

# หน่วยที่ 5



รายวิชา : ภาถ่องสมองกล IPST –  
MicroBox SE

รหัสวิชา : ง22207

ครูผู้สอน : ครูยอดชาย ขุนสังวาลย์



# วันนี้เราจะเรียนอะไร



1. การประยุกต์ใช้ปุ่มสวิตช์ OK
2. การแสดงผลทางหลอด LED
3. การแสดงผลทางลำโพง Speaker



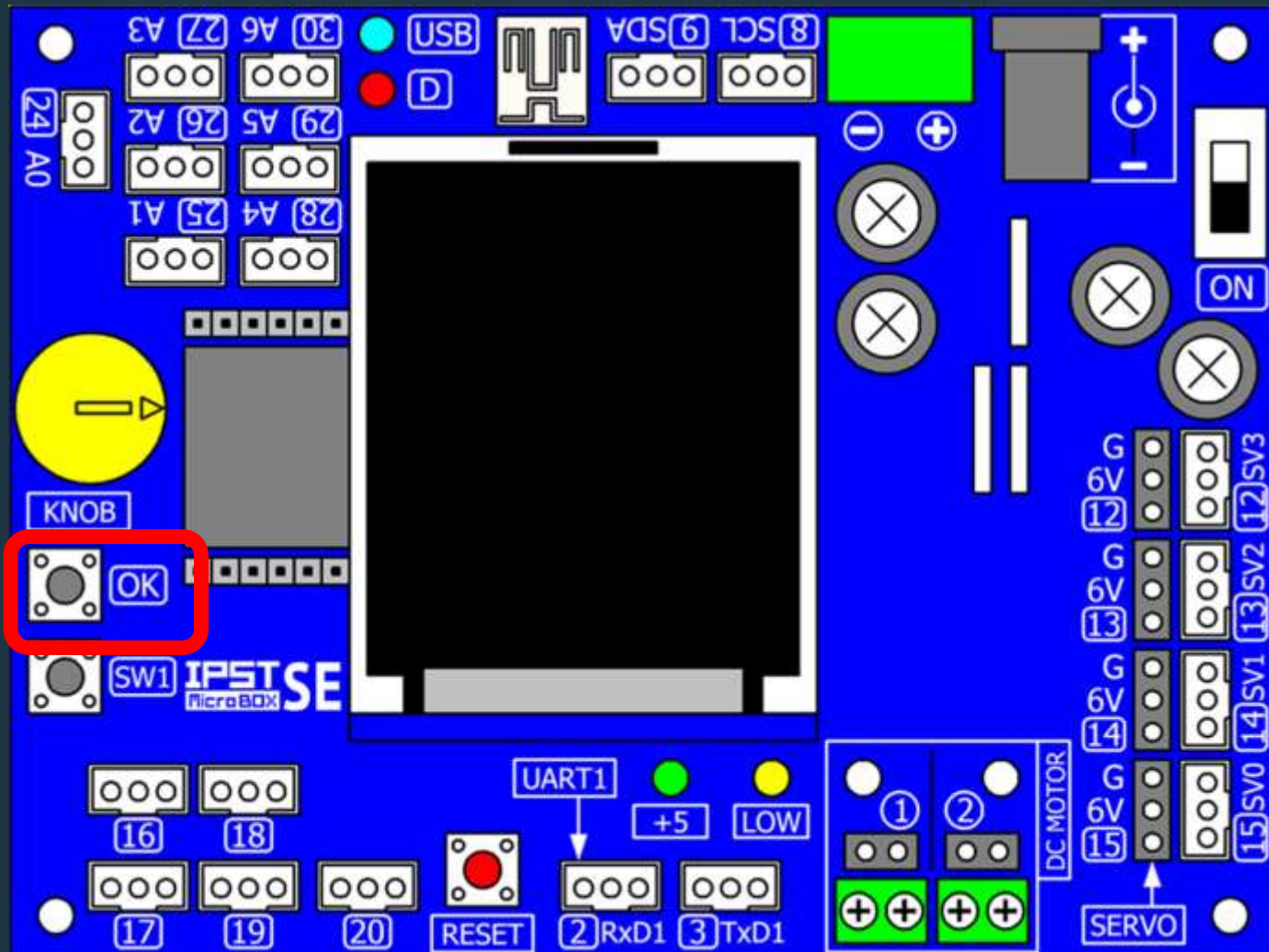
เรามาเขียนโปรแกรมเพื่อการแสดงผล  
ของ แสง และ เสียง กันเถอะ





