

ภาคผนวก

ภาคผนวก ก

โปรแกรมควบคุมการทำงาน

```

#include <16F877A.h> ; เรียกใช้งานเบอร์ PIC16F877
#device adc=10 ; กำหนดโมดูล ADC เป็นขนาด 10 บิต
#FUSES NOWDT ; ไม่ใช้งาน Watch dog timer
#FUSES HS ; กำหนดโหมดเป็น High speed clock
#FUSES NOPUT ; ไม่ใช้งาน Power Up Timer
#FUSES PROTECT ; ป้องกันการอ่านโค้ดโปรแกรม
#FUSES NODEBUG ; ไม่ใช้โหมดการตรวจสอบการทำงานของ
; ของโปรแกรม
#FUSES NOBROWNOUT ; ไม่ใช้งาน brownout reset
#FUSES NOLVP ; ไม่ใช้งาน low voltage prgming,
; B3(PIC16) or B5(PIC18) used for I/O
#FUSES NOCPD ; ไม่ใช้งาน EE protection
#FUSES NOWRT ; ไม่ป้องกันการเขียนหน่วยความจำ
#use delay(clock=10MHz) ; ใช้ clock 10 MHz
#use rs232(baud=9600,xmit=pin_b6,rcv=pin_b7,stream=COM1) ; เปิดใช้งานโมดูล UART
; กรณีต้องการตรวจสอบการทำงานของ
; โปรแกรม ปกติจะไม่ได้ใช้งาน
#use fast_io(B) ; กำหนดให้พอร์ต B ทำงานในโหมด Fast
/*****/
/* INCLUDE LIBRARY
*/
/*****/
#include <lcd20x4.c> ; เรียกใช้งานไลบรารี LCD
/*****/
/* STATUS CONTROL
*/
/*****/
#define HI 1 ; กำหนดคำว่า Hi แทนตัวเลข 1
#define LO 0 ; กำหนดคำว่า LO แทนตัวเลข 0
/*****/

```

```

/* DEFINE BUTTONS
*/
/*****/
#define BT_ST()    input(PIN_B0)           ; กำหนดให้พอร์ต B0 เป็นปุ่ม เริ่มการ
                                           ทำงาน
#define BT_OK()   input(PIN_B1)           ; กำหนดให้พอร์ต B1 เป็นปุ่ม ตกลง
#define BT_CL()   input(PIN_B2)           ; กำหนดให้พอร์ต B2 เป็นปุ่ม ยกเลิก
#define BT_UP()   input(PIN_B3)           ; กำหนดให้พอร์ต B3 เป็นปุ่ม เพิ่ม
#define BT_DW()   input(PIN_B4)           ; กำหนดให้พอร์ต B4 เป็นปุ่ม ลด

/*****/
/* SERVO MOTOR
*/
/*****/
#define Servo_Pin(x) output_bit(pin_c0,x) ; กำหนดให้พอร์ต C0 เป็นขาควบคุมเซอร์
                                           โว

/*****/
/* SERVO POSITION
*/
/*****/
#define Servo_Left 312                     ; กำหนดคำว่า Servo_Left แทนค่า 312
                                           หมายถึงตำแหน่ง เซอร์โวหันไปทางซ้าย
#define Servo_Center 490                   ; กำหนดคำว่า Servo_Center แทนค่า 490
                                           หมายถึงตำแหน่ง เซอร์โวอยู่ตรงกลาง
#define Servo_Right 625                    ; กำหนดคำว่า Servo_Right แทนค่า 625
                                           หมายถึงตำแหน่งเซอร์โวหันไปทางขวา

#define T_20MS     6250
unsigned int16 Servo_Position = Servo_Center; ; กำหนดตัวแปร สำหรับเก็บตำแหน่งของ
                                           เซอร์โว

/*****/
/* DEFINE SENSOR
*/
/*****/
#define Sensor()    input(PIN_E0)          ; กำหนดพอร์ต E0 เป็นอินพุตเรท

