

Assignment #7 - Mini Clock

65010077 - นายกิตติพิศ หลำบางช้าง

65010107 - นายศรีษฐ์ พราหมณ์

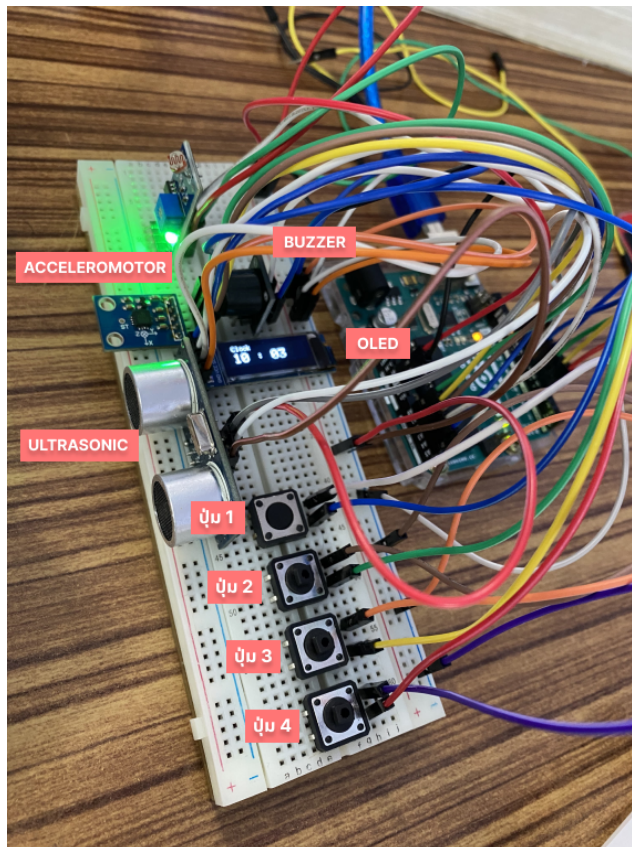
แนวคิดการออกแบบ

ได้รับแรงบันดาลใจมาจาก Apple Watch ที่มีฟังก์ชันในการทำงานที่หลากหลาย ทำให้ผู้ใช้สามารถใช้ฟังก์ชันในการทำงานได้อย่างหลากหลายและมีประสิทธิภาพในการใช้ชีวิตประจำวันที่มากขึ้น

ฟังก์ชันของ Mini Clock

- นาฬิกาดูเวลา บน OLED สามารถบันทึกเวลาได้
- นาฬิกา Stopwatch
- นาฬิกาปลุก
- นาฬิกา Timer
- ปรับความสว่างของหน้าจอ OLED ตามความเข้มแสง
- Sensor ตรวจจับมือ เพื่อหยุดเสียงของนาฬิกา

การใช้งานโดยย่อ



โหมดที่ 1 (Clock)

- โหมดนี้จะทำงานเหมือนนาฬิกาปกติ มีระบบการบันทึกเวลาและสามารถตั้งค่าเวลาในหน่วยนาที่และชั่วโมง
- วิธีการใช้งาน
 - ปุ่ม 1 : ปุ่มปรับชั่วโมง
 - ปุ่ม 2 : ปุ่มปรับนาที่
 - ปุ่ม 3 : ปุ่มเปลี่ยน Mode

โหมดที่ 2 (Stopwatch)

- โหมดนาฬิกาจับเวลา เมื่อเปลี่ยนมาโหมดนี้จะทำการจับเวลาในหน่วยนาที่และวินาทีไปเรื่อยจนกระทั่งเรากดหยุด และยังสามารถกดพัก และกดไปต่อได้
- วิธีการใช้งาน
 - ปุ่ม 1 : ปุ่ม Start และ Pause การนับเวลา
 - ปุ่ม 2 : ปุ่ม Reset เวลา
 - ปุ่ม 3 : ปุ่มเปลี่ยน Mode

โหมดที่ 3 (Alarm)

- โหมดนาฬิกาปลุก เมื่อเปลี่ยนมาโหมดนี้เราสามารถตั้งเวลาปลุกได้และเมื่อถึงเวลาที่เรากำหนดเอาไว้จะมีเสียงปลุกพร้อมกับข้อความแจ้งเตือนว่า WAKE UP!! และสามารถใช้ในการตรวจจับมือ เพื่อหยุดเสียงนาฬิกาปลุกได้
- วิธีการใช้งาน
 - ปุ่ม 1 : ปุ่มสำหรับปรับชั่วโมง
 - ปุ่ม 2 : ปุ่มสำหรับปรับนาที่
 - ปุ่ม 3 : ปุ่มเปลี่ยน Mode
 - ปุ่ม 4 : ปุ่ม On / Off นาฬิกาปลุก

