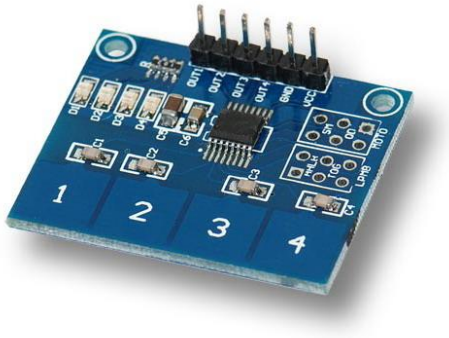
Touch Switch



รายละเอียด 4-way Switch Digital Touch Capacitive Module

สวิตช์สัมผัสแบบ Capacitive 4 ช่อง มีเลข 1-4 ของแต่ละช่อง ใช้งานง่ายเมื่อเอานิ้วไปสัมผัส ก็จะมีเอาต์พุตออกที่ช่อง OUT1-OUT4 ตามช่องที่กด พร้อมไฟ LED แสดงเอาต์พุต นำไปควบคุมอุปกรณ์ไฟฟ้าต่าง ๆ หรือส่งเป็นสวิตช์ให้กับ Arduino ก็ได้

รายละเอียด 4-way Switch Digital Touch Capacitive Module

This Capacitive Touch Module uses the touch-sensing IC TTP224 to add capacitive touch to your project . Just power with 2.4 to 5.5VDC and touch the pad to activate the sensor. These touch switches interface easily to any project - with or without a microcontroller.

When a capacitive load (such as a human hand) is in close proximity to the sense-pad, the sensor detects the change in capacitance and activates the switch. Custom sense-pads can be made from nearly any conductive material and these sensors can detect touch through thin layers of non-conductive materials such as glass, plastic, fabric or even wood. So, you can make this sensor hidden in the wall, tabletop and other places, which allows you to remove the conventional push-buttons troubles.

There are totally 4 pads on this module, these sense-pads can be extended with wire and almost any conductive material, so that you can add 4 touch switch easily for your project.

Specification

- Onboard TTP224 capacitive touch sensor IC 4 keys
- 4 board-level status indicator
- Working voltage: 2.4V-5.5V

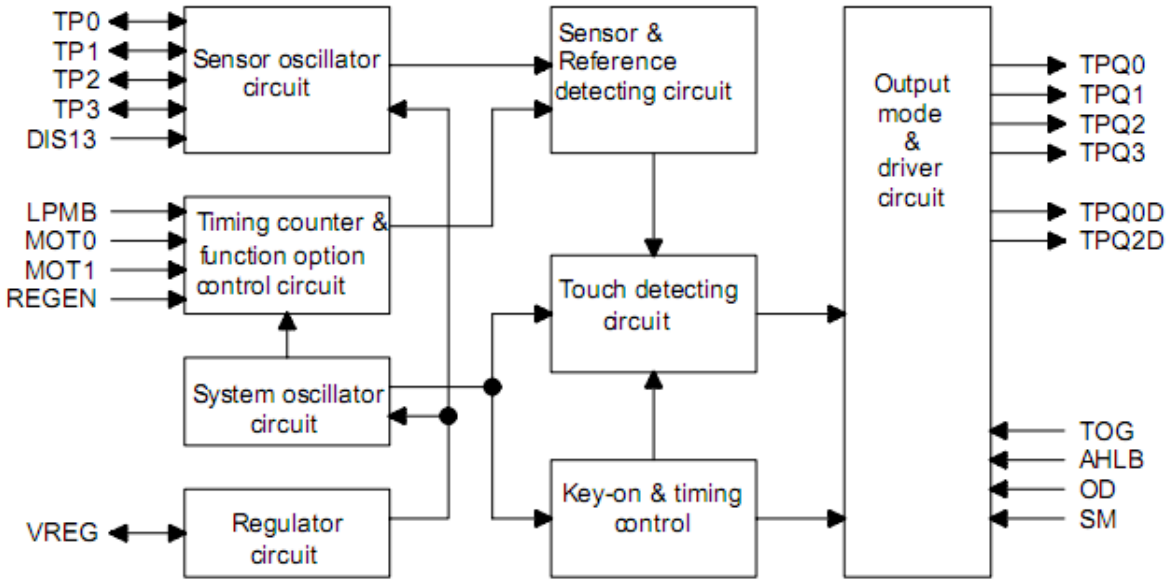
- Advanced Pins breakout, so that the output mode and working mode selectable.
- Low power selection
- PCB board size: 35 (mm) x29 (mm)

