

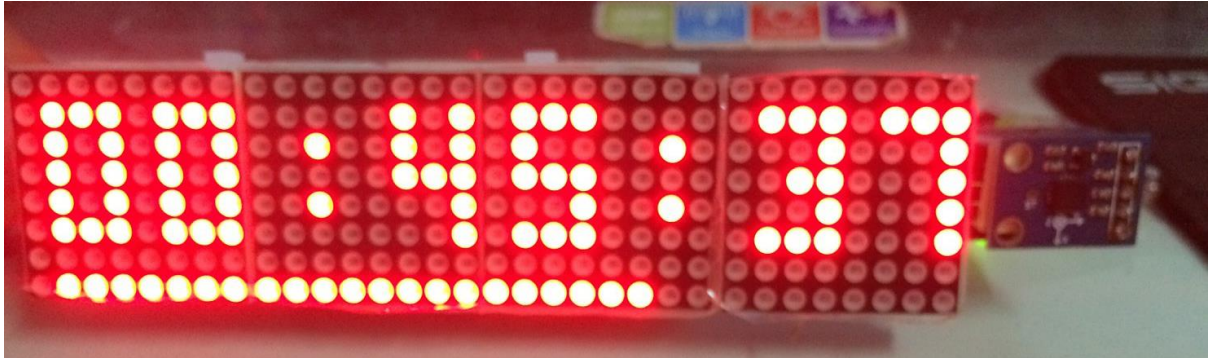
# MINI CLOCK

PRESENTED BY

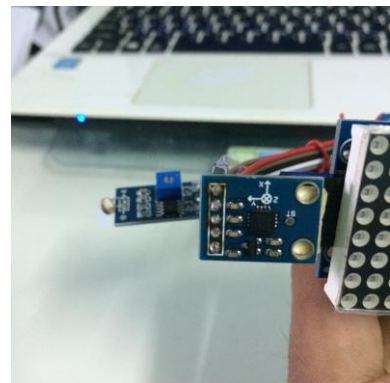
พี่ป๋าวมาแว้ววว

62010882      62010966

# CONCEPTION DESIGN



นาฬิกาของเราออกแบบมาให้มีการทำงานเหมือนนาฬิกาข้อมือ คือจะมีฟังก์ชันการแสดงผลเวลาในรูปแบบ 24 ชั่วโมง ในการแสดงผลทั่วไปจะแสดงผลเวลา โดยแสดง ชั่วโมง นาที และวินาที การแสดงผลวันที่จะเป็นการแสดงผลวันที่ เดือน ปีคริสต์ศักราช ระบบการตั้งเวลา การตั้งวันที่ นาฬิกาจับเวลา และนาฬิกาปลุก



นาฬิกาของเราจะมีทั้งหมด 3 ปุ่ม ปุ่มแรกเป็นปุ่มเปลี่ยนโหมด ปุ่มที่สองจะเป็นปุ่มสำหรับเลื่อนตำแหน่งที่จะทำการตั้งค่า สำหรับปุ่มสุดท้ายปุ่มที่สามจะเป็นปุ่มสำหรับปรับค่าต่างๆ

แต่ในโหมดจับเวลาปุ่มที่สองจะทำหน้าที่ในการเริ่ม หยุด และจับเวลาต่อ ส่วนปุ่มที่สามจะทำหน้าที่ reset ให้ค่ากลับมาเป็น 0

และนาฬิกาของเราจะมีฟังก์ชันพิเศษเพิ่มมาอีก 2 อย่างคือ การปรับค่าความสว่างอัตโนมัติตามความเข้มของแสงในธรรมชาติ และอีกฟังก์ชันคือการแสดงผลทั้งสองด้านเมื่อเกิดกลับด้านของนาฬิกา

# HOW TO USE

สำหรับการทำงานเริ่มต้นจะให้แสดงผลเวลา ณ วินาทีที่ 25 จะให้เปลี่ยนเป็นแสดงผลวันที่ แล้วกลับมาแสดงผลเวลาเมื่อวินาทีที่ 30