# การประยุกต์ใช้ปุ่มสวิตช์ OK

Switch  
OK



# คำสั่งการใช้สวิตช์ OK

รูปแบบคำสั่ง

```
sw_OK_press() ;
```

sw\_OK\_press() เป็นคำสั่งรอการกดสวิตช์ OK บน IPST-MicroBOX ซึ่งจะมีสถานะดังนี้

1 (True) เมื่อกดสวิตช์

0 (False) เมื่อไม่กดสวิตช์



//โปรแกรมกดสวิตช์ OK เพื่อแสดงข้อความ

```
#include <ipst.h>
```

```
void setup ()
```

```
{  
  glcd(0,0,"Press Switch OK");
```

```
  sw_OK_press();
```

```
  glcdClear();
```

```
  glcd(0,0,"IPST-MicroBOX");
```

```
}
```

```
void loop ()
```

```
{
```

```
}
```

// คำสั่งการกดสวิตช์ OK

ผลลัพธ์ ???



//โปรแกรมกดสวิตช์ OK เพื่อแสดงข้อความกระพริบ

```
#include <ipst.h>
```

```
void setup ()
```

```
{  
  glcd(0,0,"Press Switch OK");
```

```
  sw_OK_press();
```

```
}
```

```
void loop ()
```

```
{  
  glcdClear();  
  glcd(0,0,"IPST-MicroBOX");
```

