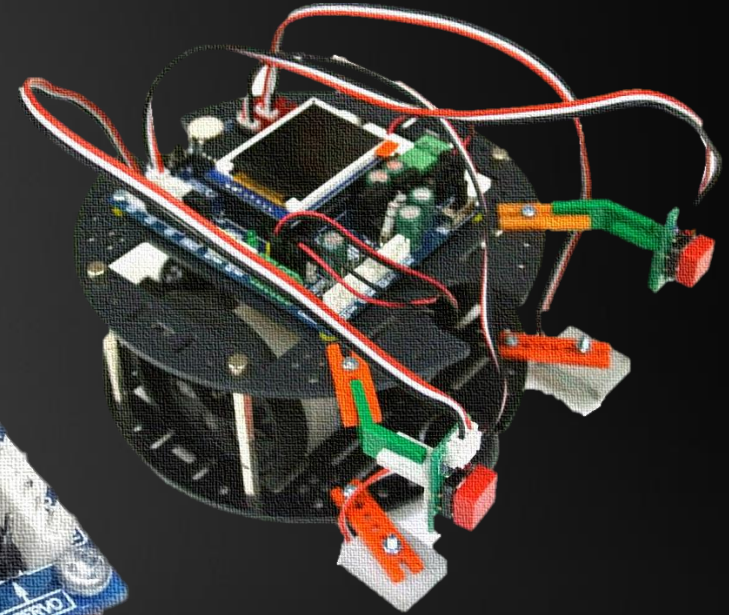
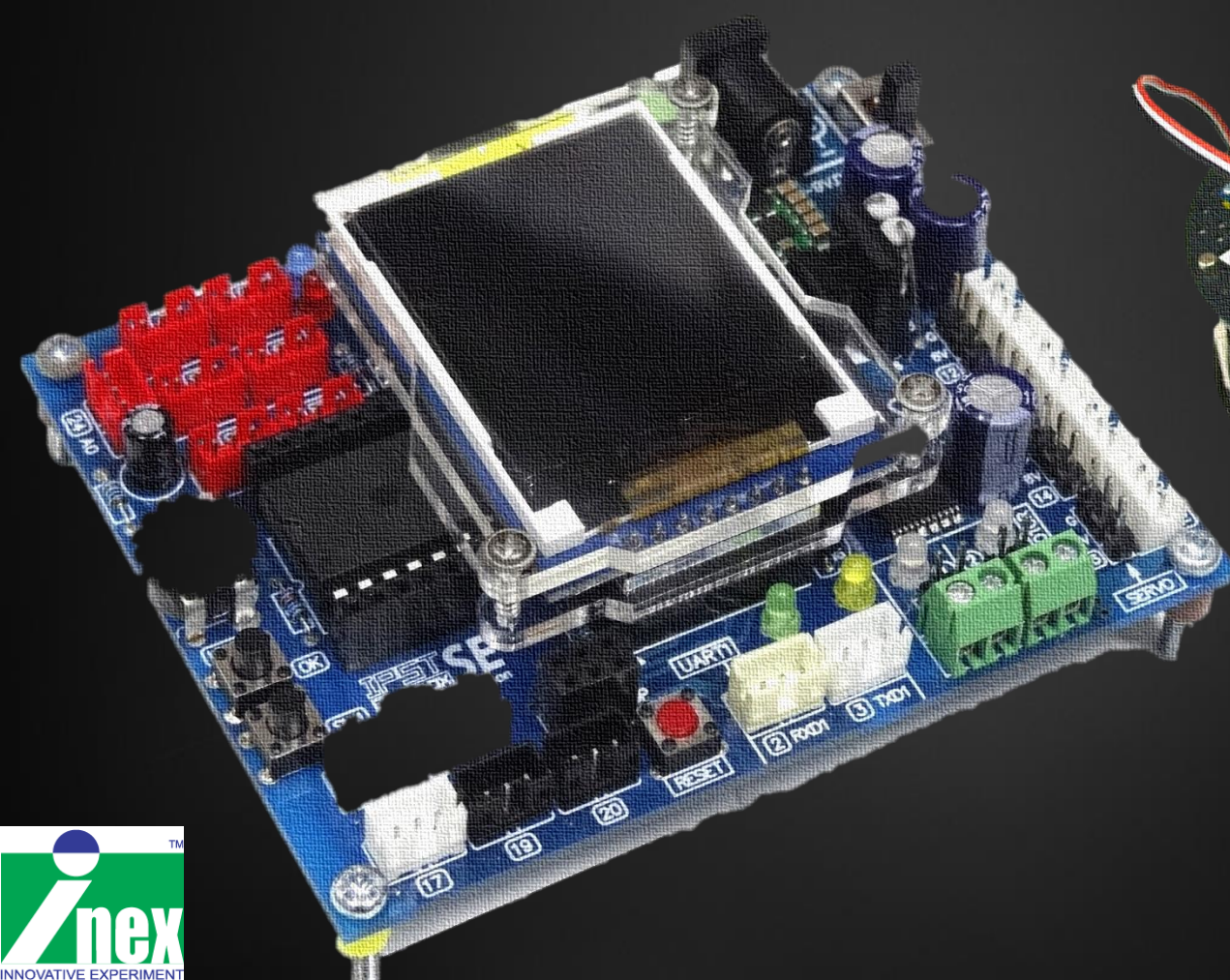


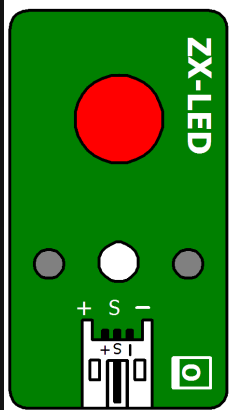
การจัดการกับ อุปกรณ์เชื่อมต่อ

IPST
MicroBOX
SE
Secondary Education

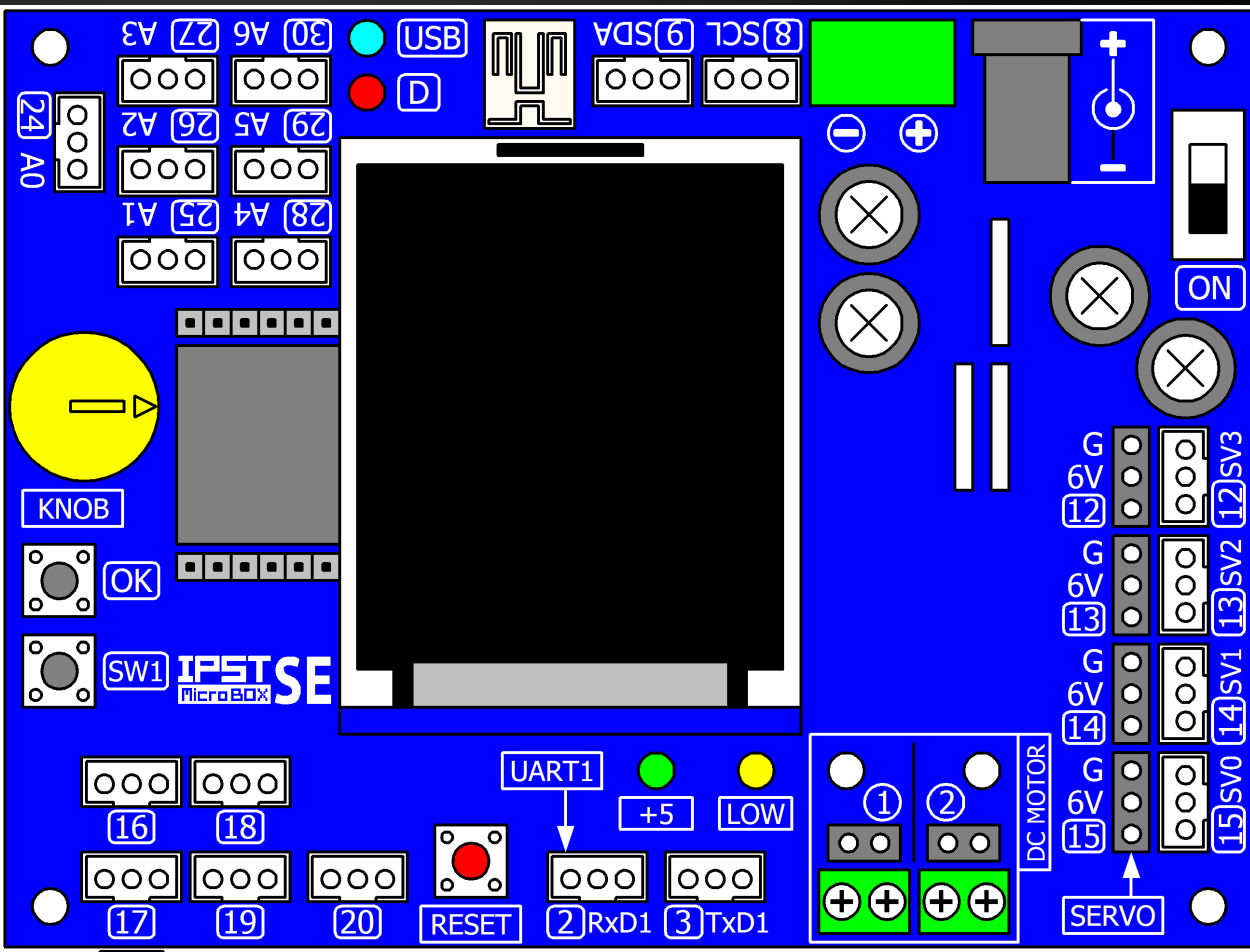
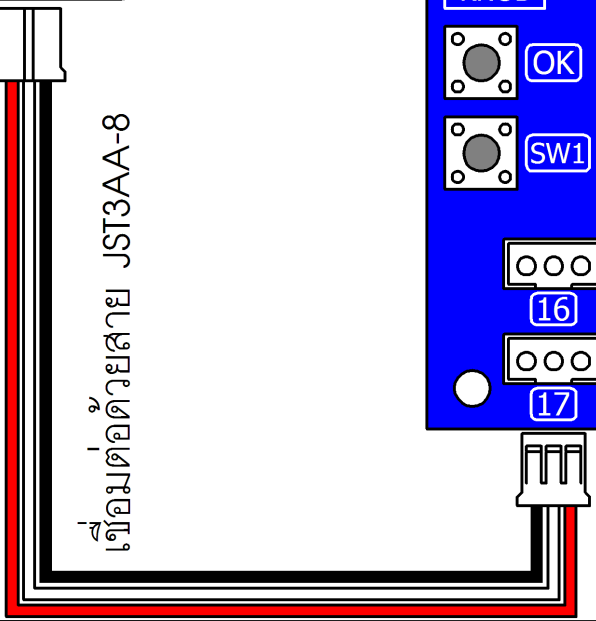


หลอด LED

เอาต์พุตดิจิทัลอย่างง่าย



เชื่อมต่อด้วยสาย JST3AA-8



คำสั่งส่งค่าออกเอาต์พุตดิจิทัล

out (ch, state) ;