TTP224

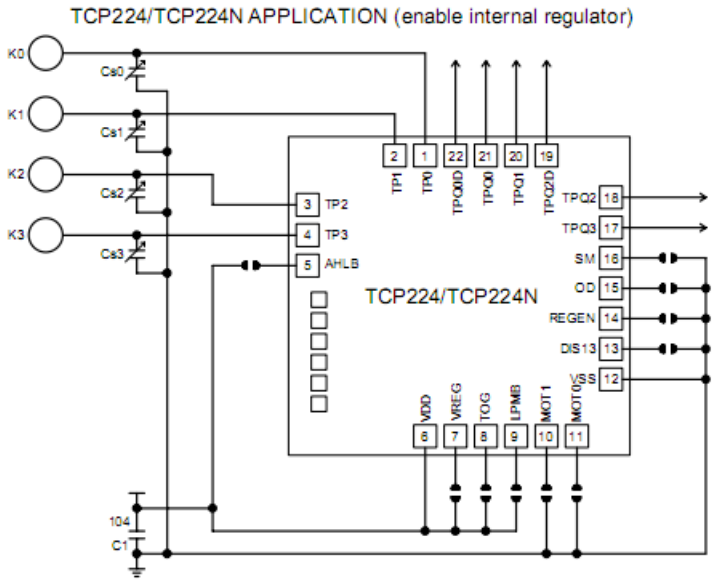
FEATURES

- Operating voltage 2.4V~5.5V
- Built-in regulator with external enable/disable option
- Operating current, @VDD=3V no load
  At low power mode typical 2.5uA
  At fast mode typical 9.0uA
- @VDD=3V operating voltage :
  The response time about 100mS at fast mode, 200mS at low power mode for TTP224
  The response time about 60mS at fast mode, 160mS at low power mode for TTP224N
- ***Sensitivity can adjust by the capacitance(0~50pF)*** outside for each touch pad
- Provides Fast mode and Low Power mode selection by pad option(LPMB pin)
- Provides direct mode or toggle mode、CMOS output or open drain ***output active high or active low by pad option(TOG/OD/AHLB pin).***
- Provides 2 output pins TPQ0D, TPQ2D that have no diode protection, active low
- Have the maximum on time 120sec/64sec/16sec/infinite by pad option(MOT1, MOT0 pin)
- ***After power-on have about 0.5sec stable-time, during the time do not touch the key pad, and the function is disabled***
- Auto calibration for life, and the re-calibration period is about 4.0sec, when key has not be touched.