```



```

unsigned long Num_YLemon; ; ตัวแปรเก็บจำนวนมะนาวสีเหลือง ของ
; โหมดนับมะนาวสีเหลือง

unsigned long GLemon = 0; ; ตัวแปรเก็บจำนวนมะนาวสีเขียว
unsigned long GLemon_Old=-1; ; ตัวแปรเก็บจำนวนมะนาวสีเขียวเก่า
unsigned long YLemon = 0; ; ตัวแปรเก็บจำนวนมะนาวสีเหลือง
unsigned long YLemon_Old=-1; ; ตัวแปรเก็บจำนวนมะนาวสีเหลืองเก่า
#define STOP 0 ; แทนค่า 0 ด้วยสถานะ หยุดทำงาน
#define START 1 ; แทนค่า 1 ด้วยสถานะ ทำงาน
int1 Status = STOP; ; ตัวแปรเก็บสถานะ
int1 Status_Old = START; ; ตัวแปรเก็บสถานะเก่า
#define T3SEC 150 ; แทนค่า 150 แทนเวลา 3 วินาที
#define T15SEC 750 ; แทนค่า 750 แทนเวลา 15 วินาที
int16 count_time;
int16 count_time1;
/*****/
/* Function Prototypes
*/
/*****/
void Show_Menu(void); ; ฟังก์ชันสำหรับโชว์เมนูบนจอ LCD
void Run_Machine(void); ; ฟังก์ชันสำหรับแยกมะนาว
void Setting_Machine(void); ; ฟังก์ชันสำหรับตั้งค่าของมะนาว
void Setup(void); ; ฟังก์ชันสำหรับเลือกโหมด
void Calibrate(void); ; ฟังก์ชันสำหรับปรับค่าสี
void Record_REF_Color(unsigned long color); ; ฟังก์ชันสำหรับบันทึกค่าสีอ้างอิง
unsigned long Read_REF_Color(void); ; ฟังก์ชันสำหรับอ่านค่าสีอ้างอิงจาก
หน่วยความจำ
void Record_Num_GLemon(unsigned long num); ; ฟังก์ชันสำหรับบันทึกจำนวนของมะนาว
สีเขียว
unsigned long Read_Num_GLemon(void); ; ฟังก์ชันสำหรับอ่านจำนวนมะนาวสีเขียว
void Record_Num_YLemon(unsigned long num); ; ฟังก์ชันสำหรับบันทึกจำนวนของมะนาว
สีเหลือง
unsigned long Read_Num_YLemon(void); ; ฟังก์ชันสำหรับอ่านจำนวนมะนาวสี
เหลือง
unsigned long read_color(void); ; ฟังก์ชันสำหรับอ่านสี
/*****/

```

```

/* Function : main()
   */
/*****/
void main (void) ; ฟังก์ชันหลัก
{
PORT_B_PULLUPS(true); ; กำหนดให้พอร์ต B ต่อ Resister
set_tris_b(0x1F); ; กำหนดให้พอร์ต B เป็นอินพุต

// Enable Timer0
setup_timer_0(RTCC_INTERNAL|RTCC_DIV_256); ; เปิดใช้งานไทมเมอร์ 0 จับเวลาขณะเครื่องทำงาน

set_timer0(61);
// Enable timer1
setup_timer_1(T1_INTERNAL|T1_DIV_BY_8); ; เปิดใช้งานไทมเมอร์ 1 สำหรับสร้างพัลส์ควบคุมเซอร์โฆ

set_timer1(0);
// Enable CCP module
setup_ccp1(CCP_PWM); ; เปิดใช้งานโมดูล PWM เพื่อใช้ควบคุมความเร็วมอเตอร์

setup_timer_2(T2_DIV_BY_16,255,1); ; เปิดใช้งานไทมเมอร์ 2 สำหรับใช้งานร่วมกับโมดูล PWM

set_timer2(0);
Motor_Stop(); ; สั่งให้มอเตอร์หยุดทำงาน
// Enable ADC module
setup_adc(ADC_CLOCK_INTERNAL|ADC_CLOCK_DIV_8); ; เปิดใช้งานโมดูลสัญญาณอนาล็อกเป็นดิจิตอล
setup_adc_ports(AN0_AN1_AN2_AN3_AN4); ; เปิดใช้งานโมดูล ADC ช่อง 0,1,2,3,4
// Enable interrupts
enable_interrupts(INT_TIMER0); ; เปิดใช้งานอินเทอร์รัปต์ไทมเมอร์ 0
enable_interrupts(INT_TIMER1); ; เปิดใช้งานอินเทอร์รัปต์ไทมเมอร์ 1
enable_interrupts(GLOBAL); ; เปิดใช้งานอินเทอร์รัปต์ทั้งหมด
// Initail LCD module
lcd_init(); ; กำหนดการทำงาน LCD
if(Read_Status() != 0xAA){ ; อ่านสถานะของหน่วยความจำ
Record_Status(); ; ทำการบันทึกค่าของหน่วยความจำเพื่อ

```

```

Record_REF_Color(600);
Record_Num_GLemon(0);
Record_Num_YLemon(0);
Record_Mode(MODE_NOCNT_LEMON);
}
Ref_Color = Read_REF_Color();
Num_GLemon = Read_Num_GLemon();
Num_YLemon = Read_Num_YLemon();
mode = Read_Mode();
Servo_Position = Servo_Right;
delay_ms(2000);
Servo_Position = Servo_Center;
Show_Menu();
while(true)
{
Run_Machine();
Setting_Machine();
}
}
/*****
/* Function : Timer0_Isr()
*/
/* Interrupt every : 20 ms
*/
*****/
#INT_TIMER0

