01, 0xff, 0xfc, 0x01, 0x07, 0xfc,
0x1f, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xcf, 0xef, 0xff, 0xff, 0xe1, 0xff, 0xfc, 0x01, 0x07, 0xf8,
0x1f, 0xff, 0xff,
  0xff, 0xff, 0xff, 0x9f, 0xff, 0xff, 0x7f, 0xf1, 0xff, 0xf8, 0x02, 0x07, 0x88,
0x3f, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xcf, 0xef, 0xf8, 0x0f, 0xff, 0xff, 0xe0, 0x00, 0x07, 0x84,
0x3f, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xe7, 0xef, 0xf0, 0x04, 0x7f, 0xff, 0xc0, 0x00, 0x07, 0x84,
0x7f, 0xff, 0xff,
  0xff, 0xff, 0xff, 0x3f, 0xff, 0xe0, 0x00, 0x1f, 0xff, 0x80, 0x00, 0x06, 0x04,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0x3f, 0x7f, 0xe1, 0xf0, 0x07, 0xff, 0x80, 0x00, 0x07, 0x06,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xc3, 0xfe, 0x03, 0xff, 0x00, 0x00, 0x03, 0x80,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xf2, 0x3f, 0xc6, 0x7f, 0x81, 0xce, 0x00, 0x00, 0x01, 0xc1,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xe0, 0x3f, 0xc0, 0x07, 0xc1, 0xfe, 0x00, 0x00, 0x0d, 0xc0,
0x7f, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xe0, 0x3f, 0xc0, 0x01, 0xe0, 0xfc, 0x00, 0x00, 0x0f, 0xc0,
0x7f, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xc0, 0x3f, 0xc0, 0x00, 0x50, 0xfc, 0x00, 0x00, 0x0e, 0xc0,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xc0, 0x3f, 0xc0, 0x00, 0x18, 0xf8, 0x00, 0x00, 0x0e, 0xc1,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xc0, 0x3f, 0xc0, 0x00, 0x00, 0xf8, 0x00, 0x00, 0x66, 0x81,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xc0, 0x1f, 0xc7, 0x80, 0x00, 0xf8, 0x00, 0x01, 0xe0, 0x00,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xc0, 0x1f, 0xc1, 0xe0, 0x01, 0xf8, 0x00, 0x03, 0xf0, 0x01,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0x80, 0x1f, 0xc0, 0x3e, 0x03, 0xf0, 0x00, 0x00, 0xe0, 0x03,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0x00, 0x1f, 0xe0, 0xe0, 0x03, 0xf2, 0x00, 0x00, 0xc0, 0x03,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0x80, 0x1f, 0xf0, 0x00, 0x07, 0xe6, 0x00, 0x00, 0xc0, 0x03,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0x80, 0x1f, 0xff, 0x00, 0x1f, 0xee, 0x00, 0x00, 0x80, 0x07,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xb8, 0x0f, 0xff, 0xf0, 0x3f, 0xdc, 0x00, 0x00, 0x00, 0x0f,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xbc, 0x0f, 0xff, 0xff, 0xff, 0xdc, 0x00, 0x00, 0x00, 0x0f,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0x9e, 0x0f, 0xff, 0xff, 0xff, 0xf8, 0x00, 0x00, 0x00, 0x1f,
0xff, 0xff, 0xff,
```

```
  0xff, 0xff, 0xff, 0x08, 0x0f, 0xff, 0xff, 0xff, 0x70, 0x00, 0x00, 0x00, 0x1f,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0x00, 0x0b, 0xff, 0xff, 0xfe, 0xe0, 0x00, 0x00, 0x00, 0x1f,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0x00, 0x0b, 0xff, 0xff, 0xf9, 0xc0, 0x00, 0x00, 0x00, 0x3f,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0x3c, 0x09, 0xff, 0xff, 0xf1, 0x80, 0x00, 0x00, 0x00, 0x7f,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0x1e, 0x08, 0x3f, 0xff, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x7f,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0x1f, 0x08, 0x03, 0xff, 0x00, 0x00, 0x00, 0x00, 0x00, 0x7f,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0x00, 0x08, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x1f,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0x80, 0x1c, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x1f,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xce, 0x1c, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x1f,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xfe, 0x1c, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3f,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0x7e, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x7f,
0xff, 0xff, 0xff
};