โหมดที่ 4 (Timer)

- โหมดนาฬิกานับถอยหลัง เมื่อเปลี่ยนมาโหมดนี้เราสามารถตั้งค่าเวลานับถอยหลังได้และเมื่อทำการกดปุ่มเริ่ม นาฬิกาก็จะทำการนับถอยหลังไปเรื่อยๆจนถึงเวลาที่เรากำหนดเอาไว้ เมื่อถึงเวลาที่เรากำหนดเอาไว้ก็จะมีเสียงแจ้งเตือนพร้อมกับข้อความว่า FINISH และสามารถใช้ในการตรวจจับมือ เพื่อหยุดเสียงแจ้งเตือนได้
- วิธีการใช้งาน
 - ปุ่ม 1 : ปุ่มสำหรับปรับชั่วโมง
 - ปุ่ม 2 : ปุ่มสำหรับปรับนาที่ / ปุ่ม Reset เวลา
 - ปุ่ม 3 : ปุ่มเปลี่ยน Mode
 - ปุ่ม 4 : ปุ่ม Start และ Pause การนับเวลา

โหมดพิเศษ

- ปรับแสงจอ-> สามารถหรี่แสงจอลงได้ แต่เมื่ออยู่ในที่มืดมากจนเกินไปจะไม่สามารถทำงานได้
- หมุนจอ -> สามารถหมุนการแสดงผลของหน้าจอตามการเอียงของหน้าจอ
- การตรวจจับมือ -> สามารถใช้มือ เพื่อให้ Sensor ตรวจจับ แล้วเสียงของนาฬิกาจะหยุด

Code

Settings :

```

1  #include <Wire.h>
2  #include <Adafruit_GFX.h>
3  #include <Adafruit_SSD1306.h>
4  #include <EEPROM.h>
5  #include <TimerOne.h>
6
7  struct Time {
8      int hour;
9      int min;
10     int sec;
11 };
12
13 // ==== OLED SETTING ====
14 #define OLED_RESET -1
15 #define SCREEN_WIDTH 128
16 #define SCREEN_HEIGHT 32
17 #define SCREEN_ADDRESS 0x3C
18 Adafruit_SSD1306 OLED(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
19
20 // ==== ACCELEROMETER SETTING ====
21 const unsigned int X_AXIS_PIN = 0;
22 const unsigned int Y_AXIS_PIN = 1;
23 const unsigned int Z_AXIS_PIN = 2;
24 const unsigned int NUM_AXES = 3;
25 const unsigned int PINS[NUM_AXES] = {
26     X_AXIS_PIN, Y_AXIS_PIN, Z_AXIS_PIN
27 };
28 const unsigned int BUFFER_SIZE = 16;
29 int buffer[NUM_AXES][BUFFER_SIZE];
30 int buffer_pos[NUM_AXES] = { 0 };

```

```

1  // ==== BUTTON SETTING ====
2  #define BTN_1 7
3  #define BTN_2 6
4  #define BTN_3 5
5  #define BTN_4 4
6  int button[4] = { BTN_1, BTN_2, BTN_3, BTN_4 };
7  int reading[4];
8  int buttonState[4];
9  int lastButtonState[4] = { HIGH, HIGH, HIGH, HIGH };
10 unsigned long long int debounceDelay = 50;
11 unsigned long long int lastDebounceTime[4];
12
13 // ==== BUZZER SETTING ====
14 #define BUZZER_PIN 8
15
16 // ==== CLOCK SETTING ====
17 Time clock = { 0, 0, 0 };
18
19 // ==== COUNT UP CLOCK SETTING ====
20 Time countUp = { 0, 0, 0 };
21 int onCountUpPause = 1;
22
23 // ==== COUNT DOWN CLOCK SETTING ====
24 // Time countDown = { 0, 0, 0 };
25 // int onCountDownStart = 0, onCountDownPause = 0;
26
27 // ==== ALARM CLOCK SETTING ====
28 Time alarm = { 0, 0, 0 };
29 Time alarmSave = { 0, 0, 0 };
30 int onAlarm = 0;
31
32 // ==== MODE SETTING ====
33 #define CLOCK_MODE 0
34 #define COUNTER_UP_MODE 1
35 #define ALARM_MODE 2
36 // #define COUNTER_DOWN_MODE 3
37 int MODE = CLOCK_MODE;