```

ไม่ให้เข้าในเงื่อนไขอีก
; กำหนดค่าเริ่มต้นของสีอ้างอิงเป็น 600
และบันทึกลงหน่วยความจำ
; กำหนดจำนวนของมะนาวสีเขียวเป็น 0
ในหน่วยความจำ
; กำหนดจำนวนของมะนาวสีเหลืองเป็น 0
ในหน่วยความจำ
; กำหนดโหมดเริ่มต้นให้เป็นโหมดไม่นับ
ลูกมะนาว
; อ่านค่าสีอ้างอิงจากหน่วยความจำ
; อ่านจำนวนมะนาวสีเขียวจาก
หน่วยความจำ
; อ่านจำนวนมะนาวสีเหลืองจาก
หน่วยความจำ
; อ่านโหมดจากหน่วยความจำ
; สั่งให้เซอร์โวหันไปทางขวาเพื่อทิ้งลูก
มะนาวที่อาจจะค้างในเครื่อง
; หน่วงเวลา 2 วินาที
; สั่งให้เซอร์โวไปยังตำแหน่งตรงกลาง
; โชว์เมนู
; โปรแกรมจะทำงานในลูปนี้ตลอด
; เรียกฟังก์ชันแยกลูกมะนาว
; เรียกฟังก์ชันตั้งค่าเครื่องแยกมะนาว
; ฟังก์ชันอินเทอร์รัปต์ไทมเมอร์ 0 เกิดขึ้น

เทอร์รับต์ทุกๆ 10 ms

```

void Timer0_Isr(void)
{
set_timer0(61);
count_time++;           ; บวกเวลาไปอีก 1
count_time1++;         ; บวกเวลาไปอีก 1
}
/*****/
/* Function : Timer1_Isr()
   */
/*****/
#INT_TIMER1             ; ฟังก์ชันอินเทอร์รับต์ไทมเมอร์ 1
void Timer1_Isr(void)
{
static int1 Servo_Stat = 0;
switch(Servo_Stat)     ; เช็คสถานะ ของพัลส์
{
case 0: set_timer1(65536-Servo_Position); ; สำหรับสร้างพัลส์ช่วง High
Servo_Pin(HI);
Servo_Stat = 1;
break;
case 1: set_timer1(65536-(T_20MS-Servo_Position)); ; สำหรับสร้างพัลส์ช่วง Low
Servo_Pin(LO);
Servo_Stat = 0;
break;
}
}
/*****/
/* Function : Show_Menu()
   */
/*****/
void Show_Menu(void)   ; โฉว์เมนูบนจอ LCD
{
lcd_gotoxy(1,1);printf(lcd_putc," Lemon ");
lcd_gotoxy(1,2);printf(lcd_putc,"Green : ");

```

```

lcd_gotoxy(1,3);printf(lcd_putc,"Yellow:      ");
lcd_gotoxy(1,4);printf(lcd_putc,"Status:    ");
}
/*****/
/* Function : Run_Machine()
*/
/*****/
void Run_Machine(void)
{
static int1 BT_Stat;                ; สร้างตัวแปรสำหรับเก็บสถานะของ
                                    สวิตช์
static int step = 0;                ; สร้างตัวแปรสแต็คการทำงานของการแยก
                                    มะนาว
unsigned long color_avg;            ; ตัวแปรเก็บค่าสีที่ได้จากการเฉลี่ยแล้ว
if(GLemon != GLemon_Old){          ; ตรวจสอบว่ามะนาวสีเขียวเปลี่ยนแปลง
                                    หรือไม่
GLemon_Old = GLemon;                ; กำหนดจำนวนของมะนาวเก่าเป็นจำนวน
                                    มะนาวใหม่
lcd_gotoxy(8,2);printf(lcd_putc," %lu    ",GLemon); ; แสดงจำนวนมะนาวสีเขียว
}
if(YLemon != YLemon_Old){          ; ตรวจสอบว่ามะนาวสีเหลืองเปลี่ยนแปลง
                                    หรือไม่
YLemon_Old = YLemon;                ; กำหนดจำนวนของมะนาวเก่าเป็นจำนวน
                                    มะนาวใหม่
lcd_gotoxy(8,3);printf(lcd_putc," %lu    ",YLemon); ; แสดงจำนวนมะนาวสีเหลือง
}
if(Status != Status_Old){          ; อ่านสถานะควบคุมว่าเปลี่ยนแปลงหรือไม่
Status_Old = Status;                ; เปลี่ยนสถานะเก่าเป็นสถานะใหม่
lcd_gotoxy(8,4);
switch(Status)                      ; ตรวจสอบสถานะควบคุม
{
                                    ; สถานะเริ่มทำงาน
case START: printf(lcd_putc," START "); break; ; แสดงสถานะเริ่มทำงานบนจอ LCD
                                    สถานะหยุดทำงาน
case STOP: printf(lcd_putc," STOP  "); break; ; แสดงสถานะหยุดทำงานบนจอ LCD
}
}

