



EXCEED

14

HARDWARE



# Challenge : Intro to multitask

ต้องการสร้างไฟ LED 2 ดวง ที่ไฟดวงแรกสว่าง 1000 ms. และดับ 500 ms.  
ส่วนอีกดวงสว่าง 700 ms. และดับ 300 ms. เข้าไปมาเรื่อยๆ โดยทั้ง2จะเริ่ม  
ต้นจากสถานะไฟสว่าง



# Code : Intro to multitask

```
#define LED1 11
#define LED2 12

void setup() {
    pinMode(11, OUTPUT);
    pinMode(12, OUTPUT);
}

void taskLED1() {
    digitalWrite(LED1, HIGH);
    delay(1000);
    digitalWrite(LED1, LOW);
    delay(500);
}
```

# Code : Intro to multitask <sup>(Continues)</sup>

```
void taskLED2() {  
    digitalWrite(LED2,HIGH);  
    delay(700);  
    digitalWrite(LED2,LOW);  
    delay(300);  
}
```

```
void loop() {  
    taskLED1();  
    taskLED2();  
}
```

ไม่เวิร์ค เพราะว่าโปรแกรมจะทำ taskLED1 ก่อน เมื่อเสร็จแล้วจึงค่อยทำ taskLED2 (การทำงานไม่ได้เป็นอิสระต่อกัน)



# Process

```
void taskLED1() {
```

```
    digitalWrite(LED1, HIGH);
```

```
    delay(1000);
```

```
    digitalWrite(LED1, LOW);
```

```
    delay(500);
```

```
}
```

```
void taskLED2() {
```

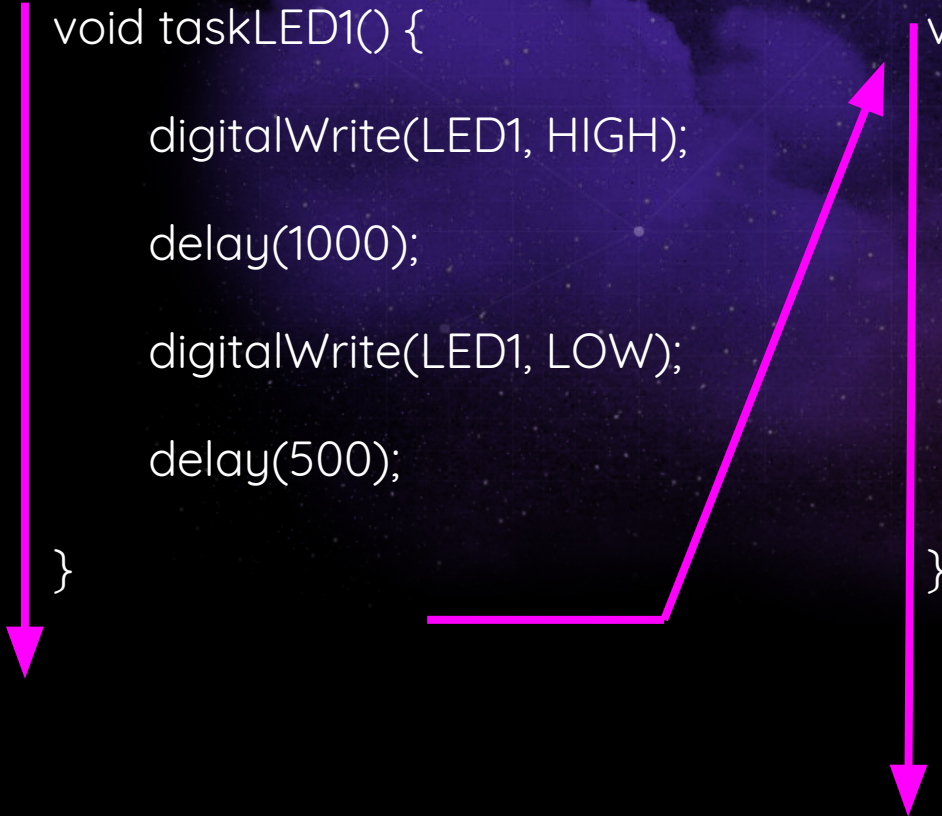
```
    digitalWrite(LED2, HIGH);
```

```
    delay(700);
```

```
    digitalWrite(LED2, LOW);
```

```
    delay(300);
```

```
}
```