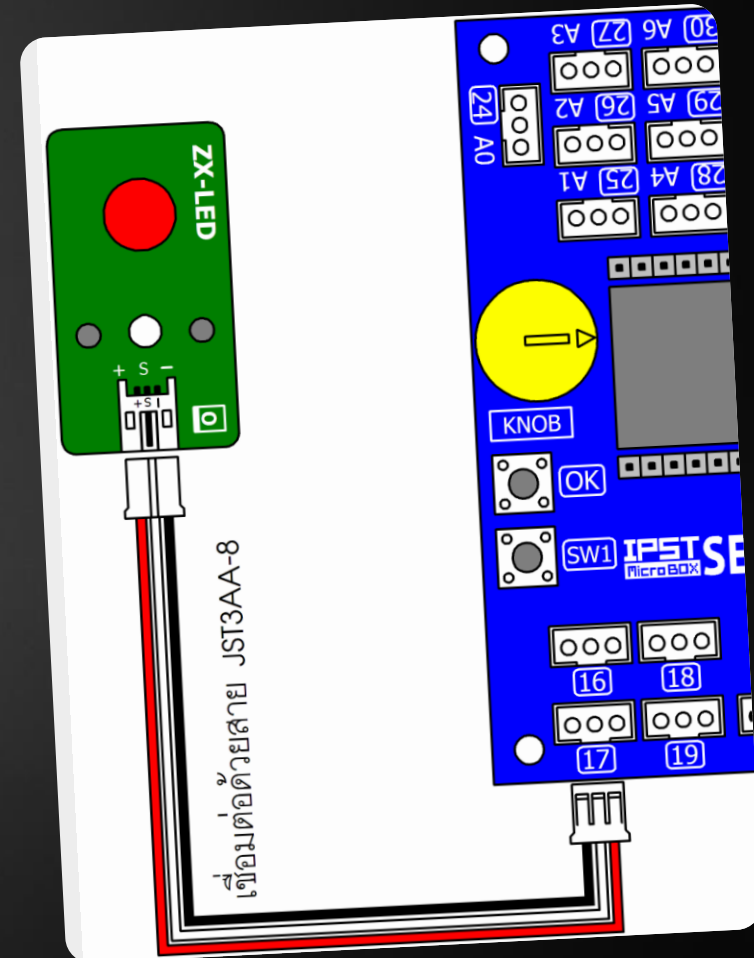
ส่งค่าสถานะ (*state*) 0 หรือ 1

ออกไปยังตำแหน่งขา (*ch*) ที่ระบุ

เช่น *out (17, 1) ;*

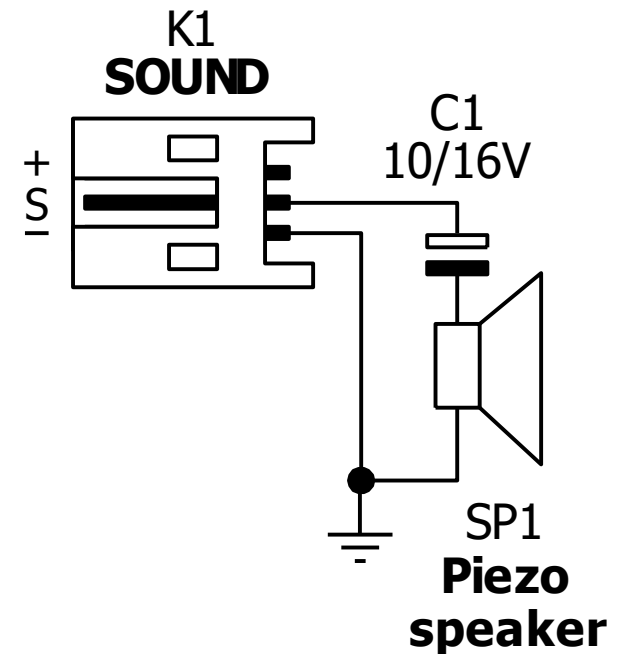
ตัวอย่าง : ไฟกะพริบ

```
#include <ipst.h>
void setup() { }
void loop() {
  out(17,1);
  delay(100);
  out(17,0);
  delay(100);
}
```

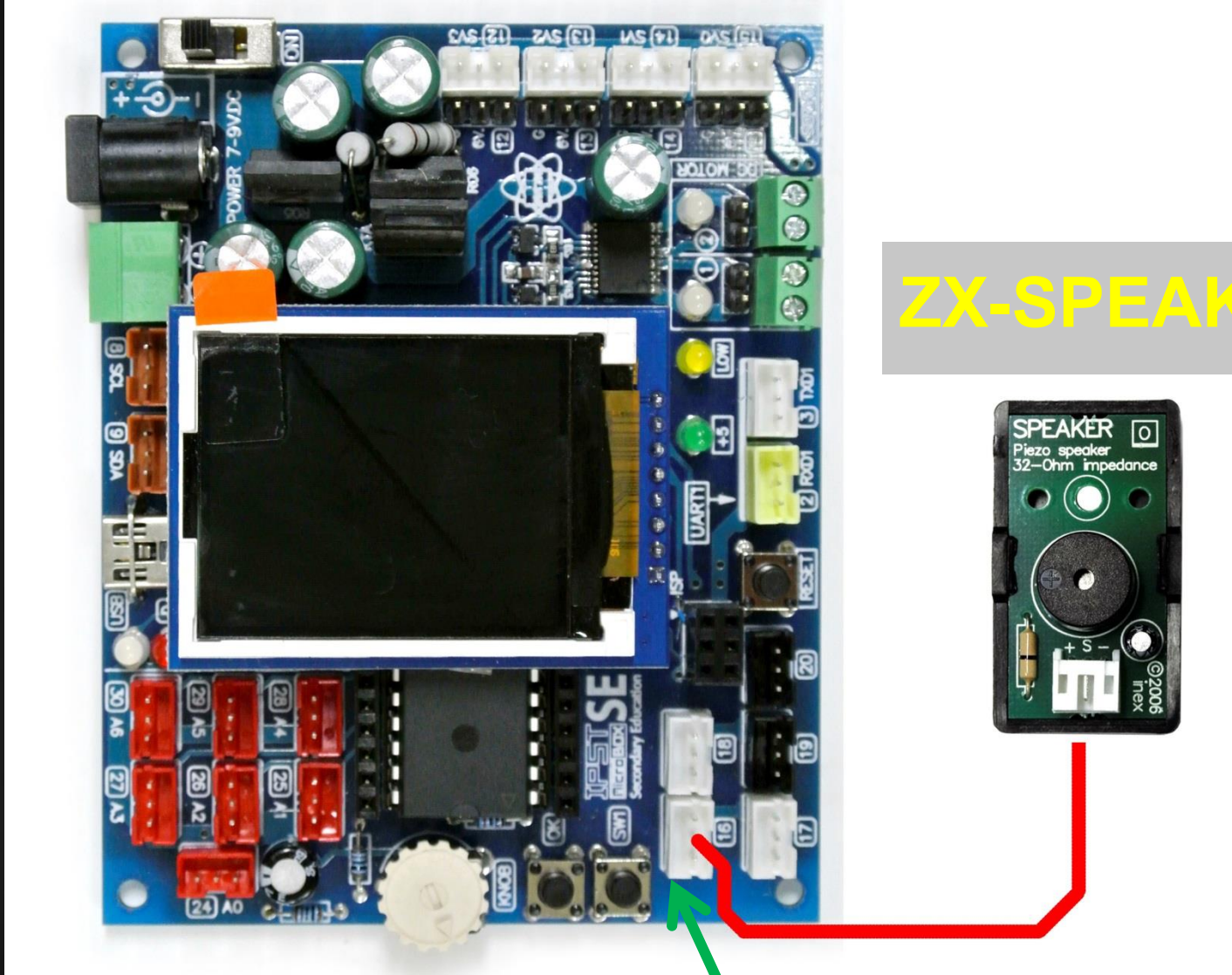


กำหนดเสียง : ลำโพงเปียโซ

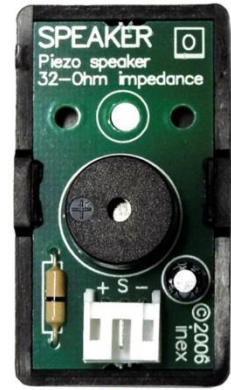
- ใช้ลำโพงเปียโซ มีอิมพีแดนซ์ 32W
- มีค่าความถี่ย่าน 300Hz ถึง 3000 Hz



ZX-SPEAKER



ZX-SPEAKER



ต่อช่อง 16

beep : กำเนิดเสียงความถี่ 500 Hz นาน 100 มิลลิวินาที

beep (ch) ;

sound : กำเนิดเสียงตามความถี่และช่วงเวลาที่กำหนด

sound (ch, freq, time) ;

freq กำหนดค่าความถี่เสียง

time กำหนดช่วงเวลายำเนิดเสียง มิลลิวินาที

ตัวอย่าง : สร้างเสียง 1

```
#include <ipst.h>
void setup() {
}
void loop() {
  beep(19);
  delay(1000);
}
```

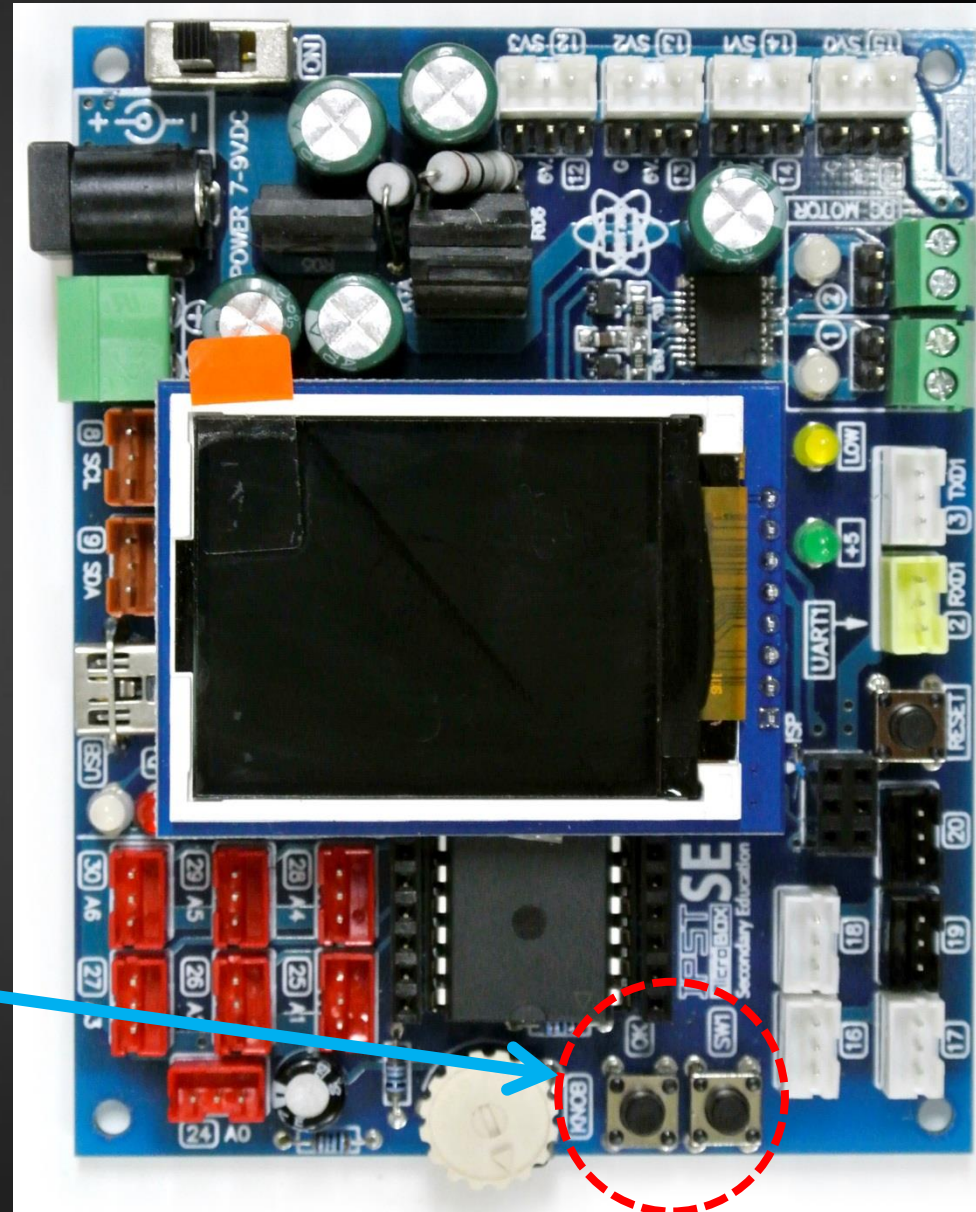
สร้างสัญญาณเสียงตื้อตื้อทุก ๆ 1 วินาที
(ความถี่เสียง 500 Hz ดังนาน 0.1 วินาที)