```

```

}
}
if(!BT_ST())&&(BT_Stat==0){
    delay_ms(10);
    if(!BT_ST()){
        BT_Stat = 1;
        Status = ~ Status;
        switch(Status){
        case STOP: Step = 0;
            Motor_Stop();
            Servo_Position = Servo_Center;
            break;
        case START: Step = 1;
            break;
        }
    }
}
if(BT_ST()){
    delay_ms(10);
    if(BT_ST()){
        BT_Stat = 0;
    }
}
if(step == 1){
    Motor_On();
    Step = 2;
    Count_time1 = 0;
}
if(step == 2){
    if(sensor()==0){
        step = 1;
        Motor_Stop();
        color_avg = read_color();

```

; ตรวจสอบว่ามีการกดสวิตช์ START
หรือไม่
; หน่วงเวลาเพื่อตรวจสอบอีกครั้ง
; ตรวจสอบว่ามีการกดสวิตช์ START
หรือไม่
; กำหนดสถานะของ BT_Stat เป็น 1
; กลับสถานะ
; ตรวจสอบสถานะ
; สถานะหยุดทำงาน
; สั่งให้มอเตอร์หยุดทำงาน
; สั่งให้เซอร์โวอยู่ตำแหน่งกลาง
; สถานะทำงาน
; ถ้าไม่มีการกดสวิตช์ START
; หน่วงเวลา
; ถ้าไม่มีการกดสวิตช์ START
; เคลียร์ BT_Stat เป็น 0
; ถ้าสแต็ปการทำงานเป็น 1
; สั่งให้มอเตอร์ทำงาน
; ถ้าสแต็ปการทำงานเป็น 2
; กำหนดให้สแต็ปเป็น 1
; สั่งให้มอเตอร์หยุดทำงาน
; อ่านค่าของสีมะนาว

```

if(color_avg < Ref_Color){
Servo_Position = Servo_Left;
YLemon++;
if(mode == MODE_CNT_YLEMON){
if(YLemon == Num_YLemon){

YLemon = 0;
Status = STOP;
Motor_Stop();
step = 0;
}
}
}
else{ // Green lemon
Servo_Position = Servo_Right;
GLemon++;
if(mode == MODE_CNT_GLEMON){
if(GLemon == Num_GLemon){

GLemon = 0;
Status = STOP;
Motor_Stop();
step = 0;
}
}
}
Servo_Position = Servo_Right;
delay_ms(500); // หน่วงเวลา
Servo_Position = Servo_Center;
delay_ms(500);
}
if(count_time1 >= T15SEC){
Status = STOP;
Motor_Stop();

```

```

; ถ้ามะนาวเป็นสีเหลือง
; สั่งให้เซอร์โวไปทางซ้าย
; เพิ่มจำนวนมะนาวสีเหลืองอีกหนึ่ง
; ถ้าเป็นโหมดนับจำนวนลูกมะนาว
; ถ้าจำนวนมะนาวสีเหลืองเท่ากับจำนวน
มะนาวที่ตั้งไว้
; เคลียร์จำนวนมะนาวสีเหลือง
; สถานะหยุดทำงาน
; สั่งให้มอเตอร์หยุดทำงาน
; สั่งให้สแต็ปเป็น 0

; สั่งให้มอเตอร์ไปทางขวา
; เพิ่มมะนาวสีเขียว
; ถ้าเป็นโหมดนับจำนวนลูกมะนาว
; ถ้าจำนวนมะนาวสีเขียวเท่ากับจำนวน
มะนาวที่ตั้งไว้
; เคลียร์จำนวนมะนาวสีเขียว
; สถานะหยุดทำงาน
; สั่งให้มอเตอร์หยุดทำงาน
; สั่งให้สแต็ปเป็น 0

; สั่งเซอร์โวไปทางขวา

; สั่งเซอร์โวไปตำแหน่งตรงกลาง
; หน่วงเวลา

; ถ้าเวลาครบ 3 วินาที
; ให้สถานะเป็นหยุดการทำงาน
; สั่งให้มอเตอร์หยุดทำงาน