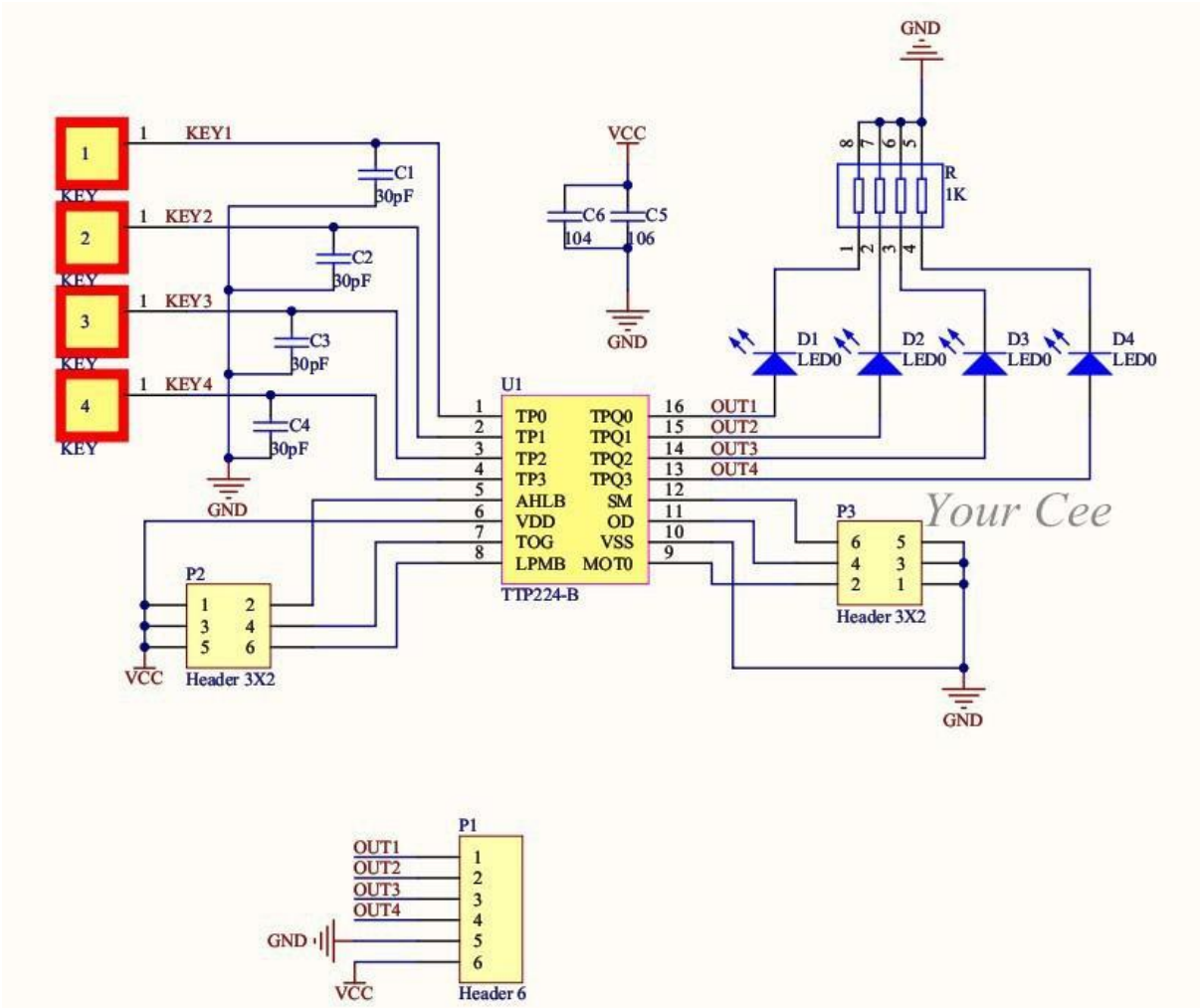
Block Diagram



APPLICATION CIRCUIT


TCP224/TCP224N APPLICATION (enable internal regulator)
K0
K1
K2
K3
Cs0
Cs1
Cs2
Cs3
TP2
TP3
AHLB
TCP224/TCP224N
TPQ2
TPQ3
SM
OD
REGEN
DIS13
VSS
104
C1

วงจร


GND
VCC
KEY1
KEY2
KEY3
KEY4
KEY
C1
C2
C3
C4
30pF
C6
C5
104
106
R
1K
D1
D2
D3
D4
LED0
U1
TP0
TP1
TP2
TP3
AHLB
VDD
TOG
LPMB
TPQ0
TPQ1
TPQ2
TPQ3
SM
OD
VSS
MOT0
OUT1
OUT2
OUT3
OUT4
TTP224-B
P2
P3
Header 3X2
P1
Header 6
Your Cee