ตัวอย่าง : สร้างเสียง 2

```
#include <ipst.h>
void setup() {
}
void loop() {
  sound(19, 1200, 500);
  delay(1000);
}
```

สร้างสัญญาณเสียงความถี่ 1200 Hz ดังนาน 0.5 วินาที
เว้นทุก ๆ 1 วินาที

กดเป็น True
ไม่กดเป็น False



คำสั่ง sw_OK()

sw_OK() ตรวจสอบสวิตช์ OK บน IPST-SE

1 (True) เมื่อกดสวิตช์

0 (False) เมื่อไม่กดสวิตช์

หมายเหตุ การกดสวิตช์ทำให้ค่าที่อ่านได้จาก Knob มีค่าเป็น 0

ตัวอย่าง

```
if (sw_OK ())  
{  
    beep (19) ;  
}
```

ผลลัพธ์ เมื่อกดสวิตช์มีเสียงออกลำโพง

วนรอกดสวิตช์ OK เมื่อปล่อยสวิตช์ จะกระโดดไปทำคำสั่งบรรทัดถัดไป

ตัวอย่าง

```
sw_OK_press ();  
beep (19) ;
```

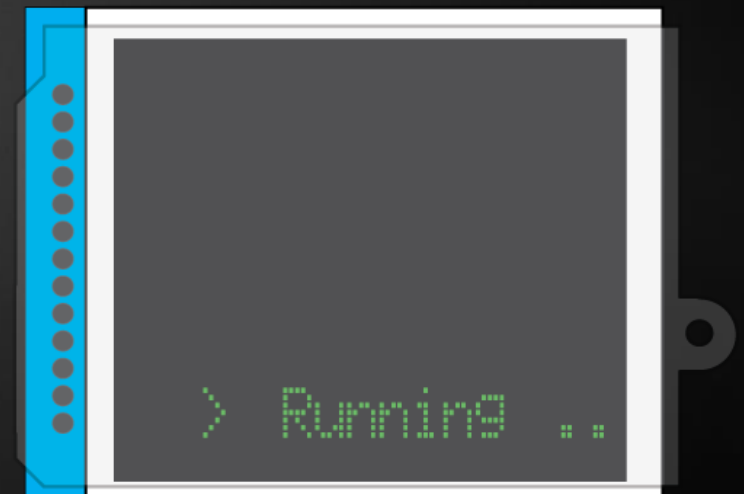
ผลลัพธ์ : รอกดสวิตช์ เมื่อกดส่งเสียงออกลำโพง

ฟังก์ชัน **OK()**

1. แสดงข้อความที่หน้าจอ
2. รอจนกระทั่งกดสวิตช์ **OK**
3. ทำงานคำสั่งถัดไป

ตัวอย่าง

```
#include <ipst.h>
void setup() {
  OK();
}
void loop() {
}
```





สวิตช์ SW1 อ่านค่าด้วยฟังก์ชัน `sw1()` และ `sw1_press()`

`sw1()` ตรวจสอบสวิตช์ SW1 บน IPST-SE

1 (True) เมื่อกดสวิตช์

0 (False) เมื่อไม่กดสวิตช์

ตัวอย่าง

```
if (sw1 ())  
{  
    out (17, 1) ;  
}
```

คำสั่ง `sw1_press()`

วนรอกดสวิตช์ **SW1** เมื่อปล่อยสวิตช์ จะกระโดดไปทำคำสั่งบรรทัดถัดไป

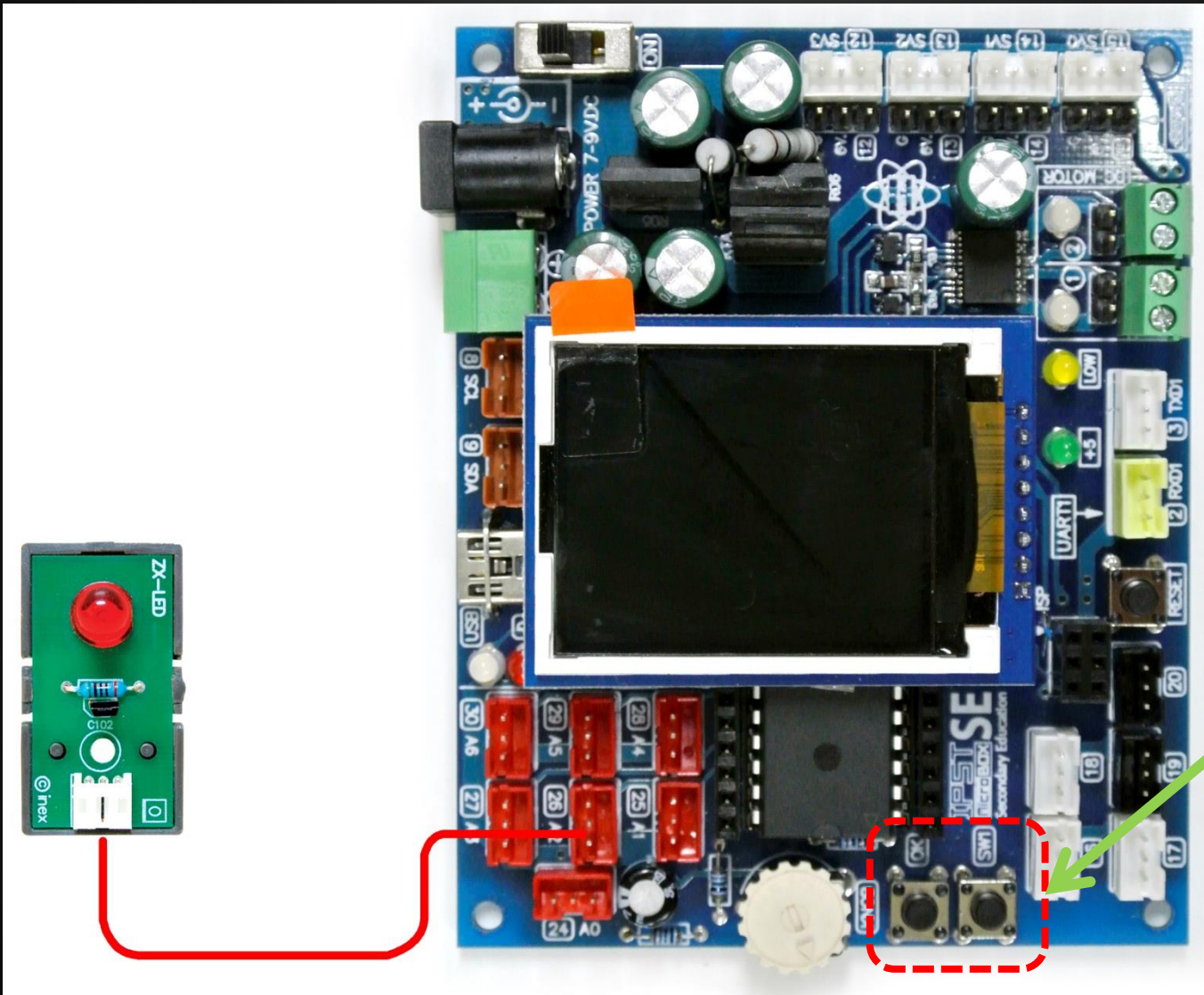
ตัวอย่าง

```
sw1_press ();  
beep (19) ;
```

ผลลัพธ์ : รอกดสวิตช์ SW1 เมื่อกดส่งเสียงออกลำโพง

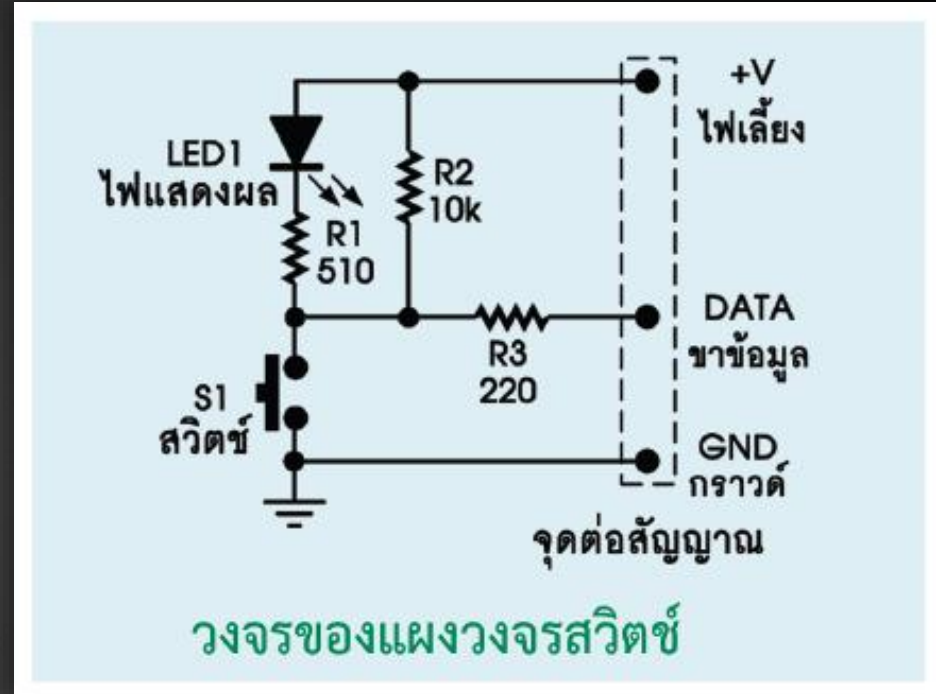
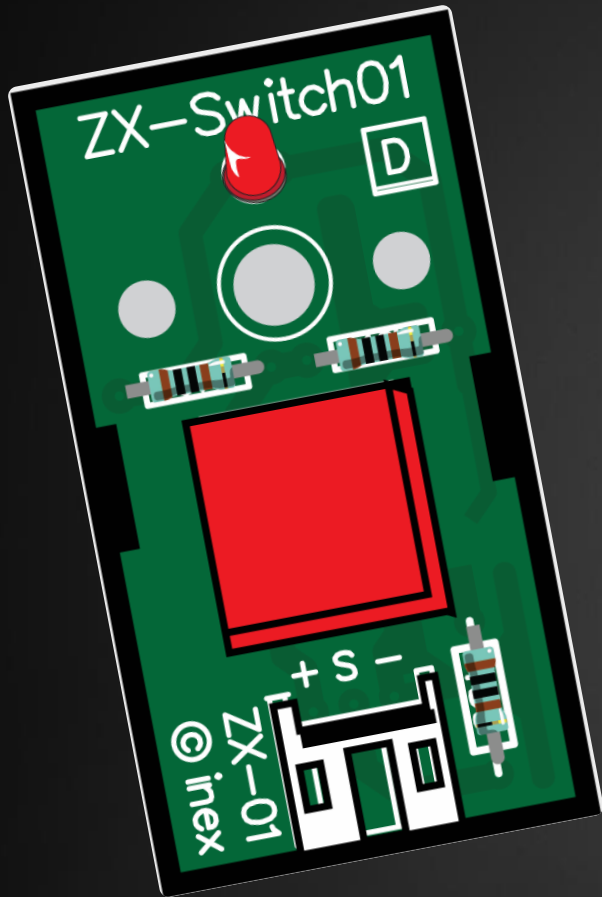
แบบทดสอบ 4