```

```

step = 0; ; เปลี่ยนสแต็ปการทำงานเป็น 0
}
}
}

/*****/
/* Function : Setting_Machine()
*/
/*****/

void Setting_Machine(void)
{
if(!IBT_OK())&&!IBT_CL()){ ; ถ้ามีการกดปุ่ม ตกลง และปุ่ม ยกเลิก
if(count_time >= T3SEC){ ; ถ้าเลาครบ 3 วินาที

Motor_Stop(); ; สั่งให้มอเตอร์หยุดการทำงาน
; แสดงโหมดตั้งค่าต่างๆ

lcd_gotoxy(1,1);printf(lcd_putc," Select Mode ");
lcd_gotoxy(1,2);printf(lcd_putc,"Press + : Setup ");
lcd_gotoxy(1,3);printf(lcd_putc,"Press - : Calibrate ");
lcd_gotoxy(1,4);printf(lcd_putc,"Press C : Exit ");

; หากมีการกดปุ่มให้รอจนกว่าไม่กดปุ่ม
while(!IBT_OK())||(!IBT_CL())||(!IBT_UP())||(!IBT_DW()); delay_us(10);
; หากมีการกดปุ่ม ยกเลิก ปุ่มเพิ่ม ปุ่มลด
while(! IBT_CL())&&( BT_UP())&&( BT_DW()); delay_us(10);
if(!IBT_UP()){ ; ถ้ากดปุ่ม เพิ่ม
Setup(); ; เรียกโหมด Setup()
}
if(!IBT_DW()){ ; ถ้ากดปุ่มลด
Calibrate(); ; เรียกโหมดตั้งค่าสี
}
Status = STOP; ; กำหนดสถานะเป็นหยุดทำงาน
Status_Old =~ Status; ; กำหนดสถานะเก่าเป็นสถานะตรงกันข้าม
กับสถานะใหม่
GLemon_Old = -1; ; กำหนดจำนวนมะนาวสีเขียวเก่ามีจำนวน
-1

```

```

YLemon_Old = -1; ; กำหนดจำนวนมะนาวสีเหลืองเก่ามี
                                     จำนวน -1
Show_Menu(); ; เรียกโชว์เมนู
}
}
else
{
count_time = 0; ; เคลียร์เวลา
}
}
/*****/
/* Function : Setup()
   */
/*****/
void Setup(void) ; ฟังก์ชัน Setup()
{
int1 Stat = 1; ; ตัวแปรเก็บสถานะว่าจะออกจากโหมด
                                     setup หรือไม่

int index = 0;
int mode_buf;
unsigned long lemon_buf;
unsigned long lemon_old;
unsigned long Time_BT_UP = 0;
unsigned long Time_BT_DW = 0;

lemon_old = -1;
mode_buf = mode; ; เอาโหมดปัจจุบัน มาเก็บให้โหมดสำรอง
                                     ; แสดงโหมด Setup บนจอ LCD

lcd_gotoxy(1,1);printf(lcd_putc," Select Lemon ");
lcd_gotoxy(1,2);printf(lcd_putc," Mode : ");
lcd_gotoxy(1,3);printf(lcd_putc," Number: - ");
lcd_gotoxy(1,4);printf(lcd_putc," ");
while(Stat){
switch(index){
case 0: lcd_gotoxy(1,2);printf(lcd_putc,">");

```

```

; แสดงโหมดนับลูกมะนาว
switch(mode_buf){
case MODE_NOCNT_LEMON:lcd_gotoxy(9,2);printf(lcd_putc," OFF ");break;
case MODE_CNT_GLEMON:lcd_gotoxy(9,2);printf(lcd_putc," Green ");break;
case MODE_CNT_YLEMON:lcd_gotoxy(9,2);printf(lcd_putc," Yellow");break;
}
while(!IBT_OK())||(!IBT_CL())||(!IBT_UP())||(!IBT_DW()); delay_us(10);
while(( BT_OK())&&( BT_CL())&&( BT_UP())&&( BT_DW())); delay_us(10);
if(!BT_OK()){
; ถ้ามีการกดปุ่ม ตกลง
switch(mode_buf){
; โหมดไม่นับลูกมะนาว
case MODE_NOCNT_LEMON: Stat = 0;
break;
; โหมดนับลูกมะนาวสีเขียว
case MODE_CNT_GLEMON: lemon_buf = Num_GLemon; index = 1;
lcd_gotoxy(1,2);printf(lcd_putc," ");
lcd_gotoxy(1,3);printf(lcd_putc,">");
while(!BT_OK()); delay_us(10);
break;
; โหมดนับลูกมะนาวสีเหลือง
case MODE_CNT_YLEMON: lemon_buf = Num_YLemon; index = 1;
lcd_gotoxy(1,2);printf(lcd_putc," ");
lcd_gotoxy(1,3);printf(lcd_putc,">");
while(!BT_OK()); delay_us(10);
break;
}
}
if(!BT_CL()){
; ถ้ากดปุ่มยกเลิก
Stat = 0;
; ออกจากโหมด
}
if(!BT_UP()){
; ถ้ามีการกดปุ่มตกลง
if(mode_buf<=1){
; ถ้าโหมด น้อยกว่าหรือเท่ากับ 1
mode_buf++;
; ให้mode_buf++
}
}