```

Code ในส่วนของการตั้งค่า Pin ของโมดูล และตั้งค่าตัวแปรสำหรับฟังก์ชันต่างๆของนาฬิกา

Accelerometer :

```

1 // ==== ACCELEROMETER ====
2 int get_axis(const int axis) {
3     delay(1);
4     buffer[axis][buffer_pos[axis]] = analogRead(PINS[axis]);
5     buffer_pos[axis] = (buffer_pos[axis] + 1) % BUFFER_SIZE;
6     long sum = 0;
7     for (unsigned int i = 0; i < BUFFER_SIZE; i++)
8         sum += buffer[axis][i];
9     return round(sum / BUFFER_SIZE);
10 }
11
12 int get_x() {
13     return get_axis(0);
14 }
15 int get_y() {
16     return get_axis(1);
17 }
18 int get_z() {
19     return get_axis(2);
20 }

```

Code ในส่วนของการรับค่าแกน x, y, z จาก Accelerometer

Debounce :

```

1 // ==== DEBOUNCE BUTTON ====
2 int debounce(int i) {
3     int isChange = 0;
4     reading[i] = digitalRead(button[i]);
5     if (reading[i] != lastButtonState[i]) {
6         lastDebounceTime[i] = millis();
7     }
8     if ((millis() - lastDebounceTime[i]) >= debounceDelay) {
9         if (reading[i] != buttonState[i]) {
10             buttonState[i] = reading[i];
11             isChange = 1;
12         }
13     }
14     lastButtonState[i] = reading[i];
15     return isChange;
16 }

```

Code ในส่วนของการ Debounce ปุ่มเพื่อไม่ให้ปุ่มเกิดการ Error เมื่อทำการกดปุ่ม

Interrupt :

```

1 // ==== TIMER ONE ====
2 void interruptClock() {
3     timer();
4     countUpTime();
5     change_timestart_alarm();
6     countDownTime();
7     digitalWrite(trigPin, HIGH);
8     delayMicroseconds(10);
9     digitalWrite(trigPin, LOW);
10
11     duration_us = pulseIn(echoPin, HIGH);
12     distance_cm = 0.017 * duration_us;
13
14     Serial.println(distance_cm);
15     if (finishCountDown && distance_cm < 7) {
16         OLED.clearDisplay();
17         countDownCheck = 0;
18         finishCountDown = 0;
19         onStartCountDown = 0;
20     } else {
21         countDownCheck = 1;
22     }
23
24     if (onAlarm && distance_cm < 7) {
25         OLED.clearDisplay();
26         onAlarm = 0;
27         alarmCheck = 0;
28     } else {
29         alarmCheck = 1;
30     }
31 }

```

Code ในส่วนของการ interrupt หรือก็คือการ Update เวลาของนาฬิกาทุกๆ 1 วินาที

Clock :

```

1 // ==== TIMER CLOCK ====
2 void timer() {
3     clock.sec += 1;
4     clock.min += clock.sec / 60;
5     clock.sec %= 60;
6     clock.hour += clock.min / 60;
7     clock.min %= 60;
8     clock.hour %= 24;
9     if (clock.sec == 0) {
10         EEPROM.update(20, clock.hour);
11         EEPROM.update(21, clock.min);
12     }
13 }
14
15 void change_hour() {
16     clock.hour += 1;
17     EEPROM.update(20, clock.hour);
18     clock.hour %= 24;
19 }
20
21 void change_min() {
22     clock.min += 1;
23     clock.hour += clock.min / 60;
24     clock.hour %= 24;
25     EEPROM.update(21, clock.min);
26     clock.min %= 60;
27 }

```

Code ในส่วนของการทำงานของนาฬิกาธรรมดา การบันทึกชั่วโมง และนาทิจึง EEPROM

Stopwatch :

```

1  // ==== COUNT UP CLOCK ====
2  void countUpTime() {
3      if (!onCountUpPause) {
4          countUp.sec += 1;
5          countUp.min += countUp.sec / 60;
6          countUp.sec %= 60;
7          countUp.hour += countUp.min / 60;
8          countUp.min %= 60;
9      }
10 }
11
12 void resetCountUpTime() {
13     countUp.hour = 0;
14     countUp.min = 0;
15     countUp.sec = 0;
16 }