const unsigned char Doraemon [] PROGMEM  = {
  0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x80, 0x00, 0x3f, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xe0, 0x00, 0x00, 0x00, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00, 0x00, 0x00, 0x1f, 0xff, 0xff,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xf8, 0x00, 0x00, 0x78, 0x00, 0x03, 0xff, 0xff,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xc0, 0x00, 0x03, 0xfe, 0x00, 0x00, 0x7f, 0xff,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00, 0x07, 0xff, 0x80, 0x00, 0x1f, 0xff,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xfc, 0x07, 0xf0, 0x0f, 0xff, 0xc0, 0x00, 0x07, 0xff,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xf0, 0x1f, 0xf8, 0x1f, 0xff, 0xe0, 0x00, 0x01, 0xff,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xe0, 0x3f, 0xfc, 0x3f, 0xff, 0xe0, 0x00, 0x00, 0xff,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0x80, 0x7f, 0xfe, 0x3f, 0xff, 0xf0, 0x00, 0x00, 0x3f,
0xff, 0xff, 0xff,
```

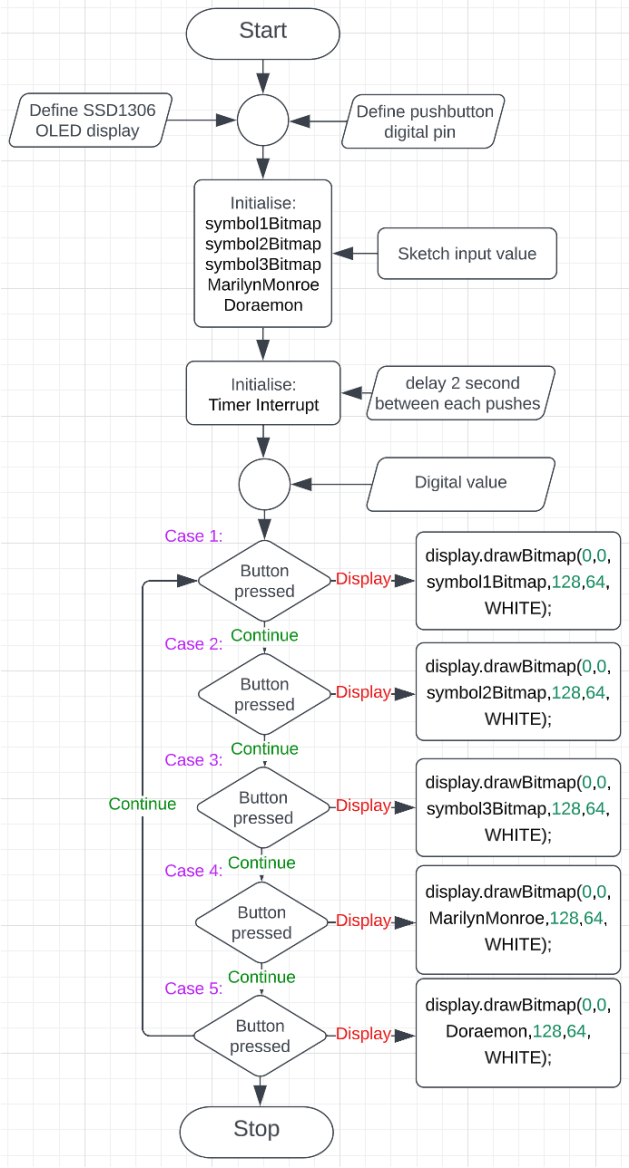
```
  0xff, 0xff, 0xff, 0xff, 0x00, 0xff, 0xfe, 0x3f, 0xff, 0xf0, 0x00, 0x00, 0x1f,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xfc, 0x00, 0xff, 0xff, 0x7f, 0xff, 0xf0, 0x00, 0x00, 0x07,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xf8, 0x01, 0xff, 0xff, 0xff, 0xff, 0xf0, 0x00, 0x00, 0x03,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xf0, 0x01, 0xff, 0xff, 0xff, 0xff, 0xf4, 0x00, 0x00, 0x01,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xe0, 0x03, 0xff, 0xff, 0xff, 0xff, 0xf7, 0x00, 0x00, 0x00,
0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xc0, 0x03, 0xff, 0xfe, 0xff, 0x87, 0xf7, 0xe0, 0x00, 0x00,
0x7f, 0xff, 0xff,
  0xff, 0xff, 0xff, 0x80, 0x03, 0xff, 0xff, 0xbf, 0x97, 0xff, 0xf8, 0x00, 0x00,
0x3f, 0xff, 0xff,
  0xff, 0xff, 0xff, 0x00, 0x03, 0xff, 0xc7, 0xff, 0x93, 0xff, 0xfc, 0x00, 0x00,
0x1f, 0xff, 0xff,
  0xff, 0xff, 0xfe, 0x00, 0x0f, 0xff, 0x89, 0xff, 0x87, 0xef, 0xf3, 0x80, 0x00,
0x1f, 0xff, 0xff,
  0xff, 0xff, 0xfe, 0x00, 0x1f, 0xff, 0x83, 0xff, 0xc7, 0xdf, 0xdf, 0xc0, 0x00,
0x0f, 0xff, 0xff,
  0xff, 0xff, 0xfc, 0x00, 0x3d, 0xff, 0x83, 0xff, 0xff, 0xff, 0x7f, 0xe0, 0x00,
0x07, 0xff, 0xff,
  0xff, 0xff, 0xfc, 0x00, 0xff, 0xff, 0xc0, 0x3f, 0xff, 0x79, 0xff, 0xf0, 0x00,
0x07, 0xff, 0xff,
  0xff, 0xff, 0xf8, 0x01, 0xff, 0xff, 0xe0, 0x1d, 0xff, 0xef, 0xff, 0x9c, 0x00,
0x03, 0xff, 0xff,
  0xff, 0xff, 0xf0, 0x03, 0xff, 0xff, 0xe7, 0x0f, 0xcf, 0xff, 0xf9, 0xfe, 0x00,
0x01, 0xff, 0xff,
  0xff, 0xff, 0xf0, 0x07, 0xff, 0xde, 0xc7, 0x0f, 0xff, 0xff, 0x9f, 0xff, 0x00,
0x01, 0xff, 0xff,
  0xff, 0xff, 0xe0, 0x0c, 0x3f, 0xff, 0xc6, 0x0f, 0xff, 0xfc, 0xff, 0xff, 0x80,
0x01, 0xff, 0xff,
  0xff, 0xff, 0xe0, 0x1f, 0xf0, 0xff, 0xe0, 0x0f, 0xff, 0xff, 0xff, 0xff, 0x80,
0x00, 0xff, 0xff,
  0xff, 0xff, 0xe0, 0x1f, 0xff, 0xff, 0xf0, 0x1f, 0xff, 0xff, 0xff, 0xf9, 0xc0,
0x00, 0xff, 0xff,
  0xff, 0xff, 0xc0, 0x3f, 0xff, 0xff, 0xf8, 0x7f, 0xff, 0xff, 0xc0, 0x7f, 0xe0,
0x00, 0x7f, 0xff,
  0xff, 0xff, 0xc0, 0x60, 0x7f, 0xff, 0xff, 0x7f, 0xff, 0xff, 0xff, 0xff, 0xe0,
0x00, 0x7f, 0xff,
  0xff, 0xff, 0xc0, 0x7f, 0xfc, 0x7f, 0xff, 0x7f, 0xff, 0xff, 0xff, 0xff, 0xf0,
0x00, 0x7f, 0xff,
  0xff, 0xff, 0x80, 0xff, 0xff, 0xff, 0xff, 0x7f, 0xff, 0xff, 0xff, 0xff, 0xf8,
0x00, 0x7f, 0xff,
  0xff, 0xff, 0x80, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf8,
0x00, 0x3f, 0xff,
  0xff, 0xff, 0x81, 0xff, 0xff, 0x3f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf8,
0x00, 0x3f, 0xff,
  0xff, 0xff, 0x81, 0xff, 0xc3, 0xff, 0xff, 0xbf, 0xff, 0xff, 0xff, 0xff, 0xfc,
0x00, 0x3f, 0xff,
```