Description

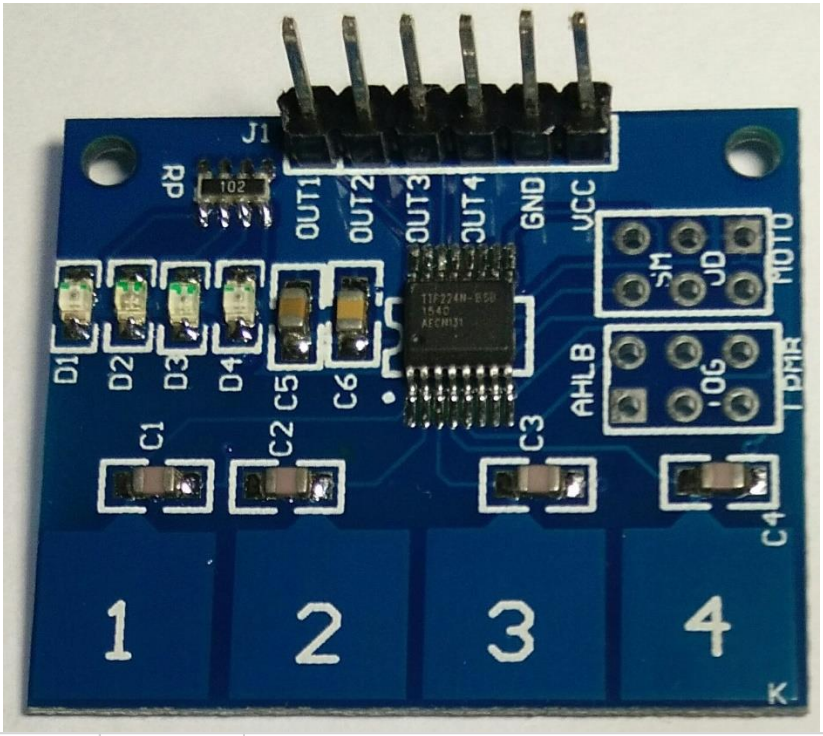
1. ***working voltage: 2.4V-5.5V***

2. 4 board-level status indicator

3. PCB board size: 35 (mm) x29 (mm)

4. TTP224 capacitive touch sensor IC 4 keys

5. the module can set the output mode, the key output mode, the maximum output time and fast / low power selection

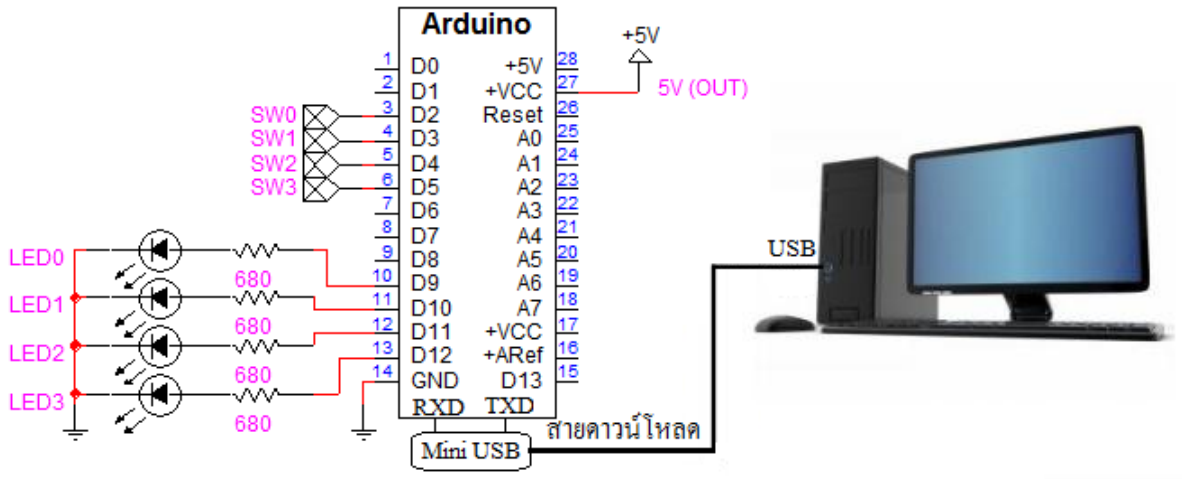


Output mode:			
TOG	OD	AHLB	Pad TPQ0~TP3 option features
open	open	open	Direct mode, CMOS active high output
open	open	VDD	Direct mode, CMOS active low output
open	VSS	open	Direct mode, Open drain active high output
open	VSS	VDD	Direct mode, Open drain active low output
VDD	open	open	Toggle mode, COMS output, Power on state=0
VDD	open	VDD	Toggle mode, COMS output, Power on state=1
VDD	VSS	open	Toggle mode, Power on state high-Z, Active high
VDD	VSS	VDD	Toggle mode, Power on state high-Z, Active low

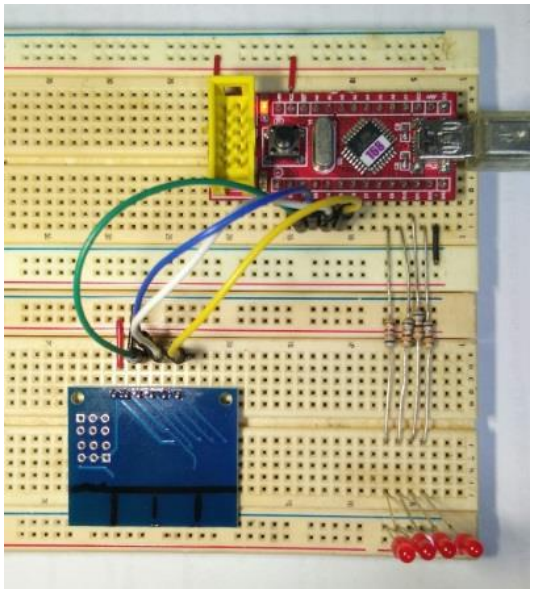
TOG	Pad TPQ0D,TPQ2D (has no diode protection) option features
open	VDD Toggle mode, Open drain active low output, Power on state high-Z
VDD	open Direct mode, Open drain active low output, Power on state high-Z