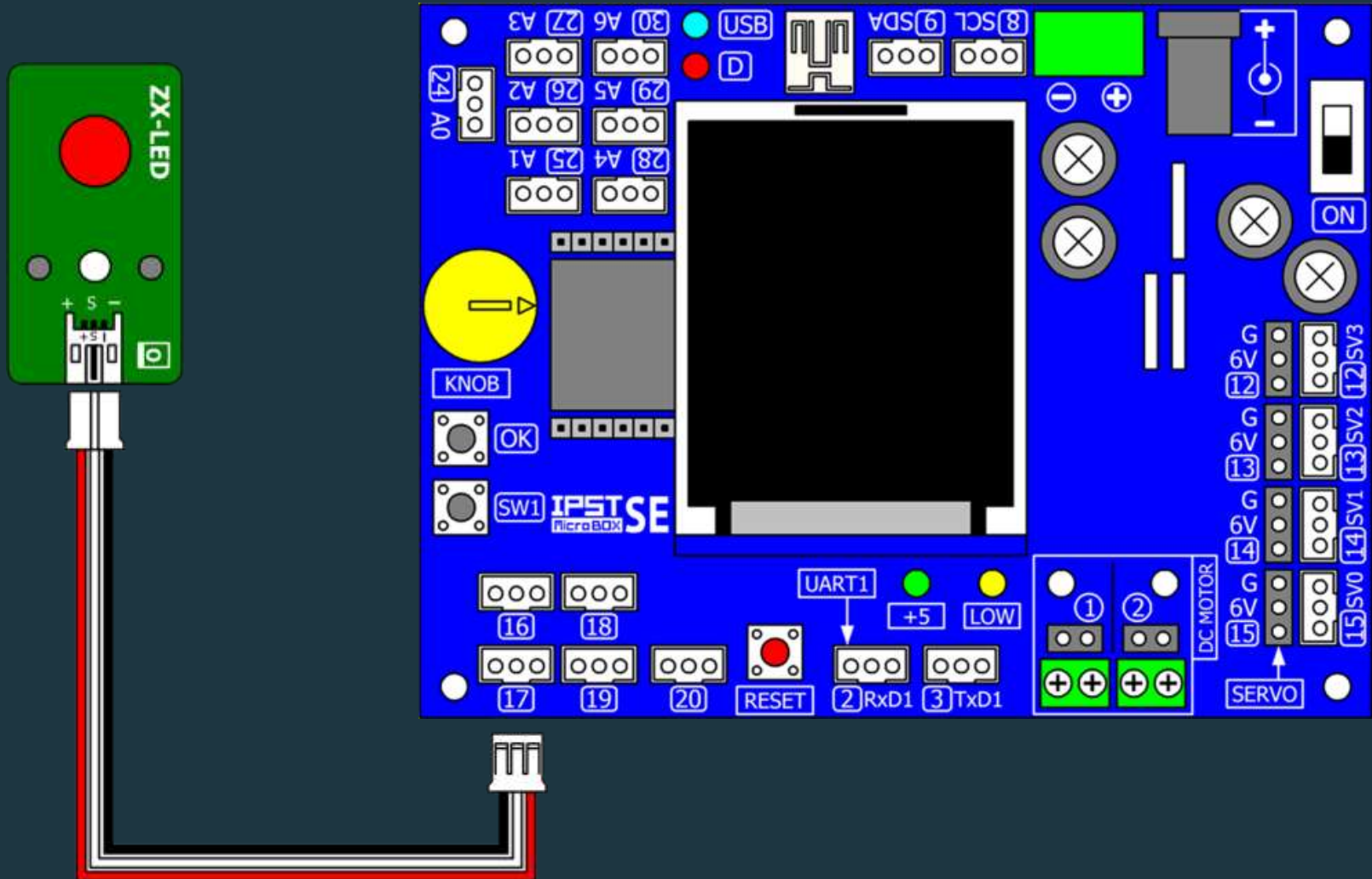
```
}
```

// คำสั่งการกดสวิตช์ OK

ผลลัพธ์ ???



# การแสดงผลทางหลอด LED





# ทางหลอด LED

รูปแบบคำสั่ง

```
out(ch,state);
```

**Ch** คือ ตำแหน่งช่องสัญญาณ

ที่เชื่อมต่อ

**state** คือ สถานะของของหลอด



```
//โปรแกรมแสดงผลทางหลอด LED 1 หลอด (เปิด)
```

```
#include <ipst.h>
```

```
void setup ()
```

```
{  
  glcd(0,0,"Press Switch OK");
```

```
  sw_OK_press();
```

```
  out(17,1);
```

```
}
```

```
void loop ()
```

```
{
```

```
}
```

// เปิดไฟหลอด LED ที่ช่อง 17

**ผลลัพธ์**

???



//โปรแกรมแสดงผลทางหลอด LED 1 หลอด (เปิด-ปิด)

```
#include <ipst.h>
```

```
void setup ()
```

```
{  
  glcd(0,0,"Press Switch OK");
```

```
  sw_OK_press();
```

```
  out(17,1);
```

```
  out(17,0);
```

```
}
```

```
void loop ()
```

```
{
```

```
}
```

// เปิดไฟหลอด LED ที่ช่อง 17

// ปิดไฟหลอด LED ที่ช่อง 17

ผลลัพธ์ ???