```
  0xff, 0xff, 0x81, 0xf8, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfc,
0x00, 0x3f, 0xff,
  0xff, 0xff, 0x83, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x8f, 0xfc,
0x00, 0x3f, 0xff,
  0xff, 0xff, 0x83, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfe, 0x00, 0x1f, 0xfe,
0x00, 0x3f, 0xff,
  0xff, 0xff, 0x83, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf8, 0x00, 0x00, 0x1f, 0xfe,
0x00, 0x3f, 0xff,
  0xff, 0xff, 0x83, 0xff, 0xff, 0xff, 0xff, 0xc0, 0x00, 0x00, 0x00, 0x3f, 0xfe,
0x00, 0x21, 0xff,
  0xff, 0xff, 0x83, 0xff, 0xff, 0xff, 0x00, 0x00, 0x00, 0x00, 0x00, 0x7f, 0xfe,
0x00, 0x7f, 0xbf,
  0xff, 0xff, 0x83, 0xff, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x7f, 0xfe,
0x01, 0xff, 0xcf,
  0xff, 0xff, 0x83, 0xff, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xff, 0xfe,
0x03, 0xff, 0xe7,
  0xff, 0xff, 0x83, 0xff, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0xff, 0xfe,
0x03, 0xff, 0xe3,
  0xff, 0xff, 0xc3, 0xff, 0xe0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0xff, 0xfe,
0x07, 0xff, 0xe3,
  0xff, 0xff, 0xc3, 0xff, 0xe0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0xff, 0xfe,
0x07, 0xff, 0xf3,
  0xff, 0xff, 0xc3, 0xff, 0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0xff, 0xfe,
0x0f, 0xff, 0xf1,
  0xff, 0xff, 0xe3, 0xff, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f, 0xff, 0xfe,
0x0f, 0xff, 0xf1,
  0xff, 0xff, 0xe3, 0xff, 0xfc, 0x00, 0x00, 0x00, 0x00, 0x00, 0x1f, 0xff, 0xfe,
0x0f, 0xff, 0xe1,
  0xff, 0xfd, 0xe1, 0xff, 0xfe, 0x00, 0x00, 0x00, 0x00, 0x00, 0x1f, 0xff, 0xfc,
0x07, 0xff, 0xe3,
  0xff, 0xff, 0xf1, 0xff, 0xff, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3f, 0xff, 0xfc,
0x07, 0xff, 0xe3,
  0xff, 0xff, 0xd1, 0xff, 0xff, 0x80, 0x00, 0x00, 0x00, 0x00, 0x7f, 0xff, 0xfc,
0x03, 0xff, 0xc3,
  0xf7, 0xff, 0xf8, 0xff, 0xff, 0xc0, 0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xf8,
0x01, 0xff, 0x87,
  0xe7, 0xff, 0xf8, 0xff, 0xff, 0xe0, 0x00, 0x00, 0x00, 0x01, 0xff, 0xff, 0xf8,
0x00, 0xfe, 0x0f,
  0xe7, 0xff, 0xfc, 0x7f, 0xff, 0xf0, 0x00, 0x00, 0x00, 0x07, 0xff, 0xff, 0xf0,
0x00, 0x00, 0x3f,
  0xc7, 0xff, 0xf6, 0x7f, 0xff, 0xf8, 0x00, 0x00, 0x00, 0x0f, 0xff, 0xff, 0xf0,
0x00, 0x03, 0xff,
  0xc7, 0xff, 0xf6, 0x3f, 0xff, 0xfe, 0x00, 0x00, 0x00, 0x1f, 0xff, 0xff, 0xc0,
0x00, 0x07, 0xff,
  0xc7, 0xff, 0xf7, 0x1f, 0xff, 0xff, 0x00, 0x00, 0x00, 0x7f, 0xff, 0xff, 0x80,
0x00, 0x07, 0xff,
  0xc3, 0xff, 0xf1, 0x9f, 0xff, 0xff, 0xc0, 0x00, 0x01, 0xff, 0xff, 0xff, 0x00,
0x00, 0x07, 0xff,
  0xe3, 0xff, 0xf0, 0x4f, 0xff, 0xff, 0xf0, 0x00, 0x07, 0xff, 0xff, 0xfe, 0x00,
0x00, 0x0f, 0xff,
```