# Multitasking with library protothreads

Multitasking คือ การทำให้มองเห็น**เสมือน**ว่าโปรแกรม  
สามารถประมวลผลหลายๆคำสั่งได้พร้อมกัน



# Code : Protothread(1)

```
#include <pt.h>
```

```
#define PT_DELAY(pt, ms, ts)\
```

```
    ts = millis();\
```

```
    PT_WAIT_WHILE(pt, millis()-ts < (ms));
```

```
#define LED1 11
```

```
#define LED2 12
```

```
struct pt pt_taskLED1;
```

```
struct pt pt_taskLED2;
```

# Code : Protothread (2)

```
PT_THREAD(taskLED1(struct pt* pt)) {  
  
    static uint32_t ts;  
  
    PT_BEGIN(pt);  
  
    while(1){  
  
        digitalWrite(LED1,HIGH);  
  
        PT_DELAY(pt, 1000, ts);  
  
        digitalWrite(LED1,LOW);  
  
        PT_DELAY(pt, 500, ts);    }  
  
    PT_END(pt); }
```

```
PT_THREAD(taskLED2(struct pt* pt)) {  
  
    static uint32_t ts;  
  
    PT_BEGIN(pt);  
  
    while(1){  
  
        digitalWrite(LED2,HIGH);  
  
        PT_DELAY(pt, 700, ts);  
  
        digitalWrite(LED2,LOW);  
  
        PT_DELAY(pt, 300, ts);    }  
  
    PT_END(pt); }
```



# Code : Protothread (3)

```
void setup(){  
    pinMode(LED1, OUTPUT);  
    pinMode(LED2, OUTPUT);  
    PT_INIT(&pt_taskLED1);  
    PT_INIT(&pt_taskLED2);  
}
```

# Code : Protothread (4)

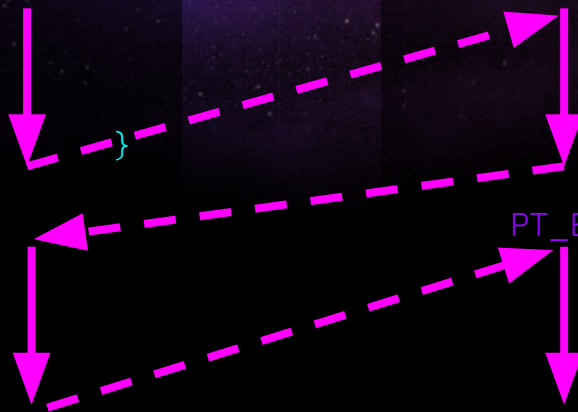
```
void loop(){  
    taskLED1(&pt_taskLED1);  
    taskLED2(&pt_taskLED2);  
}
```



# Process

```
PT_THREAD(taskLED1(struct pt* pt)) {  
    static uint32_t ts;  
  
    PT_BEGIN(pt);  
  
    while(1){  
        digitalWrite(LED1,HIGH);  
  
        PT_DELAY(pt, 1000, ts);  
  
        digitalWrite(LED1,LOW);  
  
        PT_DELAY(pt, 500, ts);  
    }  
  
    PT_END(pt); }
```

```
PT_THREAD(taskLED2(struct pt* pt)) {  
    static uint32_t ts;  
  
    PT_BEGIN(pt);  
  
    while(1){  
        digitalWrite(LED2,HIGH);  
  
        PT_DELAY(pt, 700, ts);  
  
        digitalWrite(LED2,LOW);  
  
        PT_DELAY(pt, 300, ts);  
    }  
  
    PT_END(pt); }
```



# Challenge : Practice Protothreads

LED1 จะสว่าง 1000 ms. และดับ 500 ms.

LED2 จะสว่าง 700 ms. และดับ 300 ms.

อ่านค่าจาก Temp Sensor และ แสดงที่ LCD อัปเดตค่าทุกๆ 1000 ms.

เมื่อกดค้าง switch จะทำให้ buzzer ร้อง และเมื่อปล่อยจะหยุดร้อง

Hint: ค่อยๆแก้ทีละ function



# Documents : Other sensors

<https://goo.gl/majPMn>