# โจทย์ปัญหาหาชวนคิด

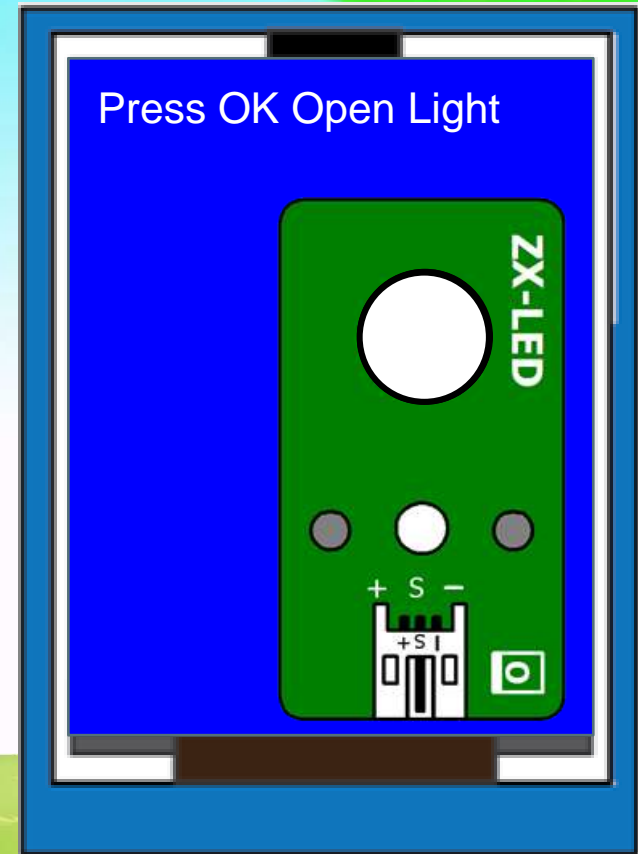
ผลลัพธ์

ให้นักเรียนเขียนโปรแกรมเพื่อทำการแสดงผล  
ทาง

หลอด LED โดยมีข้อกำหนดดังนี้

1. ให้แสดงข้อความ Press OK Open Light  
(กดสวิทช์ OK เพื่อไปสู่ขั้นตอนต่อไป)
2. หลอด LED กระพริบ 2 ครั้ง โดยกระพริบ  
ครั้งละ 1 วินาที
3. เมื่อทำงานข้อที่ 1 และ 2 เสร็จแล้วให้จบการ

ทำงาน



//โปรแกรมแสดงผลทางหลอด LED 1 หลอด (กระพริบ 2 ครั้ง)

## โปรแกรม 16

```
#include <ipst.h>
```

```
void setup ()
```

```
{  
  glcd(0,0,"Press OK Open Light ");
```

```
  sw_OK_press();
```

```
  glcdClear();
```

```
  out(17,1);
```

```
  sleep(1000);
```

```
  out(17,0);
```

```
  sleep(1000);
```

```
  out(17,1);
```

```
  sleep(1000);
```

```
  out(17,0);
```

```
  sleep(1000);
```

```
}
```

```
void loop ()
```

```
{ }
```

```
// เปิดไฟหลอด LED ที่ช่อง 17
```

```
// หน่วงเวลา 1 วินาที
```

```
// ปิดไฟหลอด LED ที่ช่อง 17
```

```
// หน่วงเวลา 1 วินาที
```

```
// เปิดไฟหลอด LED ที่ช่อง 17
```

```
// หน่วงเวลา 1 วินาที
```

```
// ปิดไฟหลอด LED ที่ช่อง 17
```

```
// หน่วงเวลา 1 วินาที
```





# โจทย์ปัญหาชวนคิด

ผลลัพธ์

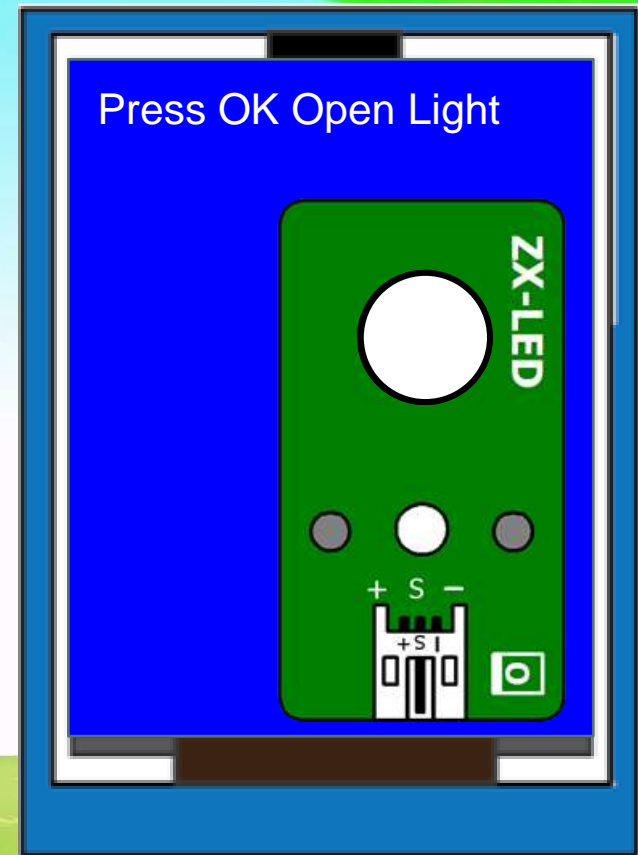
ให้นักเรียนเขียนโปรแกรมเพื่อทำการแสดงผล  
ทาง