นาฬิกาของเราจะมีทั้งหมด 3 ปุ่ม ปุ่มแรกเป็นปุ่มเปลี่ยนโหมด ปุ่มที่สองจะเป็นปุ่มสำหรับเลื่อนตำแหน่งที่จะทำการตั้งค่า สำหรับปุ่มสุดท้ายปุ่มที่สามจะเป็นปุ่มสำหรับปรับค่าต่างๆ

ในช่วงที่แสดงเวลาแล้วเราก็สามารถกดปุ่มที่ 1 เพื่อเปลี่ยนไปดูเวลา

กดอีกครั้งจะเปลี่ยนเป็นตั้งเวลา สามารถกดปุ่มที่ 2 เพื่อเปลี่ยนไปตั้งค่าในชั่วโมง นาที หรือวินาทีได้ ในการปรับเปลี่ยนเวลาสามารถทำได้โดยการกดปุ่มที่ 3 นั้นเอง

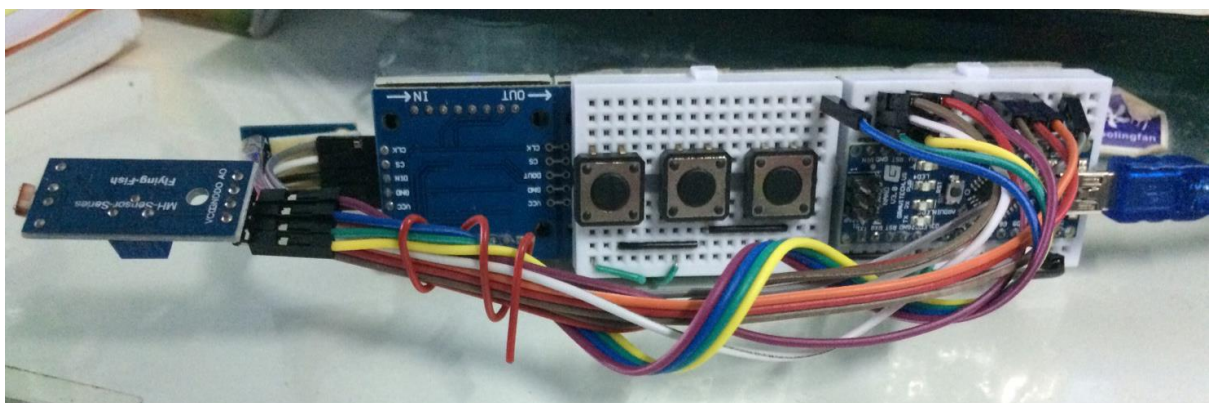
โหมดต่อไปเป็นโหมดตั้งวันที่ ทุกอย่างทำเหมือนตั้งเวลาเลย แต่ควรไปตั้งค่าเดือนก่อน เพราะบางเดือนมี 28 วัน บางเดือนมี 30 วัน และบางเดือนมี 31 วัน

โหมดต่อไปจะเป็นนาฬิกาจับเวลาโหมดนี้จะแตกต่างออกไป คือ ปุ่มที่ 2 จะไม่ใช้การเลื่อนตำแหน่งเพื่อตั้งเวลาแล้วจะเป็นการกดเพื่อเริ่มจับเวลา กดเพื่อหยุดเวลา และกดเพื่อจับเวลาต่อ ส่วนปุ่มที่ 3 จะทำหน้าที่ในการตั้งค่าให้เวลากลับมาเป็น 0

โหมดต่อไปจะเป็นการตั้งนาฬิกาปลุก โหมดนี้ทำเหมือนตั้งเวลาและวันที่เลยแต่จะมีการตั้งแค่ ชั่วโมง กับนาทีเท่านั้น

ต่อไปเป็นฟังก์ชันพิเศษคือการปรับความสว่างของ LED ให้มีความสว่างที่เหมาะสมกับช่วงเวลา เช่นถ้าตอนกลางคืนทุกอย่างจะดำมืดถ้ามีแสงสว่างจาก LED มารบกวนจะทำให้นอนไม่หลับจึงมีการออกแบบมาให้หลอด LED สว่างแปรผันตรงกับความเข้มแสงในธรรมชาติ

