ภาคผนวก

ภาคผนวก ก.

- โปรแกรมสำหรับควบคุมระบบและแสดงผลการวัด
- โปรแกรมควบคุมมอเตอร์

โปรแกรมสำหรับควบคุมระบบและแสดงผลการวัด

```
#include <SPI.h>
#define ADF4351_CS 3                    // ขา CS สำหรับ ADF4351
#define LOCK 2                          // ขาสำหรับตรวจสอบสถานะการล็อค
#define ADF_CE 4                        // ขาสำหรับสัญญาณเปิด/ปิด ADF4351
#include <AD7705.h>
AD7705 ad(8, 7);
const int sw = 10;                      // ขาสำหรับควบคุมการทำงานของมอเตอร์
const int button = 5;                   // ขาเชื่อมต่อสวิตซ์ปุ่มกด
int button_state = 0;
long data;                              // ตัวแปรสำหรับเก็บข้อมูลจาก AD7705
uint32_t registers[6] = {               // ค่ารีจิสเตอร์ที่ใช้กำหนดค่าของ ADF4351
  0xA10020, 0x8008029, 0x4E42, 0x4B3,
0x950074, 0x580005
};
void WriteReg32(const uint32_t value) {  // ฟังก์ชันสำหรับเขียนข้อมูล 32 บิตไปยัง
    digitalWrite(ADF4351_CS, LOW);       ADF4351 ผ่าน SPI
  for (int i = 3; i >= 0; i--) {         // ส่งข้อมูลทีละ 8 บิต (4 ไบต์)
    SPI.transfer((value >> 8 * i) & 0xFF); }  // เลื่อนบิตและส่งข้อมูลทีละไบต์
  digitalWrite(ADF4351_CS, HIGH);
  digitalWrite(ADF4351_CS, LOW);
}
void SetADF4351() {                     // ฟังก์ชันสำหรับตั้งค่ารีจิสเตอร์ของ ADF4351
  for (int i = 5; i >= 0; i--) {
    WriteReg32(registers[i]);           // เขียนค่าของแต่ละรีจิสเตอร์
  }}
void setup() {
  Serial.begin(38400);
  ad.conf();                            // กำหนดค่าเริ่มต้นให้ AD7705
  delay(50);
  pinMode(LOCK, INPUT);
  pinMode(ADF4351_CS, OUTPUT);
  digitalWrite(ADF4351_CS, HIGH);
  SPI.begin();                          // ตั้งค่าโหมดการสื่อสาร SPI
  SPI.setDataMode(SPI_MODE0);           // ส่งข้อมูลบิตมาก่อน (MSB first)
```

```
  SPI.setBitOrder(MSBFIRST);
  pinMode(ADF_CE, OUTPUT);
  pinMode(button, INPUT);
  pinMode(sw, INPUT);
  digitalWrite(sw, HIGH);
}
void loop() {
  button_state = digitalRead(button);        // อ่านสถานะของปุ่ม
  {
    if (button_state == LOW) {               // หากปุ่มถูกกด
      digitalWrite(ADF_CE, HIGH);            // เปิดการทำงาน ADF4351
      delay(100);
      SetADF4351();                          // ตั้งค่า ADF4351
      digitalWrite(ADF_CE, HIGH);            // ยืนยันการเปิดใช้งาน ADF4351
      delay(100);
    } else {
      digitalWrite(ADF_CE, LOW);
      delay(100);
   }}
  if (digitalRead(sw) == LOW) {              // ถ้าสวิตซ์ขา D5 ถูกกด
      ad.setSetup(1, 1, 25, 0);              // ตั้งค่า AD7705 (CH1, Gain 1, 25 Hz,
      delay(50);                             Unipolar)
      data = ad.read_unipolar();             // อ่านข้อมูลแบบ Unipolar จาก AD7705
      delay(50);
      Serial.print("CH1 ");
      Serial.print(data * 0.0000763);
      Serial.print(" Volts");
      delay(100);
  }}
```

โปรแกรมสำหรับควบคุมมอเตอร์

```
int a = 0;                                  // ตัวแปรนับรอบการทำงานของลูป
const int stepPin1 = 3;                     // กำหนดขาสัญญาณ Step สำหรับมอเตอร์
const int dirPin1 = 2;                      // กำหนดขาสัญญาณ Direction สำหรับมอเตอร์
const int enPin1 = 5;                       // กำหนดขาสัญญาณ Enable สำหรับมอเตอร์
const int s = 12;                           // กำหนดขาสัญญาณอินพุตจากบอร์ดควบคุม
void setup() {
  Serial.begin(38400);
  pinMode(stepPin1, OUTPUT);
  pinMode(dirPin1, OUTPUT);
  pinMode(enPin1, OUTPUT);
  digitalWrite(enPin1, LOW);
  pinMode(s, INPUT);
}
void loop() {
  if (digitalRead(s) == LOW) {              // ตรวจสอบสถานะของสวิตช์ (ถ้าสวิตช์ถูกกด)
    int bi = a++;                           // เพิ่มค่า a และเก็บไว้ในตัวแปร bi
    Serial.println(bi);                     // แสดงค่าของ bi บน Serial Monitor
    if (bi >= 5) {                          // หาก bi มีค่าเท่ากับหรือมากกว่า 5
      digitalWrite(dirPin1, LOW);           // ตั้งค่าทิศทางมอเตอร์ให้หมุนกลับ
    } else {
      digitalWrite(dirPin1, HIGH);          // ตั้งค่าทิศทางมอเตอร์ให้หมุนไปข้างหน้า
      if (bi >= 10) {                       // หาก bi มีค่าเท่ากับหรือมากกว่า 10
        bi = 0;                             // รีเซ็ตค่า bi กลับเป็น 0
      } else {
        bi = bi;                            // คงค่า bi เดิมไว้ (ไม่ทำอะไร)
      }}
    for (int x = 0; x < 200; x++) {         // หมุนมอเตอร์ 200 สเต็ป
      digitalWrite(stepPin1, HIGH);         // สัญญาณ Step = HIGH (เริ่มสเต็ป)
      delayMicroseconds(2000);
      digitalWrite(stepPin1, LOW);          // สัญญาณ Step = LOW (จบสเต็ป)
      delayMicroseconds(2000);
    }}}
```