```
  0xe1, 0xff, 0xf0, 0x27, 0xff, 0xff, 0xfe, 0x00, 0x3f, 0xff, 0xff, 0xf8, 0x00,
0x00, 0x0f, 0xff
};

volatile boolean flag = false;
int symbolIndex = 0;
bool buttonPressed = false;

void setup() {
    Serial.begin(9600);
    pinMode(2, INPUT_PULLUP);
    Wire.begin();
    display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
    display.clearDisplay();
    display.display();
}

void buttonPressedHandler() {
  if (!buttonPressed) {
    buttonPressed = true;
  }
}

void loop() {
  if (!digitalRead(2)) {
    delay(10);
    if (!digitalRead(2)) {
      attachInterrupt(digitalPinToInterrupt(2), buttonPressedHandler, FALLING);
      delay(50); // Wait for interrupt to be handled
      detachInterrupt(digitalPinToInterrupt(2));

    if (buttonPressed) {
        buttonPressed = false;
            symbolIndex++;
            if (symbolIndex > 5) {
                symbolIndex = 1;
            }

            display.clearDisplay();
            switch (symbolIndex) {
                case 1:
                    display.drawBitmap(0, 0, symbol1Bitmap, 128, 64, WHITE);
                    break;
                case 2:
                    display.drawBitmap(0, 0, symbol2Bitmap, 128, 64, WHITE);
                    break;
                case 3:
                    display.drawBitmap(30, 0, symbol3Bitmap, 128, 64, WHITE);
                    break;
                case 4:
                    display.drawBitmap(0, 0, MarilynMonroe, 128, 64, WHITE);
```