```

```

else{
mode_buf=0; ; ให้mode_buf=0
}
}
if(!BT_DW()){ ; ถ้ากดปุ่มยกเลิก
if(mode_buf>0){ ; ถ้าโหมด มากกว่า 0
mode_buf--; ; ให้ mode_buf--
}
else{
mode_buf=2; ; ให้ mode_buf=2
}
}
break;
case 1: if(lemon_buf != lemon_old){ ; มะนาวเก่าไม่เท่ากับมะนาวใหม่
lemon_old = lemon_buf; ; โหลดจำนวนมะนาวไปยังมะนาวเก่า
lcd_gotoxy(10,3);printf(lcd_putc,"%lu ",lemon_buf);
}
if(!BT_OK()){ ; ถ้ากดปุ่มตกลง
mode = mode_buf; ; เอาค่าโหมดใหม่ไปยังตัวแปรโหมด
Record_Mode(mode); ; บันทึกโหมด
; เซ็คโหมด

switch(mode){
; โหมดนับมะนาวสีเขียว
case MODE_CNT_GLEMON: Num_GLemon = lemon_buf; ; โหลดจำนวนมะนาวสีเขียวใหม่
; ไปยังตัวแปร Num_GLemon
Record_Num_GLemon(Num_GLemon); ; บันทึกจำนวนมะนาวลงหน่วยความจำ
break;
; โหมดนับมะนาวสีเหลือง
case MODE_CNT_YLEMON: Num_YLemon = lemon_buf; ; โหลดจำนวนมะนาวสีเหลือง
; ใหม่ไปยังตัวแปร Num_YLemon
Record_Num_YLemon(Num_YLemon); ; บันทึกจำนวนมะนาวลงหน่วยความจำ
break;
}
Stat = 0; ; ออกจากโหมดการทำงาน
}

```

```

if(!BT_CL()){
Stat = 0;
}
; ถ้ากดปุ่ม ยกเลิก
; ออกจากโหมดการทำงาน

if(!BT_UP()){
if(Time_BT_UP == 30000){
if(!BT_UP()){
if(lemon_buf < 1000){
lemon_buf++;
delay_ms(50);
}
else{
lemon_buf=0;
}
}
}
; ถ้ากดปุ่ม ตกลง
; ถ้าเวลาที่กด มากกว่า 30000 ลูป
; ถ้ากดปุ่ม ตกลง
; ถ้า lemon_buf น้อยกว่า 1000
; ให้ lemon_buf++

else{
lemon_buf=0;
}
}
; ให้ lemon_buf = 0

else{
Time_BT_UP++;
}
}

if(BT_UP()){
if(Time_BT_UP>1000){
if(lemon_buf < 1000){
lemon_buf++;
}
else{
lemon_buf=0;
}
}
}
; ถ้าเวลาที่กด มากกว่า 1000 ลูป
; ถ้า lemon_buf น้อยกว่า 1000
; ให้ lemon_buf++
; ให้ lemon_buf=0

Time_BT_UP = 0;
}

if(!BT_DW()){
if(Time_BT_DW == 30000){
if(!BT_DW()){
if(lemon_buf > 0){
}
}
}
; ถ้ามีการกดปุ่มยกเลิก
; ถ้าเวลาที่กด มากกว่า 30000 ลูป
; ถ้ามีการกดปุ่มยกเลิก
; ถ้า lemon_buf มากกว่า 0

```



```

int1 Stat = 1;
unsigned long value;

lcd_gotoxy(1,1);printf(lcd_putc,"          ");
lcd_gotoxy(1,2);printf(lcd_putc," Please put a lemon ");
lcd_gotoxy(1,3);printf(lcd_putc," into tray!  ");
lcd_gotoxy(1,4);printf(lcd_putc,"          ");
Motor_On();

// Read sensor
while(Sensor()==1);
Motor_Stop();

lcd_gotoxy(1,1);printf(lcd_putc," Please select color");
lcd_gotoxy(1,2);printf(lcd_putc,"   of lemon.  ");
lcd_gotoxy(1,3);printf(lcd_putc," color :   -   ");
lcd_gotoxy(1,4);printf(lcd_putc,"Up:Green  Dw:Yellow");
while(Stat){
  if(!BT_UP()){
    delay_us(100);
    if(!BT_UP()){
      lcd_gotoxy(10,3);printf(lcd_putc,"Green ");
      sel_color = 1;
      while(!BT_UP());delay_us(100);
    }
  }
  if(!BT_DW()){
    delay_us(100);
    if(!BT_DW()){
      lcd_gotoxy(10,3);printf(lcd_putc,"Yellow");