Key operation mode:		
SM	Option features	
open	Multi-key mode	
VSS	Single key mode	
Maximum key on duration time:		
MOT1	MOT0	Option features
VSS	VSS	Maximum on time 120sec
VSS	open	Maximum on time 64sec
open	VSS	Maximum on time 16sec
open	open	Infinite(Disable maximum on time)
Fast and Low power mode:		
LPMB	Option features	
VDD	Fast mode	
open	Low Power mode	
Input keys number select:		
DIS13	Option features	
open	Enable TP1, TP3	
VSS	Disable TP1, TP3	

วงจรทดลอง



หมายเหตุ ขา D0 กับ ขา RxD เป็นขาเดียวกัน และขา D1 ก็เป็นขาเดียวกับ TxD



ตัวอย่างที่ 1 ทดสอบสวิทช์ แตะสวิทช์ตัวไหนให้ส่งตำแหน่งของสวิทช์นั้นออกทางพอร์ทอนุกรม

```
int i;

void setup()
{
//set pins 2 - 5 as inputs
for(i=2; i<=5; i++)
  {
    pinMode(i, INPUT);
  }
  Serial.begin(9600);
}

void loop(){
  for(i=2; i<=5; i++)
  {
    if(digitalRead(i) == HIGH){
      Serial.print("Press Touch SW ");
      Serial.println(i-2);
    }
  }
  delay(100);
}
```

ตัวอย่างที่ 2 ทดลองการตรวจสอบสวิทช์แบบมีช่วงเวลา ถ้าแตะสวิทช์เป็นเวลา 100 ms – 300 ms ถือว่าเป็นการกดสวิทช์ (SW ON) ถ้าไม่ใช่ถือว่าไม่ได้กดสวิทช์ (SW OFF) โดยใช้ฟังก์ชั่น Timer interrupt และแสดงผลออกทางพอร์ทอนุกรม

ในตัวอย่างนี้มีการใช้ Libraries Timer1

**ดูการติดตั้ง Timer 1 ได้จาก https://sites.google.com/site/eplearn/arduino-project/03-led-8x8 **

```
#include <TimerOne.h>
#define on true
#define off false
#define time_100ms 10   // = 100ms/(time interval 10ms)
#define time_300ms 30   // = 300ms/(time interval 10ms)
unsigned char sw_time;
bool sw_flag = off;

int i,j;

void setup() {
  Timer1.initialize(10000); // set a timer of length 10000 us = 10ms
  Timer1.attachInterrupt( timerIsr ); // attach the service routine here
  pinMode(i, INPUT);
  Serial.begin(9600);
  Serial.println("SW ON/OFF Test");
  sw_time = 0;
}
void loop() {
  if(sw_flag){
    Serial.println("SW ON");
    sw_flag = off;
  }
}

/// --------------------------
/// Scan key
/// --------------------------
void timerIsr()
{
  if(digitalRead(2) == HIGH) {
    ++sw_time;
    if(sw_time >200) sw_time = 210;
  }else{
    if((sw_time>=10)&&(sw_time<=30)) sw_flag = on;
    else sw_flag = off;
    sw_time =0;
  }
}
```