และอีกฟังก์ชันคือไม่ว่าจะตั้งนาฬิกาฝั่งไหนก็สามารถดูเวลาได้ (ยกเว้นแนวตั้ง) ฟังก์ชันนี้จะใช้ตัว sensor วัดความเอียงเข้ามาช่วยทำให้นาฬิกาปรับทิศทางการแสดงผลได้เหมาะสมกับการวาง



# CODE PART

```
int input1;
int input2;
int input3;
int mode = 0;
int change = 0;
void BUTTONPRESSED() {
    input1 = !digitalRead(SET_BUTTON);
    mode = (mode+input1)%6;

    if(mode != change) input2 = 0;

    if(mode == 4 || mode == 5) input2 = (input2+!digitalRead(CHANGE_BUTTON))%2;
    else input2 = (input2+!digitalRead(CHANGE_BUTTON))%3;

    input3 = !digitalRead(STOP_BUTTON);

    if(input1)
        clear_display();

    ROTATE();
    LIGHT();
    ALARM(input1);

    switch(mode)
    {
        case 0: if(TIME[2].num ==25 or TIME[2].num == 31)
                test_all_led();
                if(TIME[2].num >=25 and TIME[2].num <= 30) {
                    DATE();
                } else SHOW(); break;
        case 1: DATE(); break;
        case 2:
                if(input2 == 2 and input3 == 1)
                    TIME[input2].num = 0;
                else if(input3 == 1)
                    TIME[input2].num++;
                setTime(input2); break;

        case 3: if(input3 == 1) CAL.num[3-input2-1]++;
                setDate(input2); break;
        case 4: STOPWATCH(input2, input3); break;
        case 5: if(input3 == 1) alarm[input2].num++;
                setALARM(input2); break;
        default: break;
    }

    change = mode;
}
```

## ฟังก์ชัน BUTTONPRESSED

สำหรับฟังก์ชันนี้จะเป็นการตรวจสอบการ update การกดปุ่มของแต่ละปุ่ม แต่ละปุ่ม ทำหน้าที่แตกต่างกัน จะมีการรับค่า 0 และ 1 มาใช้เพื่อไปใช้ในการดำเนินการต่างๆ

Input1 ใช้สำหรับการเปลี่ยนโหมด

Input2 ใช้สำหรับการเลื่อนตำแหน่งในการ ตั้งค่า

Input3 ใช้สำหรับเพิ่มค่าต่างๆ

switch(mode)

{

case 0 :

```
case 0: if(TIME[2].num ==25 or TIME[2].num == 31)
    test_all_led();
    if(TIME[2].num >=25 and TIME[2].num <= 30) {
        DATE();
    } else SHOW(); break;
```

<- สำหรับการทำงานเริ่มต้นจะให้  
แสดงผลเวลา ณ วินาทีที่ 25 จะให้  
เปลี่ยนเป็นแสดงผลวันที่แล้ว  
กลับมาแสดงผลเวลาเมื่อวินาทีที่  
30

```
void SHOW() {
    ROTATE();

    for(int k = 0; k<3 ; ++k) {
        TIME[k].CLOCK = "";
        keep = TIME[k].num;
        for (int i = 0; i < 2; ++i)
        {
            TIME[k].CLOCK += keep / 10;
            keep = keep % 10 * 10;
        }
    }

    line(y+6);

    for(int m = 0; m < 3; m++) {
        for (int k = 0; k < 2; ++k) {
            print_tiny_char((i * 4)+1, y, TIME[m].CLOCK[k]);
            i++;
        }
        if(m!=2 && TIME[2].num!=0)
            print_tiny_char((i * 4)+1, y, ':');
        i++;
    }
}
```

การประกาศตัวแปรของเวลา

```
typedef struct State {
    String CLOCK;
    unsigned long int num;
} SType;
```

```
SType TIME[3] = { {"",23}, {"",59}, {"",0} };
SType stopwatch[3] = { {"",0}, {"",0}, {"",0} };
SType alarm[2] = { {"",0}, {"",0}};
```