เขียนโปรแกรม ใช้สวิตช์ OK เปิด ใช้สวิตช์ SW1 ปิด



สวิตช์
OK และ SW1

แผงวงจรสวิตช์ : ZX-SWITCH01



กดสวิตช์ = ลอจิก “0”

ไม่กด = ลอจิก “1”

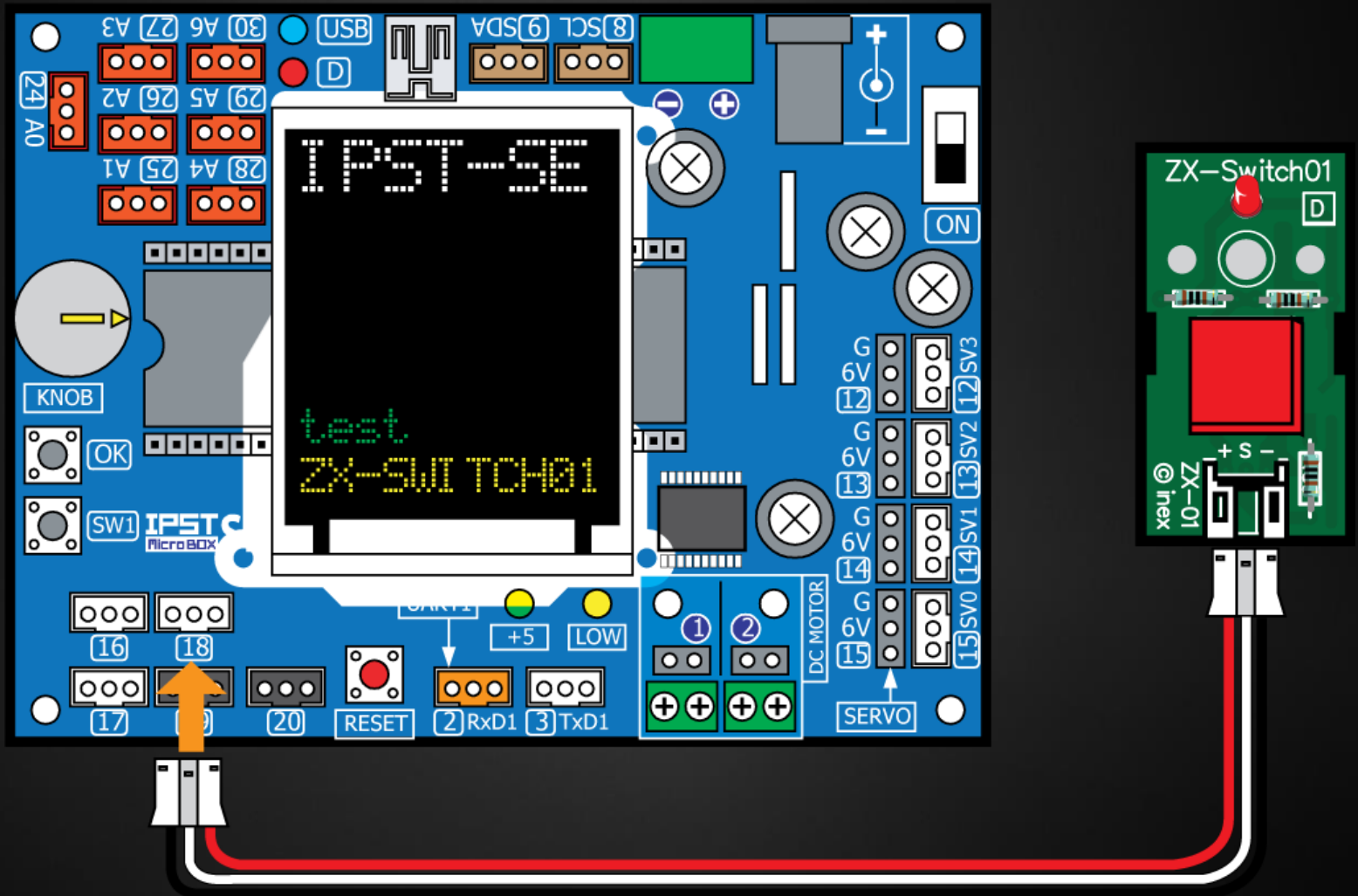
ฟังก์ชัน : **in()**

อ่านค่าขาดิจิตอลจากพอร์ตใด ๆ ของบอร์ด IPST-SE

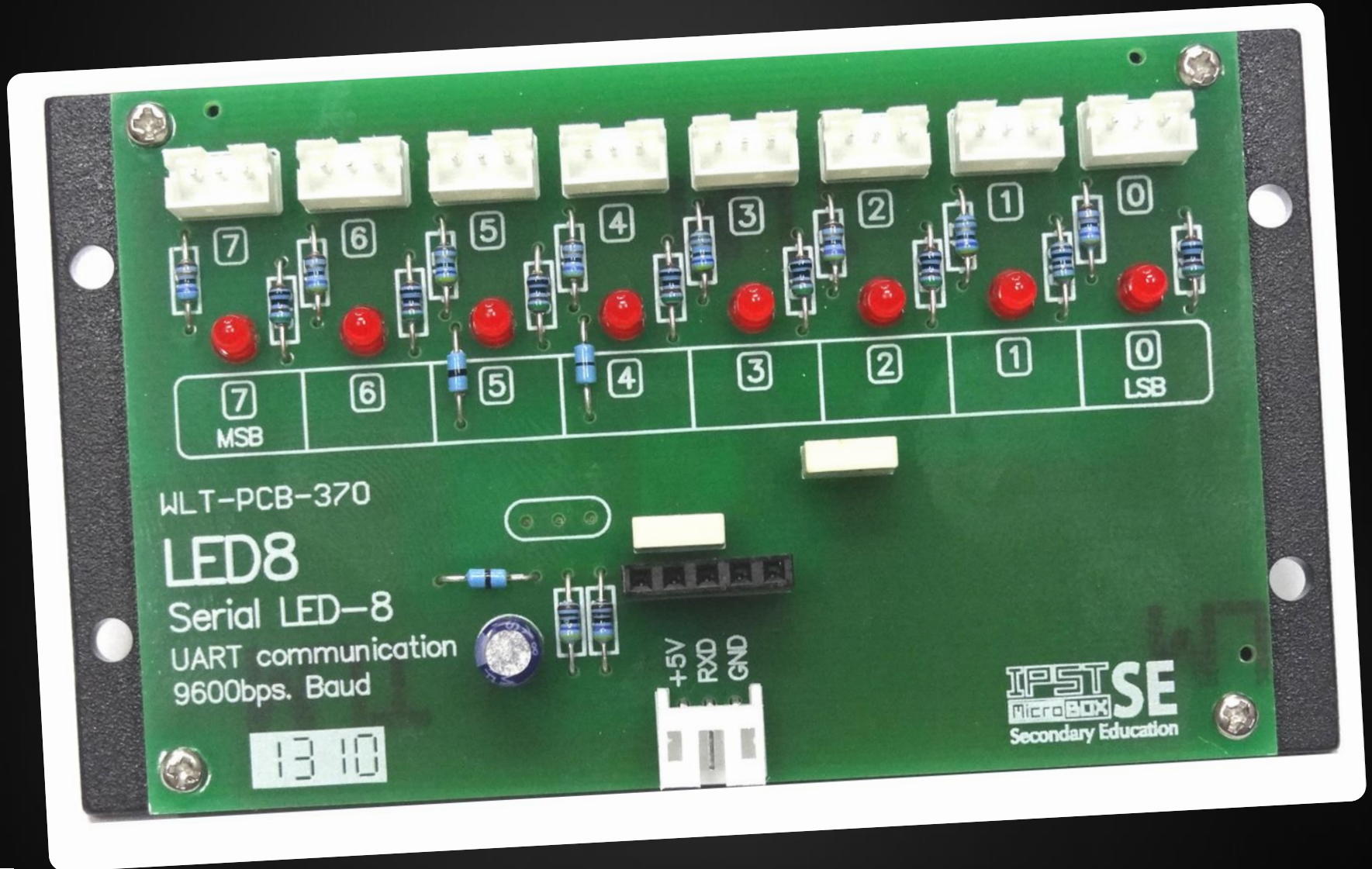
in (ch) ;

ch คือขาพอร์ตที่ต้องการอ่านค่าอินพุต
การคืนค่า คืนค่าสัญญาณดิจิตอลของตำแหน่งขาพอร์ตที่
อ่าน มีค่าเป็น 0 หรือ 1 เท่านั้น

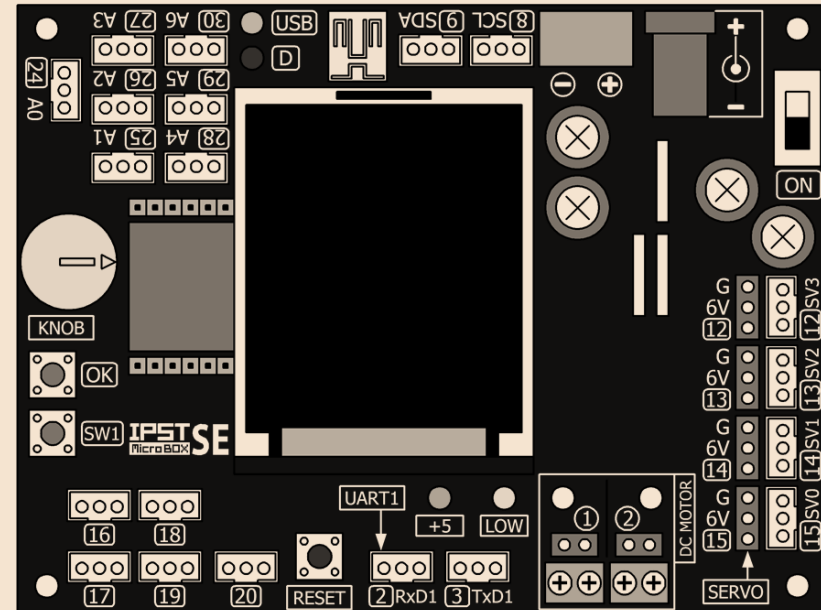
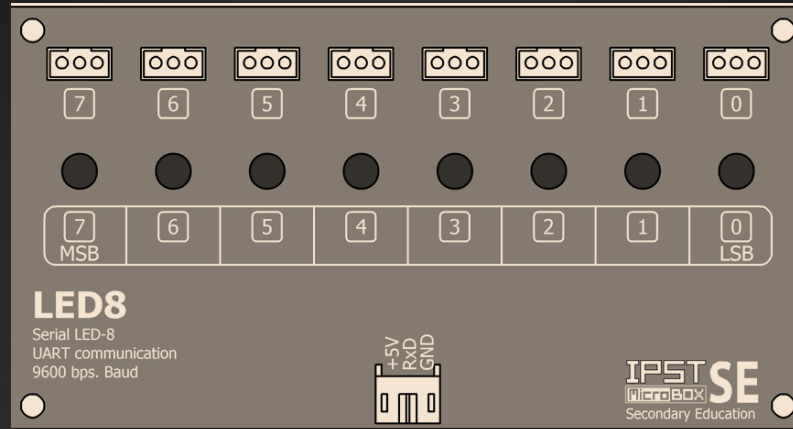
ทดสอบเขียนโปรแกรมกับ ZX-Switch01