```

Code ในส่วนของนาฬิกาจับเวลา และ Reset การจับเวลา

Timer :

```

1  // ==== COUNT DOWN CLOCK ====
2  void change_min_countdown() {
3      countDown.min += 1;
4      countDown.min %= 60;
5  }
6
7  void change_sec_countdown() {
8      countDown.sec += 1;
9      countDown.sec %= 60;
10 }
11
12 void countDownTime() {
13     if (!onCountDown) {
14         if (countDown.sec == 0 && countDown.min != 0) {
15             if (countDown.min != 0) {
16                 countDown.min -= 1;
17             }
18             countDown.sec = 60;
19         }
20         countDown.sec -= 1;
21         if (countDown.min == 0 && countDown.sec == 0) {
22             onCountDown = 1;
23             finishCountDown = 1;
24         }
25     }
26 }
27
28 void resetcountDownTime() {
29     countDown.hour = 0;
30     countDown.min = 0;
31     countDown.sec = 0;
32 }
33 }

```

Code ของนาฬิกานับถอยหลัง และ Reset เวลา

Alarm Clock :

```

1 // ==== ALARM CLOCK ====
2 void change_hour_alarm() {
3     alarm.hour += 1;
4     alarm.hour %= 24;
5 }
6
7 void change_min_alarm() {
8     alarm.min += 1;
9     alarm.min %= 60;
10 }
11
12 void change_timestart_alarm() {
13     if (clock.sec == 0) {
14         alarm.min = clock.min;
15     }
16     if (clock.min == 0) {
17         alarm.hour = clock.hour;
18     }
19 }

```

Code ของนาฬิกาปลุก

Clock Mode :

```

1 // ==== CLOCK MODE ====
2 void change_mode() {
3     MODE += 1;
4     MODE %= 4;
5     if (MODE == ALARM_MODE) {
6         alarm.hour = clock.hour;
7         alarm.min = clock.min;
8     }
9 }

```

Code ของการเปลี่ยน Mode ของนาฬิกา

Clock Text :

```

1 // ==== Time Text ====
2 String timerText = "00 : 00";
3 String countUpText = "00 : 00";
4 String countDownText = "00 : 00";
5 String alarmText = "00 : 00";
6 String saveAlarmText = "00 : 00";
7 void time_text() {
8     timerText[0] = (clock.hour / 10) + '0';
9     timerText[1] = (clock.hour % 10) + '0';
10
11     timerText[5] = (clock.min / 10) + '0';
12     timerText[6] = (clock.min % 10) + '0';
13
14     countUpText[0] = (countUp.min / 10) + '0';
15     countUpText[1] = (countUp.min % 10) + '0';
16
17     countUpText[5] = (countUp.sec / 10) + '0';
18     countUpText[6] = (countUp.sec % 10) + '0';
19
20     countDownText[0] = (countDown.min / 10) + '0';
21     countDownText[1] = (countDown.min % 10) + '0';
22
23     countDownText[5] = (countDown.sec / 10) + '0';
24     countDownText[6] = (countDown.sec % 10) + '0';
25
26     alarmText[0] = (alarm.hour / 10) + '0';
27     alarmText[1] = (alarm.hour % 10) + '0';
28
29     alarmText[5] = (alarm.min / 10) + '0';
30     alarmText[6] = (alarm.min % 10) + '0';
31
32     saveAlarmText[0] = (alarmSave.hour / 10) + '0';
33     saveAlarmText[1] = (alarmSave.hour % 10) + '0';
34
35     saveAlarmText[5] = (alarmSave.min / 10) + '0';
36     saveAlarmText[6] = (alarmSave.min % 10) + '0';
37 }

```

```

1 void display_text(int x1, int y1, String name, int x2, int y2, String timer, int size1, int size2) {
2     OLED.setTextSize(size1);
3     OLED.setCursor(x1, y1);
4     OLED.println(name);
5     OLED.setTextSize(size2);
6
7     OLED.setCursor(x2, y2);
8     OLED.println(timer);
9 }
10
11 void popup_text(String text) {
12     OLED.clearDisplay();
13     OLED.setTextColor(WHITE);
14     if (LDRvalue > 900) {
15         OLED.dim(true);
16     } else {
17         OLED.dim(false);
18     }
19     OLED.setCursor(10, 10);
20     OLED.setTextSize(2);
21     OLED.print(text);
22     OLED.display();
23 }

```

Code ของการแสดงผลเวลาบนหน้าจอนาฬิกา

Arduino Setup :

```

1 void setup() {
2     Serial.begin(9600);
3     if (!OLED.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
4         Serial.println("SSD1306 allocation failed");
5     } else {
6         Serial.println("All OLED Start Work !!!");
7     }
8     for (int i = 0; i < 4; i++) {
9         pinMode(button[i], INPUT_PULLUP);
10    }
11    Timer1.initialize(1000000);
12    Timer1.attachInterrupt(interruptClock);
13    // ==== WRITE EEPROM ====
14    // EEPROM.write(20, clock.hour);
15    // EEPROM.write(21, clock.min);
16
17    // ==== READ EEPROM ====
18    clock.hour = EEPROM.read(20);
19    clock.min = EEPROM.read(21);
20
21    pinMode(trigPin, OUTPUT);
22    pinMode(echoPin, INPUT);
23 }