```
            break;
        case 5:
            display.drawBitmap(0, 0, Doraemon, 128, 64, WHITE);
            break;
    }
    display.display();
    delay(1000);
}}}}
```

Flowchart:



2. Credit task

Credit: Using a timer interrupt, create a digital clock and display in the LCD (LCD (SSD1306 OLED Display or TFT-LCD display). A push button is used as well. When push button is pressed, hardware interrupt is detection, the digital clock will be converted to analog clock display.

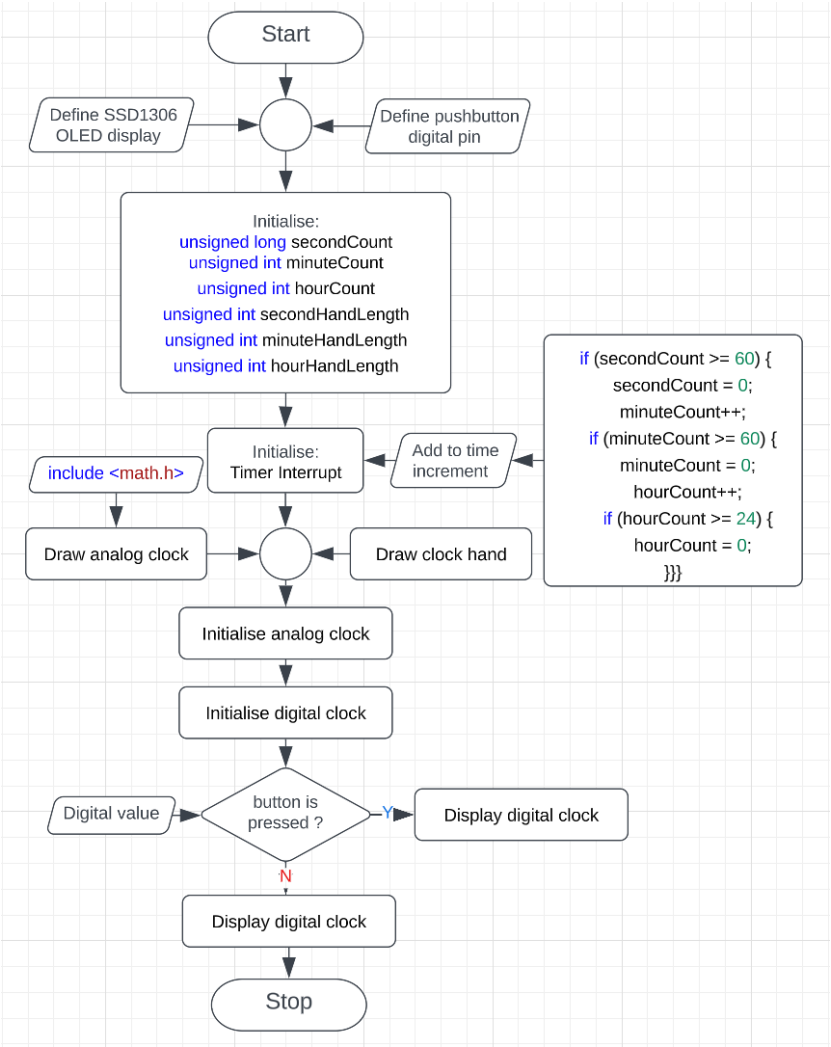
Arduino Sketch:

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <math.h>

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1
const int button1Pin = 1;
const int button2Pin = 2;
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

unsigned long secondCount = 55;
unsigned int minuteCount = 35;
unsigned int hourCount = 10;
unsigned int secondHandLength = 23;
unsigned int minuteHandLength = 18;
unsigned int hourHandLength = 14;

void setup() {
  Serial.begin(9600);
  Wire.begin();
  pinMode(button1Pin, INPUT_PULLUP);
  pinMode(button2Pin, INPUT_PULLUP);

  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println("SSD1306 initialization failed");
    while(true);
  }

  display.clearDisplay();
  display.setTextColor(WHITE);
  display.setTextSize(1);
  display.setCursor(5, 25);
  display.println("Analog Digital Clock");
  display.setCursor(33, 35);
  display.println("by Khoa Le");
  display.display();
  delay(1000);
  display.clearDisplay();

  noInterrupts();
  TCCR1A = 0;
  TCCR1B = 0;
  TCNT1 = 0;
  OCR1A = 31250;
  TCCR1B |= (1 << WGM12);
  TCCR1B |= (1 << CS12) | (1 << CS10);
  TIMSK1 |= (1 << OCIE1A);
  interrupts();
}
```

```
ISR(TIMER1_COMPA_vect) {
  // increment the sec/min count and reset it to 0 if it reaches 60
  secondCount++;
  if (secondCount >= 60) {
    secondCount = 0;
    minuteCount++;
    if (minuteCount >= 60) {
      minuteCount = 0;
      hourCount++;
      if (hourCount >= 24) {
        hourCount = 0;
      }}}
  Serial.print("Time Clock: ");
  Serial.print(hourCount);
  Serial.print(":");
  Serial.print(minuteCount);
  Serial.print(":");
  Serial.println(secondCount);
}

void analog(void) {
  display.clearDisplay();
  display.drawCircle(SCREEN_WIDTH / 2, SCREEN_HEIGHT / 2, 30, WHITE);
  for (int i = 0; i < 60; i++) {
    int angle = i * 6;