โมดูล LED8



การเชื่อมต่อโมดูล LED8 กับ IPST-SE



เชื่อมต่อด้วยสาย JST3AA-8

ฟังก์ชัน LED8()

ส่งข้อมูล 1 ไบต์ไปยังบอร์ด LED8

LED8 (*pin*, *dat*);

pin ขาพอร์ตที่ต้องการติดต่อ

dat ข้อมูล 1 ไบต์ที่จะส่ง

ค่า 0 LED ดับหมด

ค่า 255 LED ติดทั้งหมด

| | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
|-----|-----|----|----|----|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 255 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |


```
#include <ipst.h>
byte x=1;
void setup() {}
void loop() {
    x=1;
    while (x<128) {
        LED8 (20 , x) ; x=x*2;
    }
    delay (200) ;
    while (x>1) {
        LED8 (20 , x) ; x=x/2;
    }
    delay (200) ;
}
```

ปุ่มปรับค่าอะนาล็อก knob()

หมุนเพื่อปรับค่า
80-1023



glcd แสดงข้อความที่จอ GLCD ได้ 21 ตัว 16 บรรทัด (size 1)

รูปแบบ

glcd(x, y, *p, ...)

พารามิเตอร์

x คือตำแหน่งบรรทัดมีค่าตั้งแต่ 0-15

y คือตำแหน่งตัวอักษรมีค่าตั้งแต่ 0-24

***p** คือข้อความที่ต้องการนำมาแสดง

ค่าพิเศษ

%d แสดงตัวเลขจำนวนเต็มในช่วง -32,768 ถึง 32,767

%h แสดงตัวเลขฐานสิบหก

%b แสดงตัวเลขฐานสอง

%l แสดงตัวเลขจำนวนเต็มในช่วง -2,147,483,648 ถึง 2,147,483,647

%f แสดงผลตัวเลขจำนวนจริง (แสดงทศนิยม 3 หลัก)

การแสดงผลค่าตัวเลขด้วย **glcd**


`glcd(0, 0, "%d", 100);`



ค่าพิเศษ

%d แสดงตัวเลขจำนวนเต็มในช่วง -32,768 ถึง 32,767

%h แสดงตัวเลขฐานสิบหก

%b แสดงตัวเลขฐานสอง

%l แสดงตัวเลขจำนวนเต็มในช่วง -2,147,483,648 ถึง 2,147,483,647

%f แสดงผลตัวเลขจำนวนจริง (แสดงทศนิยม 3 หลัก)

คำสั่ง `knob()`

`knob` ใช้อ่านค่าตัวต้านทานปรับค่าได้บน IPST-SE
ทำงานเหมือน คำสั่ง `analog(7)` ค่าอยู่ในช่วง **80-1023**

ตัวอย่าง

```
int val=0;
  val=knob ();
  lcd (" %d", val);
```

คำสั่ง knob(x)

รูปแบบ

knob(x) ; โดย x คือค่า scale

การคืนค่า

ค่าที่อ่านได้จาก knob มีค่าระหว่าง 0 ไปจนถึงค่า x

ตัวอย่าง

```
int val=0;  
val=knob(180);  
glcd("%d ",val);
```

หน้าจอ glcd แสดงค่าตัวเลข 0-180

รูปแบบ

knob(x,y);

x คือค่า Scale ช่วงเริ่มต้น

y คือค่า Scale ช่วงท้าย

ตัวอย่าง

```
glcd(1,1,"%d ",knob(10,90));
```

หน้าจอจะแสดงค่า 10-90 ตามการหมุน knob()

ตัวอย่าง **bar graph** เมื่อปรับ knob

```
#include <ipst.h> // include file for IPST-SE
int x;
void setup() {
  setTextSize(3);
}
void loop() {
  glcd(1,0,"%d  ",knob(128));
  glcdFillRect(0,80,knob(128),10,GLCD_RED);
  glcdFillRect(knob(128),80,128-knob(128),10,GLCD_BLACK);
}
```