หลอด LED โดยมีข้อกำหนดดังนี้

1. ให้แสดงข้อความ Press OK Open Light  
(กดสวิทช์ OK เพื่อไปสู่ขั้นตอนต่อไป)
2. หลอด LED **กระพริบตลอดเวลา** โดย

กระพริบ

ครั้งละ 1 วินาที



```
//โปรแกรมแสดงผลทางหลอด LED 1 หลอด (กระพริบตลอดเวลา)
#include <ipst.h>
void setup ()
{
  glcd(0,0,"Press OK Open Light ");
  sw_OK_press();
  glcdClear();
}
void loop ()
{
  out(17,1);
  sleep(1000);
  out(17,0);
  sleep(1000);
}
```

```
// เปิดไฟหลอด LED ที่ช่อง 17
// หน่วงเวลา 1 วินาที
// ปิดไฟหลอด LED ที่ช่อง 17
// หน่วงเวลา 1 วินาที
```

//โปรแกรมแสดงผลทางหลอด LED 2 หลอด (เปิดพร้อมกัน)

```
#include <ipst.h>
```

```
void setup ()
```

```
{  
  glcd(0,0,"Press Switch OK");
```

```
  sw_OK_press();
```

```
  out(17,1);
```

```
  out(18,1);
```

```
}
```

```
void loop ()
```

```
{
```

```
}
```

// เปิดไฟหลอด LED ที่ช่อง 17

// เปิดไฟหลอด LED ที่ช่อง 18

ผลลัพธ์ ???





# โจทย์ปัญหาหาชวนคิด



ผลลัพธ์

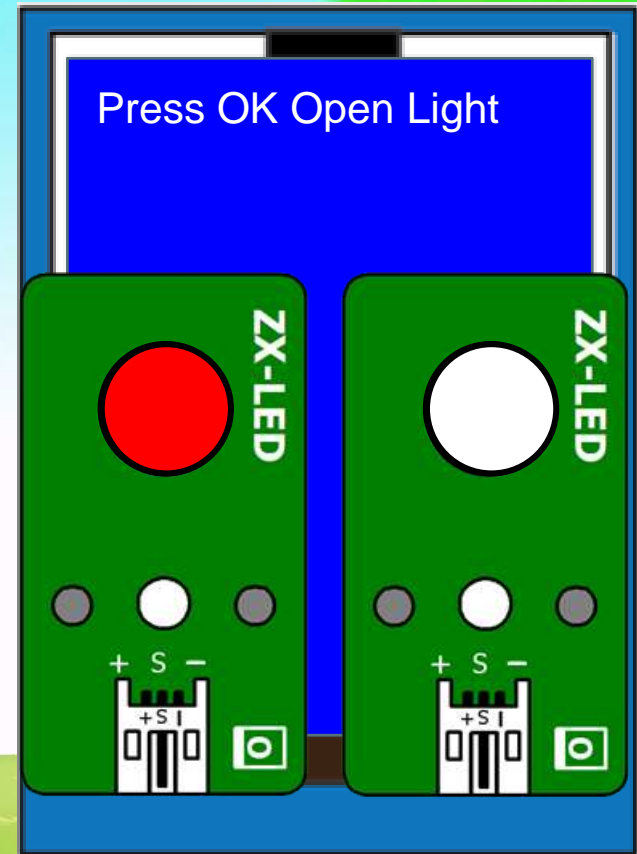
ให้นักเรียนเขียนโปรแกรมเพื่อทำการแสดงผล  
ทาง

หลอด LED โดยมีข้อกำหนดดังนี้

1. ให้แสดงข้อความ Press OK Open Light  
(กดสวิทช์ OK เพื่อไปสู่ขั้นตอนต่อไป)