ฟังก์ชัน SHOW

สำหรับฟังก์ชันนี้จะเป็นการแสดงผลเวลาในรูปแบบ ชั่วโมง นาที และวินาที

จะมีการนำค่าตัวเลขมาแปลงให้เป็นตัวอักษร แล้วส่งค่าตัวอักษรไปทำการแสดงผลบนหน้าจอ

case 1 :

```
void DATE() {
    ROTATE();

    for(int i=1; i<=30; ++i)
        plot(i , y+6, 1);

    CAL.num[2] = CAL.num[2] % CAL.numDate[CAL.num[1]];
    CAL.num[1] = CAL.num[1]%12;
    CAL.num[0] = CAL.num[0]%100;

    int k=0;
    print_tiny_char((i * 4)+count, y, (CAL.num[2]+1)/10 + '0'); i++;
    print_tiny_char((i * 4)+count, y, (CAL.num[2]+1)%10 + '0'); i++;
    while(CAL.month[CAL.num[1]][k]) {
        print_tiny_char((i * 4)+count+1, y, CAL.month[CAL.num[1]][k]); i++; k++;
    }
    print_tiny_char((i * 4)+count+1, y, '.');
    print_tiny_char((i * 4)+count+3, y, CAL.num[0]/10 + '0'); i++;
    print_tiny_char((i * 4)+count+3, y, CAL.num[0]%10 + '0');
}
```

การประกาศตัวแปลของวันที่

```
typedef struct SCalender {
    int num[3] = { 1, 1, 1}; //yy/mm/dd
    int numDate[12] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
    int numMonth = 10;
    String month[12] = { "JAN", "FEB", "MAR", "APR", "MAY", "JUN", "JUL", "AUG", "SEP", "OCT", "NOV", "DEC"};
    int numYear = 19;
} cal;

cal CAL;
```

ฟังก์ชัน DATE

สำหรับฟังก์ชันนี้จะเป็นการแสดงผลเวลาในรูปแบบ วัน เดือน ปี

จะมีการนำค่าตัวเลขมาแปลงให้เป็นตัวอักษร แล้วส่งค่าตัวอักษรไปทำการแสดงผลบนหน้าจอ

case 2 :

```
case 2:
    if(input2 == 2 and input3 == 1)
        TIME[input2].num = 0;
    else if(input3 == 1)
        TIME[input2].num++;
    setTime(input2); break;

void setTime(int mode) {
    BLINK(mode);
    delay(100);
    SHOW();
}
```

Case นี้ จะมีการนำค่า input2 มาใช้ในการเลื่อนตำแหน่ง และใช้ input3 ในการเพิ่มค่า และ reset ค่าเมื่อทำการ set ค่า วินาที

ฟังก์ชัน setTime(int mode)

สำหรับฟังก์ชันนี้ จะมีการนำฟังก์ชัน BLINK ที่เป็นฟังก์ชันสำหรับการกระพริบในตำแหน่งที่ต้องการตั้งค่า มาใช้ในการแสดงผล

case 3 :

```
case 3: if(input3 == 1) CAL.num[3-input2-1]++;
        setDate(input2); break;

void setDate(int mode) {
    BLINK(mode);
    delay(100);
    DATE();
}
```

Case นี้ จะมีการนำค่า input2 มาใช้ในการเลื่อนตำแหน่ง และใช้ input3 ในการเพิ่มค่า

ฟังก์ชัน setDate(int mode)

สำหรับฟังก์ชันนี้ จะมีการนำฟังก์ชัน BLINK ที่เป็นฟังก์ชันสำหรับการกระพริบในตำแหน่งที่ต้องการตั้งค่า มาใช้ในการแสดงผล



case 4 :        `case 4: STOPWATCH(input2, input3); break;`

```
int pause; bool checkcontinue = false; bool checkreset = true; int stopstart;
void STOPWATCH(int start, int reset) {
    ROTATE();
    if(!checkcontinue and start) {
        stopstart = TIME[2].num;
        checkreset = false;
        checkcontinue = true;
        stopwatch[2].num = pause;
    }
    else if(checkcontinue and start) {
        pause = stopwatch[2].num;
        checkcontinue = false;
    }
    else if(!checkreset and reset) {
        checkreset = true;
        checkcontinue = false;
        for(int i=0; i<3; ++i)
            stopwatch[i].num = 0;
        pause = 0;
    }
    else if(checkreset)
        stopwatch[2].num = 0;
    else if(!checkcontinue )
        stopwatch[2].num = pause;
    else if(checkcontinue ) {
        if(TIME[2].num == 0 and stopstart == 59)
            stopstart = -1;
        stopwatch[2].num += TIME[2].num - stopstart;
        stopstart = TIME[2].num;
    }
}
```