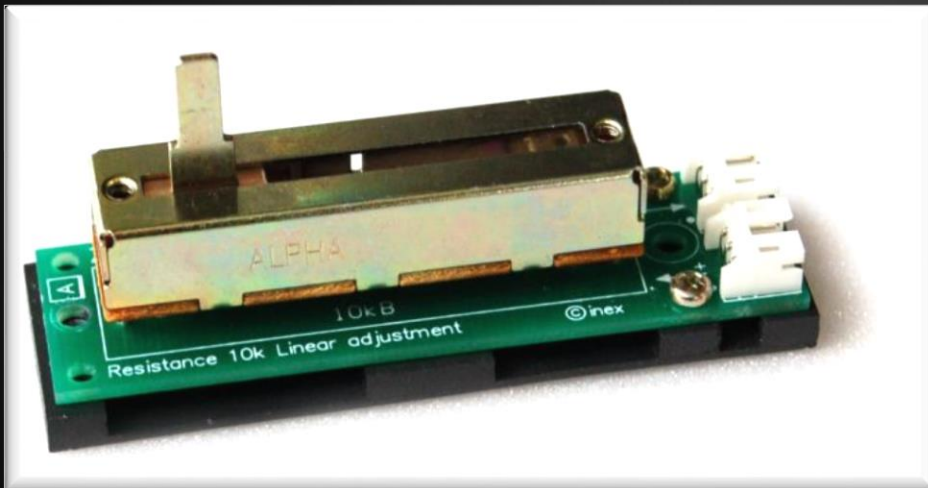
แผงวงจร ตัวต้านทานปรับค่าได้



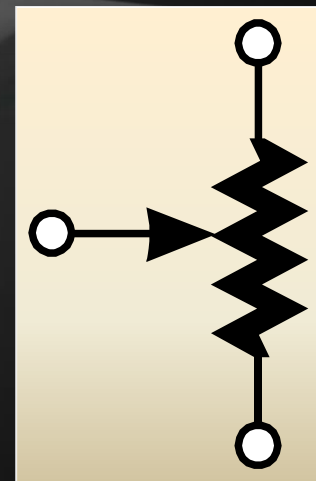
ตัวต้านทานปรับค่าได้แบบตัวนอน



ตัวต้านทานปรับค่าได้แบบตัวตั้ง

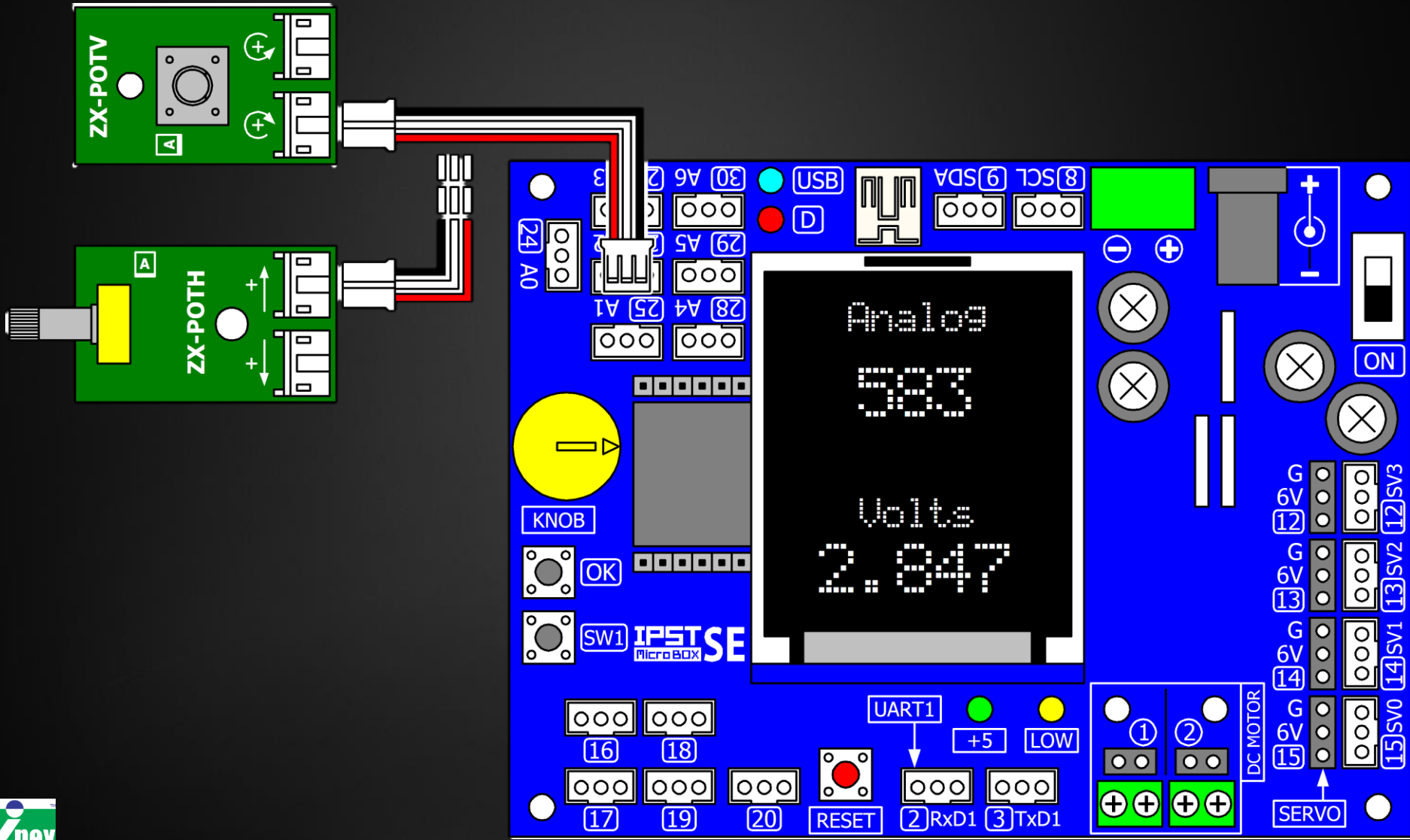


ตัวต้านทานปรับค่าได้แบบเลื่อน



สัญลักษณ์

แผงวงจร ตัวต้านทานปรับค่าได้



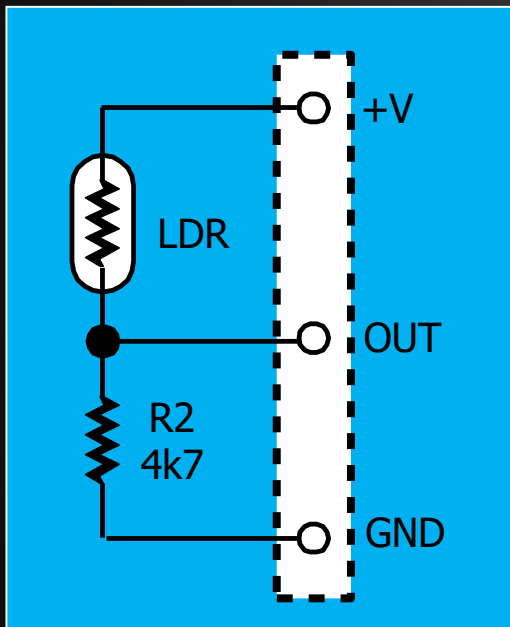
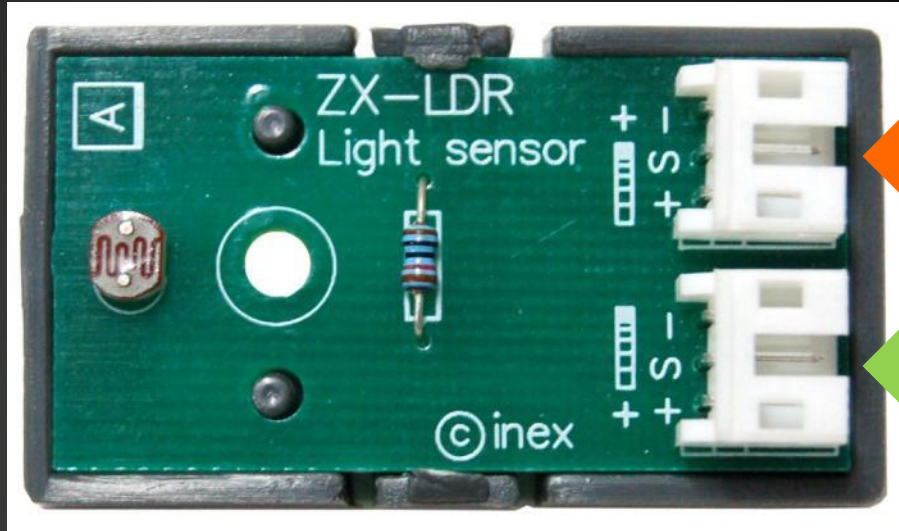
อ่านค่าอะนาลอกจากตำแหน่งพอร์ตที่ระบุ (A0-A6)

analog(ch) ;

ch คือขาพอร์ตอะนาลอก (A0-A6) : ระบุเฉพาะตัวเลข

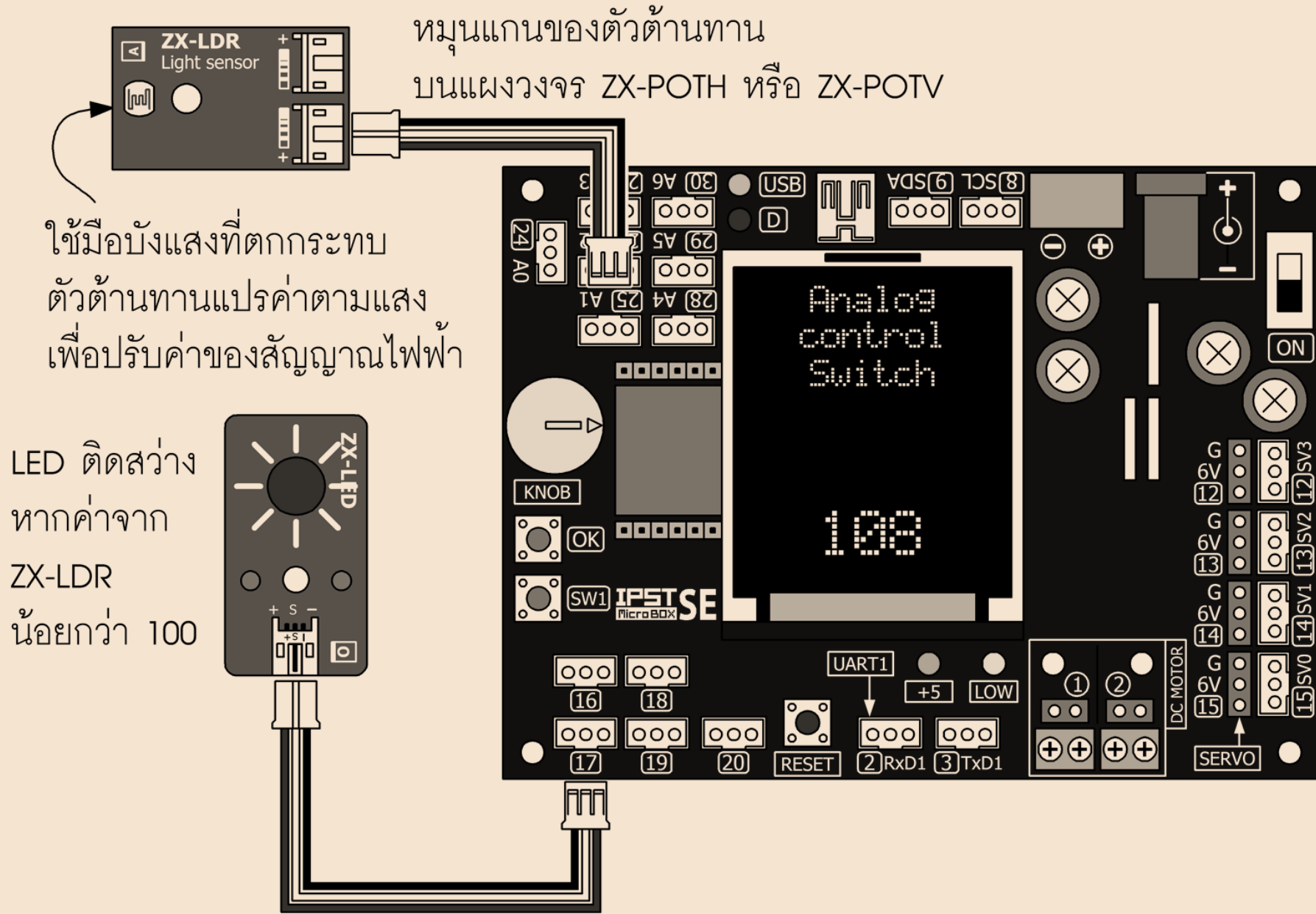
ผลลัพธ์ : ค่า 0-1023 (10 บิต) จากตำแหน่งขาพอร์ตที่ต้องการ

แผงวงจรตรวจจับแสง ZX-LDR



ใช้ตรวจจับแสงสว่าง เลิกเอาต์พุตได้ 2 แบบคือ
 แรงดันเอาต์พุตเพิ่ม เมื่อแสงตกกระทบมากขึ้น
 แรงดันเอาต์พุตลดลง เมื่อแสงตกกระทบมากขึ้น

สวิตช์เปิดไฟกลางคืน



ตัวอย่าง : ไฟฉายสั่งเปิด/ปิดไฟ

```
#include <ipst.h>
int x,y=0;
void setup() {}
void loop() {
  x=analog(6);
  glcd(0,0,"LDR=%d  ",x);
  if(x<200) {
    while(analog(6)<200);delay(300);
    if(y==0) {
      out(16,1); y=1;
    }
    else{
      out(16,0);y=0;
    }
  }
}
```

ตัวอย่าง : โปรแกรมนับคนเข้าห้องสมุด

```
#include <ipst.h>
int x,y=0;
void setup(){setTextSize(3);}
void loop(){
  x=analog(6);
  glcd(0,0,"LDR=%d  ",x);
  if(x<200){
    while(analog(6)<200);delay(300);
    y++;
    glcd(2,0,"%d  ",y);
  }
}
```