2. หลอด LED จำนวน 2 หลอด กระจับ  
สลับกัน

ตลอดเวลา



```
//โปรแกรมแสดงผลทางหลอด LED 2 หลอด (กระพริบสลับกันตลอดเวลา)
```

```
#include <ipst.h>
```

```
void setup ()
```

```
{  
  glcd(0,0,"Press OK Open Light ");
```

```
  sw_OK_press();
```

```
  glcdClear();
```

```
  glcd(0,0, "Blink");
```

```
}
```

```
void loop ()
```

```
{
```

```
  out(17,1);
```

```
// เปิดไฟหลอด LED ที่ช่อง 17
```

```
  out(18,0);
```

```
// ปิดไฟหลอด LED ที่ช่อง 18
```

```
  sleep(1000);
```

```
// หน่วงเวลา 1 วินาที
```

```
  out(17,0);
```

```
// ปิดไฟหลอด LED ที่ช่อง 17
```

```
  out(18,1);
```

```
// เปิดไฟหลอด LED ที่ช่อง 18
```

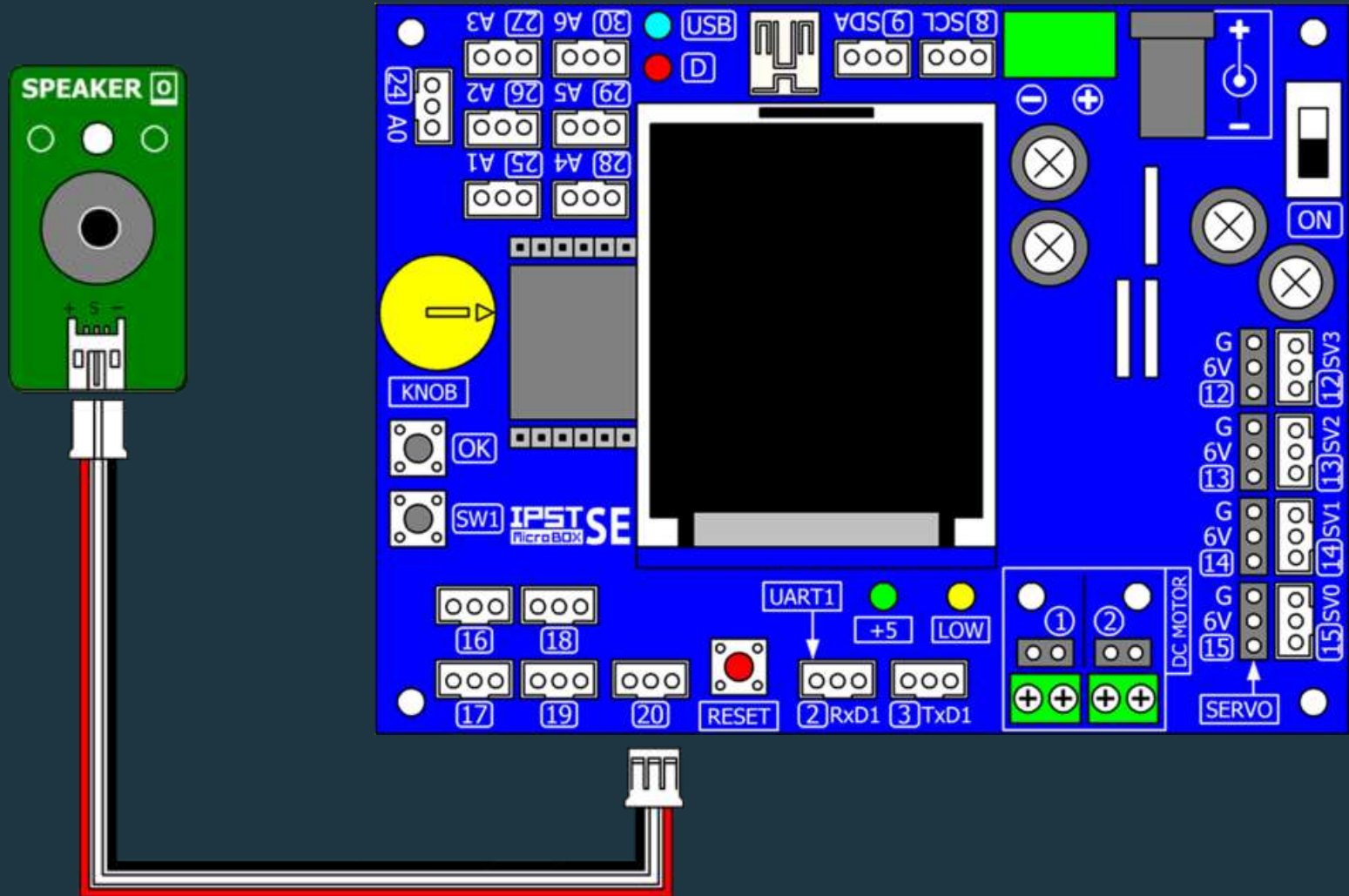
```
  sleep(1000);
```

```
// หน่วงเวลา 1 วินาที
```

```
}
```



