ภาคผนวก

ภาคผนวก ก.

- โปรแกรมสำหรับควบคุมระบบและแสดงผลการวัด
- โปรแกรมควบคุมมอเตอร์

โปรแกรมสำหรับควบคุมระบบและแสดงผลการวัด

```
#include <SPI.h>
#define ADF4351_CS 3                    // กำหนดขา CS สำหรับ ADF4351
#define LOCK 2                          // กำหนดขาสำหรับสัญญาณล็อก
#define ADF_CE 4                        // กำหนดขาสำหรับสัญญาณเปิด/ปิด
const int button = 5;                   ADF4351
int button_state = 0;                   // กำหนดขาเชื่อมต่อสวิตซ์ปุ่มกด

uint32_t registers[6] = {0xA10020,
0x8008029, 0x4E42, 0x4B3, 0x950074,     // ค่าเริ่มต้นของรีจิสเตอร์ใน ADF4351
0x580005};

int analogPin = A0;
double val = 0;                         // ขาเชื่อมต่อสำหรับอ่านค่า Analog
double S11 = 0;
const int sw = 10;
                                        // ขาเชื่อมบอร์ดควบคุมมอเตอร์
void WriteReg32(const uint32_t value) {
  digitalWrite(ADF4351_CS, LOW);        // ฟังก์ชันสำหรับเขียนข้อมูล 32 บิตไปยัง
  for (int i = 3; i >= 0; i--)          ADF4351 ผ่าน SPI
  SPI.transfer((value >> 8 * i) & 0xFF); // ส่งข้อมูลทีละ 8 บิต (4 ไบต์)
  digitalWrite(ADF4351_CS, HIGH);       // เลื่อนบิตและส่งข้อมูลทีละไบต์
  digitalWrite(ADF4351_CS, LOW);
}

void SetADF4351() {
  for (int i = 5; i >= 0; i--) {
  WriteReg32(registers[i]); } }         // เขียนค่าของแต่ละรีจิสเตอร์

void setup() {
  Serial.begin(38400);
  pinMode(LOCK, INPUT);
  pinMode(ADF4351_CS, OUTPUT);
  digitalWrite(ADF4351_CS, HIGH);
  SPI.begin();
```

```
  SPI.setDataMode(SPI_MODE0);
  SPI.setBitOrder(MSBFIRST);                // ตั้งค่าโหมดการสื่อสาร SPI
  pinMode(ADF_CE, OUTPUT);                  // ส่งข้อมูลบิตมาก่อน (MSB first)
  pinMode(button, INPUT);
  pinMode(sw, INPUT);
  digitalWrite(sw, HIGH);
}

void loop() {
  button_state = digitalRead(button); {
    if (button_state == LOW) {              // อ่านสถานะของปุ่ม
      digitalWrite(ADF_CE, HIGH);           // หากปุ่มถูกกด
      delay(100);                           // เปิดการทำงาน ADF4351
      SetADF4351();
      digitalWrite(ADF_CE, HIGH);
      delay(100);
    } else {
      digitalWrite(ADF_CE, LOW);
      delay(100);
   } }
  if (digitalRead(sw) == LOW) {
    val = analogRead(analogPin);
    S11 = val * 0.488;                      // อ่านค่าจากขา Analog
    Serial.println(S11, 3);
    delay(100);                             // แสดงค่า S11 บน Serial Monitor ทศนิยม
  }}                                        3 ตำแหน่ง
```

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);
    }}}
```