```

Code ของ Setup Arduino

Loop :

```

1 void loop() {
2
3
4   time_text();
5
6   LDRvalue = analogRead(LDR_PIN);
7
8   OLED.clearDisplay();
9   OLED.setTextColor(WHITE);
10
11  if (LDRvalue > 900) {
12    OLED.dim(true);
13  } else {
14    OLED.dim(false);
15  }
16
17  if (debounce(0)) {
18    if (!digitalRead(button[0])) {
19      switch (MODE) {
20        case CLOCK_MODE:
21          change_hour();
22          break;
23        case COUNTER_UP_MODE:
24          onCountUpPause = onCountUpPause == 1 ? 0 : 1;
25          break;
26        case ALARM_MODE:
27          change_hour_alarm();
28          break;
29        case COUNTER_DOWN_MODE:
30          if (!onStartCountDown) {
31            change_min_countdown();
32          }
33          break;
34      }
35    }
36  }

```

```

1  if (onAlarm && alarmSave.hour == clock.hour && alarmSave.min == clock.min && clock.sec == 0) {
2    while (alarmCheck) {
3      tone(BUZZER_PIN, 440, 100);
4      popup_text("WAKE UP!!");
5      if (debounce(0)) {
6        if (!digitalRead(button[0])) {
7          OLED.clearDisplay();
8          onAlarm = 0;
9          break;
10         }
11       }
12     }
13   }
14
15   if (finishCountDown) {
16     while (countDownCheck) {
17       tone(BUZZER_PIN, 440, 100);
18       popup_text("FINISH!!");
19       if (debounce(0)) {
20         if (!digitalRead(button[0])) {
21           OLED.clearDisplay();
22           finishCountDown = 0;
23           onStartCountDown = 0;
24           break;
25         }
26       }
27     }
28   }
29
30   if (get_X() > 360) {
31     OLED.setRotation(2);
32   } else {
33     OLED.setRotation(0);
34   }
35
36   switch (MODE) {
37     case CLOCK_MODE:
38       display_text(10, 0, "Clock", 10, 12, timerText, 1, 2);
39       break;
40     case COUNTER_UP_MODE:
41       display_text(10, 0, "Stopwat:h", 10, 12, countUpText, 1, 2);
42       break;
43     case ALARM_MODE:
44       display_text(10, 0, "Alarm", 10, 9, alarmText, 1, 2);
45       display_text(10, 25, saveAlarmText, 80, 0, onAlarm == 0 ? "OFF" : "ON", 1, 1);
46       break;
47     case COUNTER_DOWN_MODE:
48       display_text(10, 0, "Timer", 10, 12, countDownText, 1, 2);
49       break;
50   }
51
52   OLED.display();
53 }

```

```

1  if (debounce(1)) {
2    if (!digitalRead(button[1])) {
3      switch (MODE) {
4        case CLOCK_MODE:
5          change_min();
6          break;
7        case COUNTER_UP_MODE:
8          onCountUpPause = 1;
9          resetCountUpTime();
10         break;
11        case ALARM_MODE:
12          change_min_alarm();
13          break;
14        case COUNTER_DOWN_MODE:
15          if (!onStartCountDown) {
16            change_sec_countdown();
17          } else if (onStartCountDown) {
18            resetcountDownTime();
19            onStartCountDown = 0;
20            onCountDown = 1;
21          }
22          break;
23      }
24    }
25  }
26
27  if (debounce(2)) {
28    if (!digitalRead(button[2])) {
29      change_mode();
30    }
31  }
32
33  if (debounce(3)) {
34    if (!digitalRead(button[3])) {
35      switch (MODE) {
36        case ALARM_MODE:
37          onAlarm = onAlarm == 0 ? 1 : 0;
38          if (onAlarm) {
39            alarmSave.hour = alarm.hour;
40            alarmSave.min = alarm.min;
41          }
42          break;
43        case COUNTER_DOWN_MODE:
44          if (!(countDown.min == 0 && countDown.sec == 0)) {
45            onCountDown = onCountDown == 1 ? 0 : 1;
46            onStartCountDown = 1;
47          }
48          break;
49      }
50    }
51  }

```

Code การทำงานของนาฬิกา