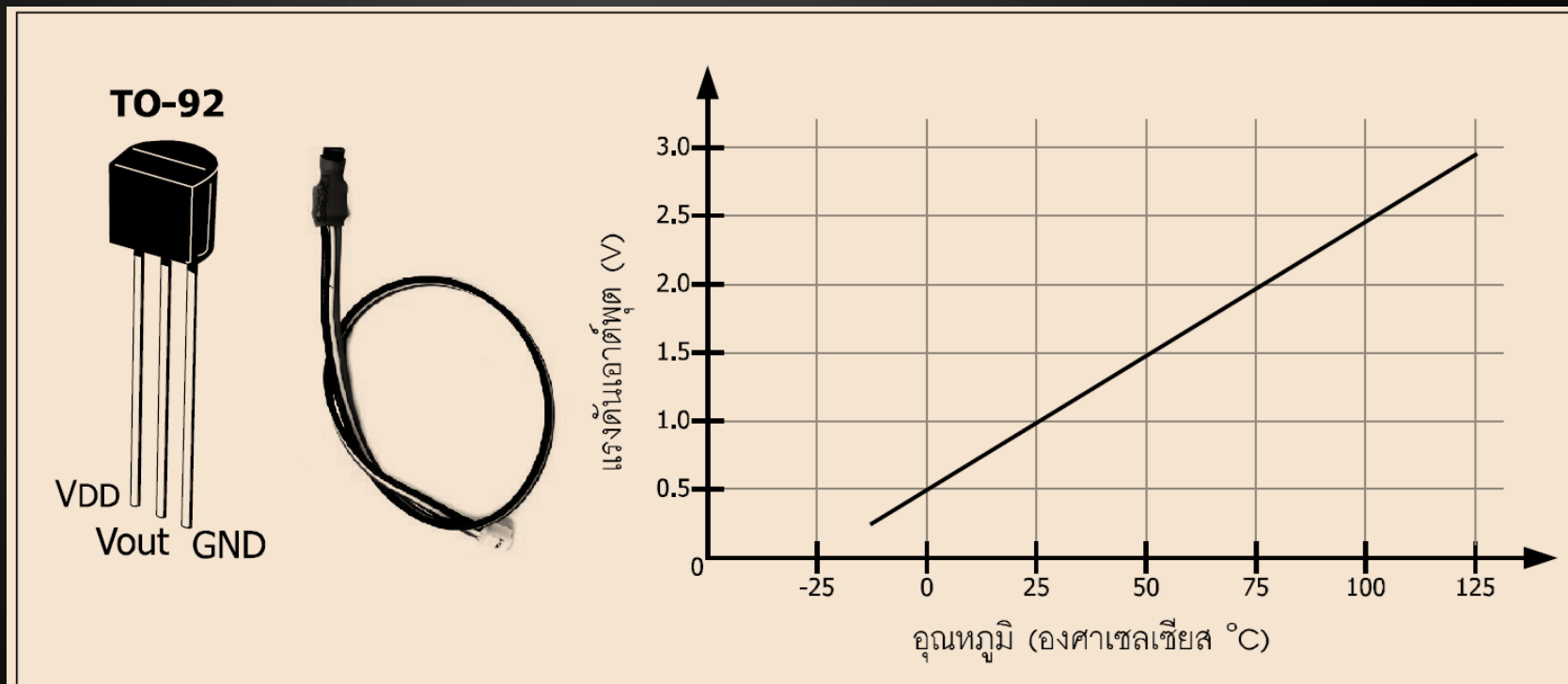
ไอซีวัดอุณหภูมิ MCP9701

แรงดันเอาต์พุตเปลี่ยนแปลง **19.5mV/องศา**

คำนวณจากค่าอะนาล็อกที่อ่านได้ จากสูตร

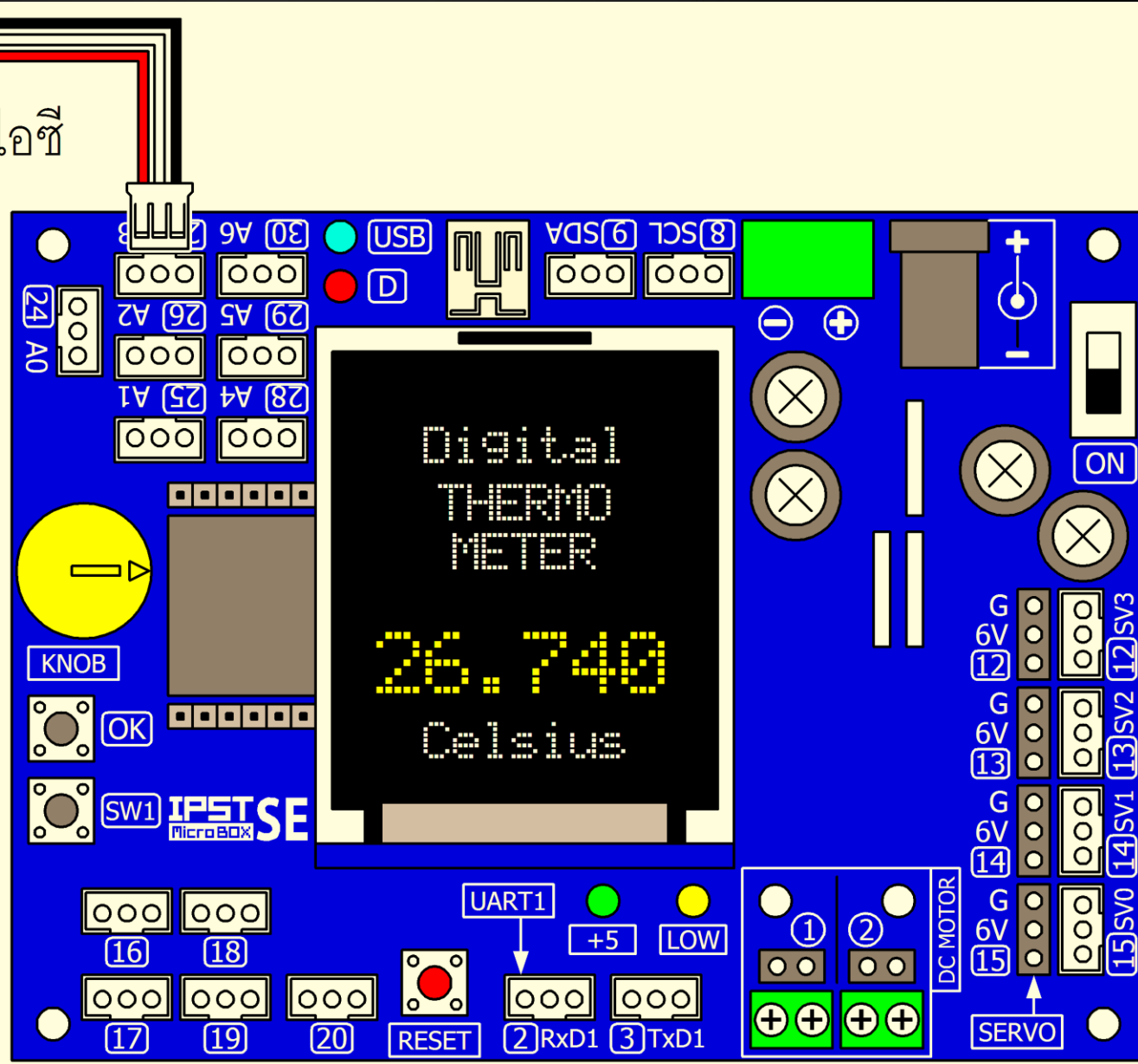
$$\text{Temp} = (\text{val} \times 0.25) - 20.51$$

val ค่าอะนาล็อกที่อ่านได้จาก **IPST-SE**



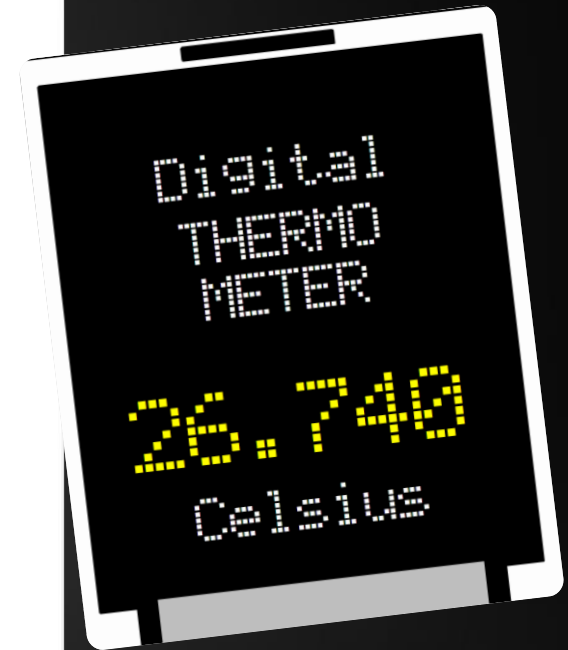
คำสั่ง `sw_ok_press()`

สายวัดอุณหภูมิใช้ไอซี
เบอร์ MCP9701

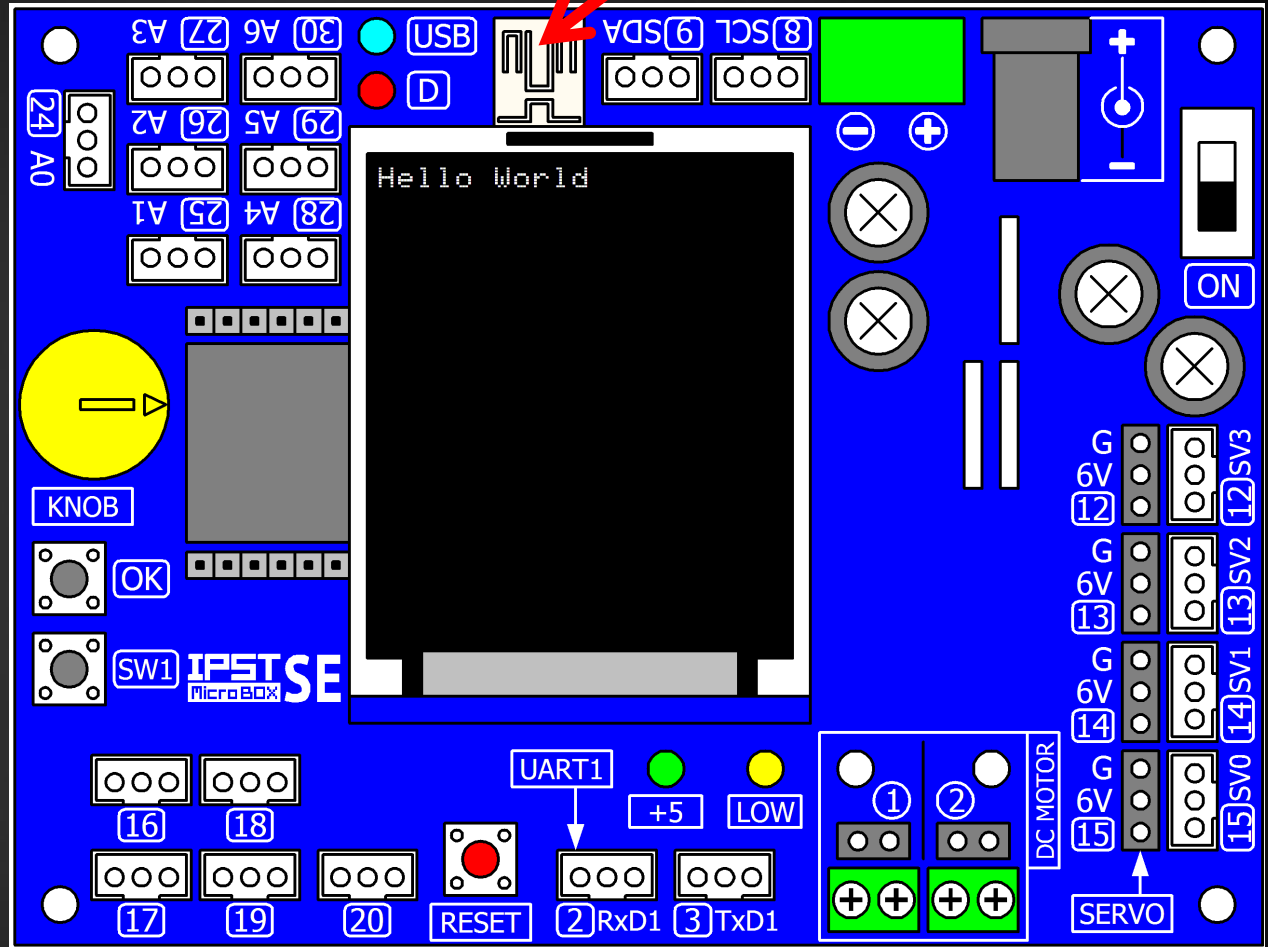


ไอซีวัดอุณหภูมิ MCP9701

```
#include <ipst.h>
int val,i;
float Temp;
void setup(){ glcdClear(); setTextSize(2); }
void loop(){
  glcd(1,2,"Digital");
  glcd(2,2,"THERMO");
  glcd(3,3,"METER");
  val=0;
  for (i=0;i<20;i++) {val = val+analog(3); }
  val = val/20;
  Temp = (float(val)*0.25) - 20.51 ;
  setTextSize(3);
  setTextColor(GLCD_YELLOW);
  glcd(3,1,"%f",Temp);
  setTextColor(GLCD_WHITE);
  setTextSize(2);
  glcd(6,2,"Celsius");
  delay(500);
}
```