# การแสดงผลทางลำโพง Speaker



# การแสดงผลทางลำโพง Speaker

การแสดงผลออกทางลำโพง หรือ  
การสร้างเสียงใน IPST-MicroBox จะมี  
2 ฟังก์ชัน ดังนี้

1. ฟังก์ชัน beep
2. ฟังก์ชัน sound



# การแสดงผลทางลำโพง Speaker

## ฟังก์ชัน beep

ฟังก์ชันกำเนิดเสียงอย่างง่าย beep : ทำหน้าที่  
กำเนิดเสียงความถี่ 500 Hz นาน 100 มิลลิวินาที

รูปแบบคำสั่ง

**beep(ch);**

**Ch** คือ ตำแหน่งช่องสัญญาณที่เชื่อมต่อ

รูปแบบการใช้งาน เช่น beep(20);



```
//โปรแกรมแสดงผลทางโพง beep (1 ครั้ง)
```

```
#include <ipst.h>
```

```
void setup ()
```

```
{  
  glcd(0,0,"Press Switch OK");
```

```
  sw_OK_press();
```

```
  beep(20);
```

```
}
```

```
void loop ()
```

```
{
```

```
}
```

// ส่งเสียง beep ที่ช่อง 20

ผลลัพธ์ ???



```
//โปรแกรมแสดงผลทางโพง beep (ทุกๆ 1 วินาที)
```

```
#include <ipst.h>
```

```
void setup ()
```

```
{  
  glcd(0,0,"Press Switch OK");  
  sw_ok_press();  
}
```

```
void loop ()
```

```
{  
  beep(20);  
  sleep(1000);  
}
```

ผลลัพธ์ ???



// ส่งเสียง beep ที่ช่อง 20



# การแสดงผลทางลำโพง Speaker

## ฟังก์ชัน Sound

ฟังก์ชันกำเนิดเสียงความถี่ใดๆ sound : ทำหน้าที่กำเนิดเสียงความถี่ ตามช่วงเวลาที่กำหนด

รูปแบบคำสั่ง

**sound(ch,freq,time);**

**Ch** คือ ตำแหน่งช่องสัญญาณที่เชื่อมต่อ

**freq** คือ ความถี่ของสัญญาณเสียง

**time** คือ ระยะเวลาที่กำเนิดเสียงมีหน่วยเป็นมิลลิวินาที

รูปแบบการใช้งาน เช่น **sound(20,1500,500);**



```
//โปรแกรมแสดงผลทางโพงที่ความถี่ 1500 Hz (1 ครั้ง)
```

```
#include <ipst.h>
```

```
void setup ()
```

```
{  
  glcd(0,0,"Press Switch OK");
```

```
  sw_OK_press();
```

```
  sound(20,1500,1000); // ส่งเสียงความถี่ 1500 Hz นาน 1 วินาที ที่ช่อง 20
```

```
}
```

```
void loop ()
```

```
{
```

```
}
```

ผลลัพธ์ ???