```

ต้องการตั้งค่าสี
 ; ตัวแปร Stat สำหรับเช็คว่ามีอาการออกจาก
 โหมดตั้งค่าสีหรือไม่
 ; ตัวแปรเก็บค่าของสี
 ; แสดงข้อความว่าให้ใส่ลูกมะนาวลงในถาด
 ; สั่งให้มอเตอร์หมุนเพื่อให้มะนาวหล่นลง
 มาที่ช่องอ่านสีมะนาว
 ; สั่งให้มอเตอร์หยุดทำงาน
 ; แสดงข้อความโหมดตั้งค่าสีมะนาว
 ; เช็คว่ามีอาการออกจากโหมดนี้หรือไม่
 ; ถ้ากดปุ่มตกลง
 ; หน่วงเวลา
 ; ถ้ากดปุ่มตกลง
 ; แสดงข้อความบอกว่าเลือกมะนาวสีเขียว
 ในการปรับค่า
 ; ให้ sel_color เป็นการเลือกมะนาวสีเขียว
 ; ถ้ามีการกดปุ่มเพิ่ม
 ; ถ้ามีการกดปุ่มยกเลิก
 ; หน่วงเวลา
 ; ถ้ามีการกดปุ่ม ยกเลิก
 ; แสดงข้อความบอกว่าเลือกมะนาวสี
 เหลืองในการปรับค่า

```

sel_color = 2;
เหลืออง
while(!BT_DW());delay_us(100);
}
}
if(!BT_OK()){
delay_us(100);
if(!BT_OK()){
if(sel_color>0){
// Read color
value = read_color();
switch(sel_color){

case 1: Ref_Color = value-20; break;

case 2: Ref_Color = value+20; break;

}
Record_REF_Color(Ref_Color);
stat = 0;
}
while(!BT_OK());delay_us(100);
}
}
if(!BT_CL()){
delay_us(100);
if(!BT_CL()){
stat = 0;
while(!BT_OK());delay_us(100);
}
}
}
Servo_Position = Servo_Right;
delay_ms(500);
Servo_Position = Servo_Center;

```

; ให้ sel_colorเป็นการเลือกมะนาวสี
; รอจนกว่าจะปล่อยปุ่มลด
; ถ้ามีการกดปุ่มตกลง
; หน่วงเวลา
; ถ้ามีการกดปุ่ม ตกลง
; อ่านค่าของสีมะนาว
; ตรวจสอบสีของมะนาวที่ใช้ในการปรับ
ค่า
; ถ้ามะนาวสีเขียว ให้ค่าอ้างอิงสีเท่ากับ
Ref_Color = value-20
; ถ้ามะนาวสีเหลือง ให้ค่าอ้างอิงสีเท่ากับ
Ref_Color = value-20
; บันทึกสีลงหน่วยความจำ
; กำหนดให้ตัวแปร Stat = 0
; รอจนกว่าจะปล่อยปุ่ม ตกลง
; ถ้ามีการกดปุ่มยกเลิก
; หน่วงเวลา
; ถ้ามีการกดปุ่มยกเลิก
; กำหนดให้ตัวแปร Stat = 0
; รอจนกว่าจะปล่อยปุ่ม ตกลง
; สั่งเซอร์โวไปทางขวา
; หน่วงเวลา
; สั่งเซอร์โวไปตรงกลาง

```

}
/*****/
/* Function : Record_REF_Color(unsigned long color)
*/
/*****/
void Record_REF_Color(unsigned long color)
{
write_eeprom(1,color/1000);           ; เก็บค่าสีหลัก 1000
write_eeprom(2,(color%1000)/100);    ; เก็บค่าสีหลัก 100
write_eeprom(3,((color%1000)%100)/10); ; เก็บค่าสีหลัก 10
write_eeprom(4,((color%1000)%100)%10); ; เก็บค่าสีหลัก 1
}
/*****/
/* Function : unsigned long Read_REF_Color()
*/
/*****/
unsigned long Read_REF_Color(void)
{
unsigned long color;

color = (unsigned long)read_eeprom(1)*1000; ; อ่านค่าสีหลัก 1000
color += (unsigned long)read_eeprom(2)*100; ; อ่านค่าสีหลัก 100
color += (unsigned long)read_eeprom(3)*10; ; อ่านค่าสีหลัก 10
color += (unsigned long)read_eeprom(4); ; อ่านค่าสีหลัก 1
return(color);
}

/*****/
/* Function : Record_Num_GLemon(unsigned long num)
*/
/*****/
void Record_Num_GLemon(unsigned long num)
{
write_eeprom(5,num/1000);           ; เก็บมะนาวสีเขียวหลัก 1000
write_eeprom(6,(num%1000)/100);    ; เก็บมะนาวสีเขียวหลัก 100