ตัวอย่างที่ 3 ทดลองการตรวจสอบสวิทช์ 1 ตัว แต่ทำงานได้ 2 หน้าที่ โดยถ้า ถ้าแตะสวิทช์เป็นเวลา 100 ms – 300 ms ถือว่าเป็นการกดสวิทช์ (SW ON) ถ้าแตะสวิทช์นานกว่า 1 วินาที ถือว่า Power sw ON โดยใช้ฟังก์ชั่น Timer interrupt และแสดงผลออกทางพอร์ทอนุกรม

```
#include <TimerOne.h>
#define on true
#define off false
#define t_100ms 10   // = 100ms/(time interval 10ms)
#define t_300ms 30   // = 300ms/(time interval 10ms)
#define t_1s 100   // = 2000ms/(time interval 10ms)
unsigned char sw_time;

bool sw_flag = off;
bool power_flag = off;
```

```
int i,j;

void setup() {
  Timer1.initialize(10000); // set a timer of length 10000 us = 10ms
  Timer1.attachInterrupt( timerIsr ); // attach the service routine here
  pinMode(i, INPUT);
  Serial.begin(9600);
  Serial.println("SW ON/OFF Test");
  sw_time = 0;
}
void loop() {
  if(sw_flag)
  {
    Serial.println("SW ON");
    sw_flag = off;
  }
  else if(power_flag)
  {
    Serial.println("Power ON");
    power_flag = off;
    sw_time = 0;
   }
}

/// -------------------------
/// Scan key
/// -------------------------
void timerIsr()
{
  if(digitalRead(2) == HIGH) {
    ++sw_time;
    if(sw_time >100)
    {
      sw_time = 100;
      power_flag = on;
    }
  }else{
      if((sw_time>=t_100ms)&&(sw_time<=t_300ms)) sw_flag = on;
      else
      {
        sw_flag = off;
        power_flag = off;
      }
      sw_time =0;
  }
}
```

ตัวอย่างที่ 4 ใช้สวิทช์ SW0 ควบคุม LED 0 – LED3

```
#include <TimerOne.h>
#define sw0 2
#define sw0_ON digitalRead(sw0)== HIGH
#define on true
#define off false
bool sw0_click = off;
bool Power_sw = off;

#define time_period 10            //interrupt time's period = 10 ms
#define t_100ms 100/time_period   // = 100ms/(time interval 10ms)
#define t_300ms 300/time_period   // = 300ms/(time interval 10ms)
#define t_1s 1000/time_period     // = 2000ms/(time interval 10ms)
unsigned char sw_time;
unsigned char led_state, operation, led;

#define led0 9
#define led3 12
#define standby_mode 0
#define ready_mode 1

void setup() {
  Timer1.initialize(10000); // set a timer of length 10000 us = 10ms
  Timer1.attachInterrupt( timerIsr ); // attach the service routine here
  pinMode(sw0, INPUT);
  Serial.begin(9600);
  Serial.println("The 2_function SW Test");
  sw_time = 0;
  //set pins 9 - 12 as outputs
   for(led=led0; led<=led3; led++)
  {
    pinMode(led, OUTPUT);
  }
  pinMode(sw0, INPUT);

  led_state = 0;
  operation = standby_mode;
}
void loop() {
  switch (operation) {
    case standby_mode:
        standby_op();
      break;
    case ready_mode:
      ready_op();
      break;
  }
}

/// -------------------------
/// Scan key
/// -------------------------
void timerIsr()
{
  if(sw0_ON) {
```

```
    ++sw_time;
    if(sw_time >100)
    {
      sw_time = 100;
      Power_sw = on;
      sw0_click = off;
    }
  }else{
      if((sw_time>=t_100ms)&&(sw_time<=t_300ms))
      {
        sw0_click = on;
      }
      else
      {
        sw0_click = off;
        Power_sw = off;
      }
      sw_time =0;
  }
}

void standby_op()
{
  led_state = 0;
  display_state();
  if(Power_sw)
  {
    led_state = 1;
    display_state();
    operation = ready_mode;
    Power_sw = off;
    sw0_click = off;
    sw_time = 0;
  }
}

void ready_op()
{
  if(sw0_click)
  {
    ++led_state;
    if(led_state >4) led_state = 1;
    display_state();
    sw0_click = off;
    sw_time = 0;
  }
  if(Power_sw)
  {

    led_state = 0;
    display_state();
    Power_sw = off;
    sw0_click = off;
    sw_time = 0;
    operation = standby_mode;
  }
}
```

```
void display_state()
{
   for(led=led0; led<=led3; led++)
   {
     digitalWrite(led, LOW);
   }
   switch (led_state) {
   case 1: digitalWrite(9, HIGH); break;
   case 2: digitalWrite(10, HIGH); break;
   case 3: digitalWrite(11, HIGH); break;
   case 4: digitalWrite(12, HIGH); break;
   default: break;
  }
}
```

งานมอบหมายให้ใช้สวิทช์ 2 ตัว SW0 และ SW3 ควบคุมการทำงานของ LED 0 ถึง LED3