//โปรแกรมแสดงผลทางโพงที่ความถี่ 1500 Hz นาน 0.5 วินาที  
โดยเว้นห่างทุกๆ 1 วินาที

```
#include <ipst.h>
```

```
void setup ()
```

```
{  
  glcd(0,0,"Press Switch OK");
```

```
  sw_OK_press();
```

```
}
```

```
void loop ()
```

```
{
```

```
  sound(20,1500,500); // ส่งเสียงความถี่ 1500 Hz นาน 1 วินาที ที่ช่อง 20
```

```
  sleep(1000);
```

```
}
```

ผลลัพธ์ ???



# ใบงานชุดที่ 5

1. จงระบุชุดคำสั่งโปรแกรมพร้อมทั้งอธิบายรูปแบบคำสั่ง

1.1 คำสั่งแสดงผลทางหลอด LED

รูปแบบคำสั่ง .....

อธิบายรูปแบบคำสั่ง .....

.....

1.2 คำสั่งแสดงผลทางลำโพง Speaker (แบบ beep และ Sound)

รูปแบบคำสั่ง.....

อธิบายรูปแบบคำสั่ง .....

.....



# ใบงานชุดที่ 4

## 2. จงอธิบายหลักการทำงานของชุดคำสั่งโปรแกรมที่กำหนดให้

```
void loop ()  
{  
  beep(20);  
  sleep(1000);  
}
```

อธิบายหลักการทำงาน

.....

.....

.....

```
void loop ()  
{  
  sound(20,2500,500);  
  sleep(1000);  
}
```

อธิบายหลักการทำงาน

.....

.....

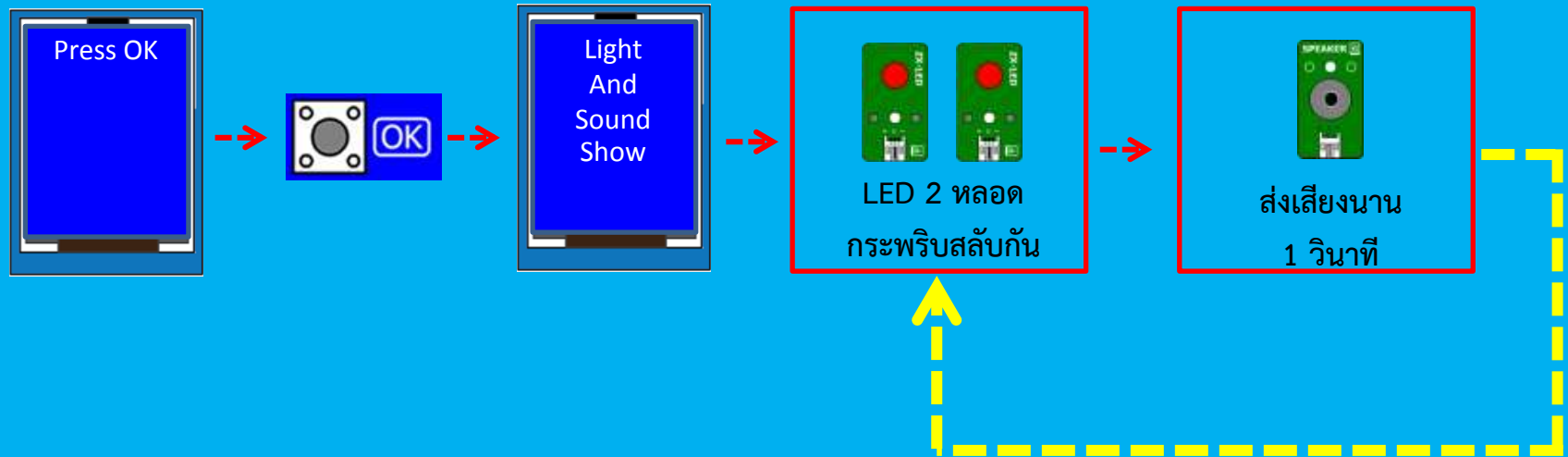
.....





# ใบงานชุดที่ 4

3. จงเขียนชุดคำสั่งโปรแกรมเพื่อแสดงผลลัพธ์ออกทางจอภาพตามลำดับภาพที่กำหนดให้ ?



# จบการนำเสนอ