UART



UART1

คำสั่งสำหรับสื่อสารอนุกรม

uart_available() ถ้ามีข้อมูลถูกป้อนเข้ามาเงื่อนไขเป็นจริง

uart_getkey() รับค่าข้อมูล 1 ไบต์

uart ส่งข้อมูลหลายๆ ไบต์ออกไป

uart_putc ส่งข้อมูลออกไปตัวเดียว

uart1_available() ถ้ามีข้อมูลถูกป้อนเข้ามาเงื่อนไขเป็นจริง

uart1_getkey() รับค่าข้อมูล 1 ไบต์

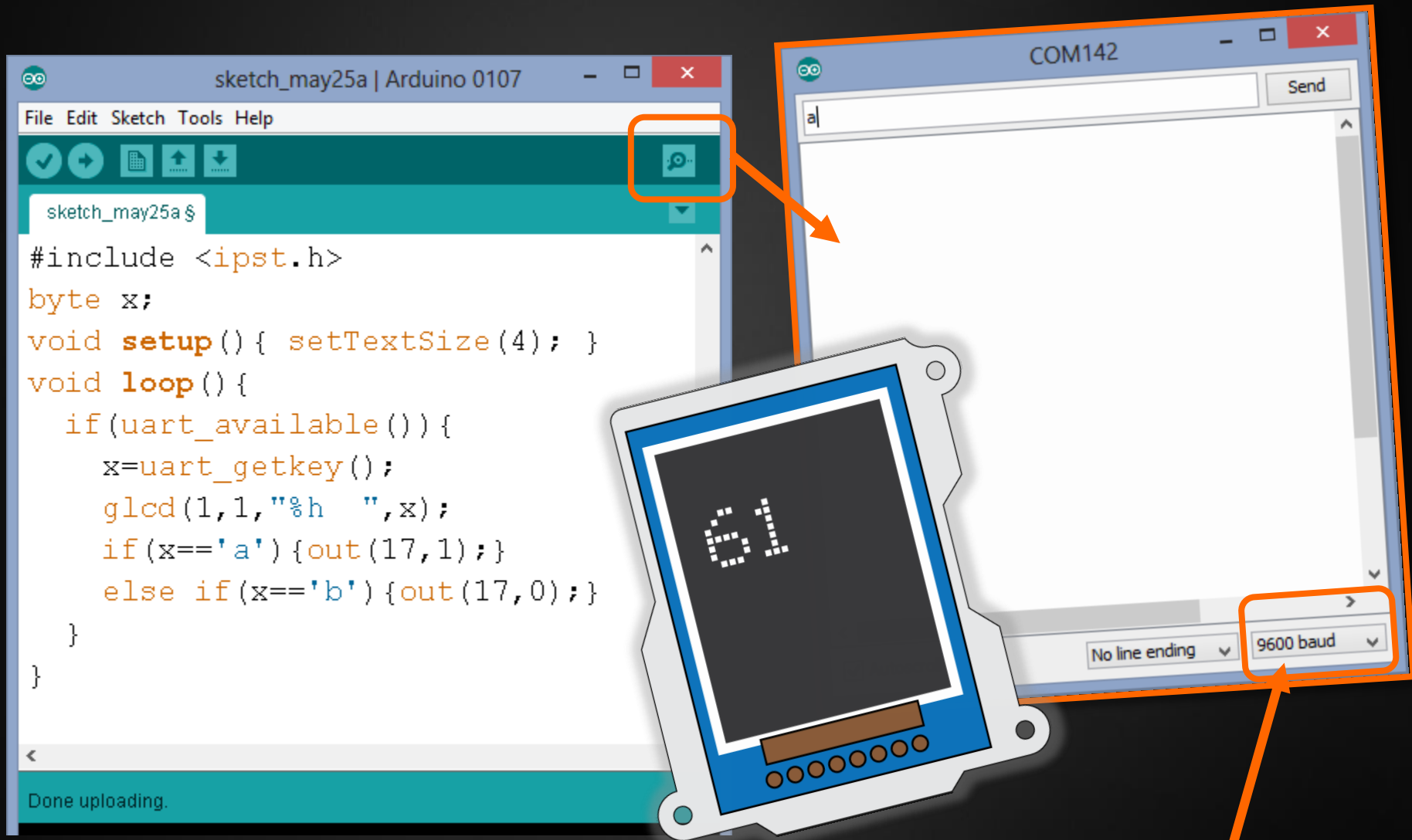
uart1 ส่งข้อมูลหลายๆ ไบต์ออกไป

uart1_putc ส่งข้อมูลออกไปตัวเดียว

ตัวอย่าง : รับค่าจากคอมพิวเตอรื → LED

```
#include <ipst.h>
byte x;
void setup() { setTextSize(4); }
void loop() {
  if(uart_available()) {
    x=uart_getkey();
    glcd(1,1,"%h  ",x);
    if(x=='a') {out(17,1);}
    else if(x=='b') {out(17,0);}
  }
}
```

การเปิดหน้าต่าง Serial monitor



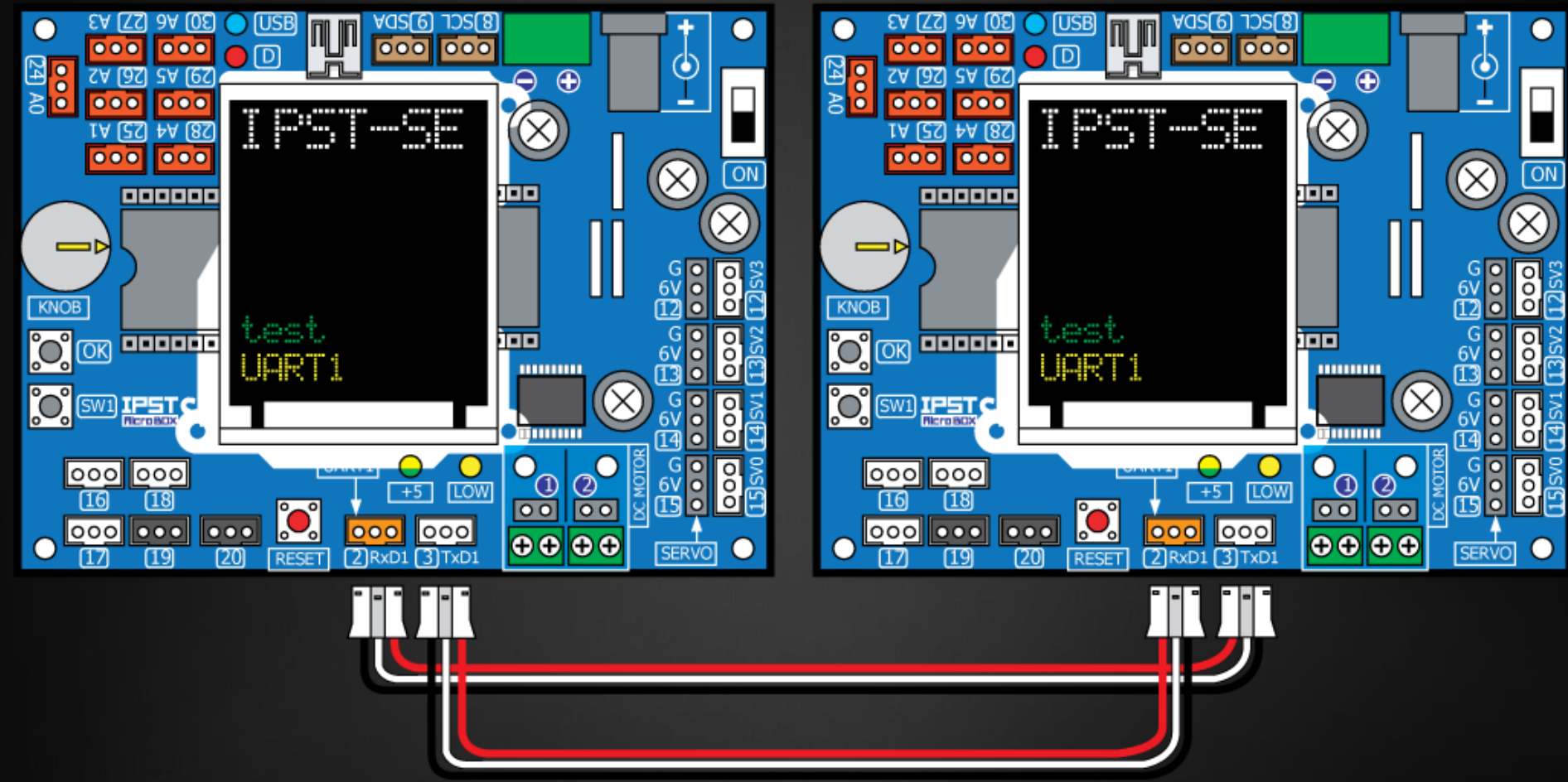
The image shows the Arduino IDE interface. The main window displays the following code:

```
sketch_may25a | Arduino 0107
File Edit Sketch Tools Help
sketch_may25a $
#include <ipst.h>
byte x;
void setup() { setTextSize(4); }
void loop() {
  if(uart_available()) {
    x=uart_getkey();
    glcd(1,1,"%h ",x);
    if(x=='a') {out(17,1);}
    else if(x=='b') {out(17,0);}
  }
}
```

The Serial Monitor window (COM142) is open, showing the character 'a' in the input field. The baud rate is set to 9600. The status bar at the bottom of the IDE indicates "Done uploading."

บอดเรตต้องเท่ากับ 9600

สื่อสารระหว่างไมโครคอนโทรลเลอร์ 2 บอร์ด



ตัวอย่างการรับ และ ส่งข้อมูล

ภาครับ

```
#include <ipst.h>
int x;
void setup() {
}
void loop() {
    if(uart1_available()) {
        x=uart1_getkey();
        glcd(1,1,"%d ",x);
    }
}
```

ภาคส่ง

```
#include <ipst.h>
int x=0;
void setup() {}
void loop() {
    uart1_putc(x);
    x++;
    delay(300);
}
```


ไฟล์ไลบรารี ipst.h

จอแสดงผลกราฟิก LCD สี



ipst_glcd.h

- glcd
- setTextColor
- setTextBackgroundColor
- glcdClear
- glcdFillScreen
- glcdMode
- setTextSize
- getTextColor
- getTextBackgroundColor
- getTextSize
- glcdGetMode
- glcdPixel
- glcdRect
- glcdFillRect
- glcdLine
- glcdCircle
- glcdFillCircle
- glcdArc

รับส่งข้อมูลอนุกรมผ่านหน้าต่าง Serial Monitor ของ Wiring IDE



ipst_serial.h

- uart
- uart_putc
- uart_puts
- uart_set_baud
- uart_get_baud
- uart_available
- uart_getkey
- uart1
- uart1_putc
- uart1_puts
- uart1_set_baud
- uart1_get_baud
- uart1_available
- uart1_getkey



แผงวงจรขับ เซอร์โวมอเตอร์ 16 ช่อง



ไมโครบลูทูธ



ไมโคร XBEE

ipst_sound.h

- beep
- sound



แผงวงจร ขับลำโพง

ipst_led8.h

- pinLED8
- LED8



แผงวงจร LED8

ipst_servoMotor.h

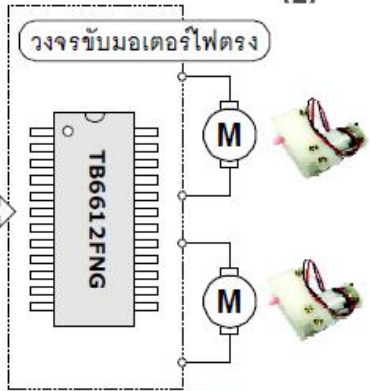
- servo



เซอร์โวมอเตอร์

ipst_motor.h

- motor
- ao
- motor_stop
- fd
- fd2
- bk
- bk2
- tl
- tr
- sl
- sr



วงจรขับมอเตอร์ไฟตรง



แผงวงจรขับ LED ZX-LED

ipst_in_out.h

- in - sw1
- out



แผงวงจรสวิตช์ ZX-SWITCH01

ipst_analog.h

- analog
- knob
- sw_ok
- sw_ok_press



ตัวตรวจจับแบบต่างๆ ที่ให้ผลการทำงานเป็นแรงดันไฟฟ้า

ipst_sleep.h

- sleep
- delay
- delay_us