```

```

write_eeprom(7,((num%1000)%100)/10);           ; เก็บมะนาวสีเขียวหลัก 10
write_eeprom(8,((num%1000)%100)%10);         ; เก็บมะนาวสีเขียวหลัก 1
}
/*****/
/* Function : unsigned long Read_Num_GLemon()
*/
/*****/
unsigned long Read_Num_GLemon(void)
{
unsigned long num;
num = (unsigned long)read_eeprom(5)*1000;     ; อ่านมะนาวสีเขียวหลัก 1000
num += (unsigned long)read_eeprom(6)*100;    ; อ่านมะนาวสีเขียวหลัก 100
num += (unsigned long)read_eeprom(7)*10;     ; อ่านมะนาวสีเขียวหลัก 10
num += (unsigned long)read_eeprom(8);        ; อ่านมะนาวสีเขียวหลัก 1
return(num);
}

/*****/
/* Function : Record_Num_YLemon(unsigned long num)
*/
/*****/
void Record_Num_YLemon(unsigned long num)
{
write_eeprom(9,num/1000);                     ; เก็บมะนาวสีเหลืองหลัก 1000
write_eeprom(10,(num%1000)/100);             ; เก็บมะนาวสีเหลืองหลัก 100
write_eeprom(11,((num%1000)%100)/10);       ; เก็บมะนาวสีเหลืองหลัก 10
write_eeprom(12,((num%1000)%100)%10);       ; เก็บมะนาวสีเหลืองหลัก 1
}
/*****/
/* Function : unsigned long Read_Num_YLemon()
*/
/*****/
unsigned long Read_Num_YLemon(void)
{

```

```

unsigned long num;
num = (unsigned long)read_eeprom(9)*1000;           ; อ่านเมนาวสีเหลืองหลัก 1000
num += (unsigned long)read_eeprom(10)*100;         ; อ่านเมนาวสีเหลืองหลัก 100
num += (unsigned long)read_eeprom(11)*10;          ; อ่านเมนาวสีเหลืองหลัก 10
num += (unsigned long)read_eeprom(12);             ; อ่านเมนาวสีเหลืองหลัก 1
return(num);
}
/*****/
/* Function : unsigned long read_color()
*/
/*****/
unsigned long read_color(void)
{
unsigned long avg_color;
// Select sensor 1
Select_Sensor1();           ; เลือกเซนเซอร์ตัวที่ 1
delay_us(15);              ; หน่วงเวลา
avg_color = Read_Sensor();  ; อ่านค่าสี
// Select sensor 2
Select_Sensor2();           ; เลือกเซนเซอร์ตัวที่ 2
delay_us(15);              ; หน่วงเวลา
avg_color += Read_Sensor(); ; อ่านค่าสี
// Select sensor 3
Select_Sensor3();           ; เลือกเซนเซอร์ตัวที่ 3
delay_us(15);              ; หน่วงเวลา
avg_color += Read_Sensor(); ; อ่านค่าสี
// Select sensor 4
Select_Sensor4();           ; เลือกเซนเซอร์ตัวที่ 4
delay_us(15);              ; หน่วงเวลา
avg_color += Read_Sensor(); ; อ่านค่าสี
// Average color
avg_color = avg_color/4;   ; หาค่าเฉลี่ยสี
return(avg_color);        ; ส่งค่าเฉลี่ยสี
}

```

ภาคผนวก ข คู่มือการใช้งาน

คู่มือการใช้งาน

เครื่องนี้เป็นอุปกรณ์ที่ช่วยเพิ่มประสิทธิภาพในการคัดแยกสีและนับจำนวนมะนาวโดยจะทำการคัดสีโดยเปรียบเทียบค่าของสีมะนาวและนำไปแสดงผลในจอแอลซีดี การใช้งานสามารถนำไปคัดแยกมะนาวที่กำลังจะสุกหรือสุกแล้วเพื่อคัดออกจากมะนาวสีเขียวเพื่อที่จะเก็บมะนาวได้นานขึ้น เป็นต้น

ขั้นตอนการใช้งาน

- 1) เสียบปลั๊กและเปิดสวิตช์
- 2) ตั้งโปรแกรมโดยกดปุ่ม Ok และ Cancel พร้อมกัน 3 วินาที เพื่อเข้าสู่การตั้งโปรแกรม
- 3) เลือกฟังก์ชันการตั้งค่าสีและจำนวนมะนาว โดยกดปุ่ม Up และ Down
- 4) เมื่อได้ฟังก์ชันที่ต้องการแล้วกดปุ่ม Ok
- 5) กดปุ่ม Start เพื่อเริ่มทำงาน และปุ่ม Stop เพื่อหยุดการทำงาน