    float x = SCREEN_WIDTH / 2 + sin(angle * (PI / 180.0)) * 28;
    float y = SCREEN_HEIGHT / 2 - cos(angle * (PI / 180.0)) * 28;
    display.drawPixel(x, y, WHITE);
  }
  int hourAngle = hourCount * 30 + minuteCount / 60;
  float hourX = SCREEN_WIDTH / 2 + sin(hourAngle * (PI / 180.0)) * hourHandLength;
  float hourY = SCREEN_HEIGHT / 2 - cos(hourAngle * (PI / 180.0)) * hourHandLength;
  display.drawLine(SCREEN_WIDTH / 2, SCREEN_HEIGHT / 2, hourX, hourY, WHITE);

  int minuteAngle = minuteCount * 6 + secondCount / 60;
  float minuteX = SCREEN_WIDTH / 2 + sin(minuteAngle * (PI / 180.0)) *
minuteHandLength;
  float minuteY = SCREEN_HEIGHT / 2 - cos(minuteAngle * (PI / 180.0)) *
minuteHandLength;
  display.drawLine(SCREEN_WIDTH / 2, SCREEN_HEIGHT / 2, minuteX, minuteY, WHITE);

  int secondAngle = secondCount * 6;
  float secondX = SCREEN_WIDTH / 2 + sin(secondAngle * (PI / 180.0)) *
secondHandLength;
  float secondY = SCREEN_HEIGHT / 2 - cos(secondAngle * (PI / 180.0)) *
secondHandLength;
  display.drawLine(SCREEN_WIDTH / 2, SCREEN_HEIGHT / 2, secondX, secondY, WHITE);
  display.display();
}
```

```
void digital(void) {
    display.clearDisplay();
    display.setTextSize(2);
    display.setCursor(15, 20);
    display.print(hourCount);
    display.print(":");
    display.print(minuteCount);
    display.print(":");
    display.print(secondCount);
    display.display();
    delay(1000);
    display.clearDisplay();}

void loop() {
  if (digitalRead(button2Pin) == LOW) {
  display.clearDisplay();
  !analog;
  digital();
  display.clearDisplay();
  }

  else if (digitalRead(button1Pin) == LOW) {
  display.clearDisplay();
  !digital;
  analog();
  display.clearDisplay();
  }}
```

Flowchart:


Start
Define SSD1306 OLED display
Define pushbutton digital pin
Initialise:
unsigned long secondCount
unsigned int minuteCount
unsigned int hourCount
unsigned int secondHandLength
unsigned int minuteHandLength
unsigned int hourHandLength
if (secondCount >= 60) {
secondCount = 0;
minuteCount++;
if (minuteCount >= 60) {
minuteCount = 0;
hourCount++;
if (hourCount >= 24) {
hourCount = 0;
}}}
Initialise:
Timer Interrupt
Add to time increment
include <math.h>
Draw analog clock
Draw clock hand
Initialise analog clock
Initialise digital clock
Digital value
button is pressed ?
Y
Display digital clock
N
Display digital clock
Stop