Code ส่วนนี้เป็นการเพิ่มค่านำตัวแปร Boolean มาใช้ในการตรวจสอบ state ว่ากำลัง จับเวลา หยุดเวลา หรือ reset เวลา

เมื่อทำการหยุดเวลาจะมีตัวแปร pause มารับเวลาที่จับมาและเมื่อทำการจับเวลาต่อจะนำมาจากตัวแปร pause มานับเวลาต่อ

```
if(stopwatch[2].num == 60)
    stopwatch[1].num++;
if(stopwatch[1].num == 60)
    stopwatch[0].num++;
```

```
stopwatch[0].num%=24;
stopwatch[1].num%=60;
stopwatch[2].num%=60;
```

```
for(int k = 0; k<3 ; ++k) {
    stopwatch[k].CLOCK = "";
    keep = stopwatch[k].num;
    for (int i = 0; i < 2; ++i)
    {
        stopwatch[k].CLOCK += keep / 10;
        keep = keep % 10 * 10;
    }
}
```

```
line(y+6);
```

```
for(int m = 0; m < 3; m++) {
    for (int k = 0; k < 2; ++k) {
        print_tiny_char((i * 4)+1, y, stopwatch[m].CLOCK[k]);
        i++;
    }
    if(m!=2 && stopwatch[2].num!=0)
        print_tiny_char((i * 4)+1, y, ':');
    i++;
}
```

Code ส่วนนี้เป็นการเพิ่มค่าและตั้งค่า นาฬิกา ชั่วโมง

Code ส่วนนี้จะมีการนำค่าตัวเลขมาแปลงให้เป็นตัวอักษร แล้วส่งค่าตัวอักษรไปทำการแสดงผลบนหน้าจอ



case 5 :

```
case 5: if(input3 == 1) alarm[input2].num++;  
      setALARM(input2); break;
```

Case นี้ จะมีการนำค่า input2 มาใช้ในการเลื่อนตำแหน่ง และใช้ input3 ในการเพิ่มค่า

```
void setALARM(int mode) {  
    BLINK(mode);  
    delay(100);  
    ROTATE();  
    |  
    alarm[0].num%=24;  
    alarm[1].num%=60;  
    for(int k = 0; k<2 ; ++k) {  
        alarm[k].CLOCK = "";  
        keep = alarm[k].num;  
        for (int i = 0; i < 2; ++i)  
        {  
            alarm[k].CLOCK += keep / 10;  
            keep = keep % 10 * 10;  
        }  
    }  
    for(int i=1; i<=30; ++i)  
        plot(i , y+6, 1);  
  
    for(int m = 0; m < 2; m++) {  
        for (int k = 0; k < 2; ++k) {  
            print_tiny_char((i * 4)+3, y, alarm[m].CLOCK[k]);  
            i++;  
        }  
        if(m!=1)  
            print_tiny_char((i * 4)+3, y, ':');  
        i++;  
    }  
    print_bell((i * 4), y);  
}
```

## ฟังก์ชัน setAlarm

สำหรับฟังก์ชันนี้จะเป็นการตั้งค่านาฬิกาปลุก จะเหมือนกับการตั้งเวลาแต่จะมีแค่ชั่วโมง กับนาฬิกา  
จะมีการนำค่าตัวเลขมาแปลงให้เป็นตัวอักษร แล้วส่งค่าตัวอักษรไปทำการแสดงผลบนหน้าจอ

## FUNCTION พิเศษที่ใช้ในการแสดงผล

### ฟังก์ชัน ROTATE สำหรับการหมุนพลิกของนาฬิกา

```
int a,b = 1; int rotate = 1; int light; int LDR; int i , count, y;
void ROTATE() {
  x_raw = jitter(A0);

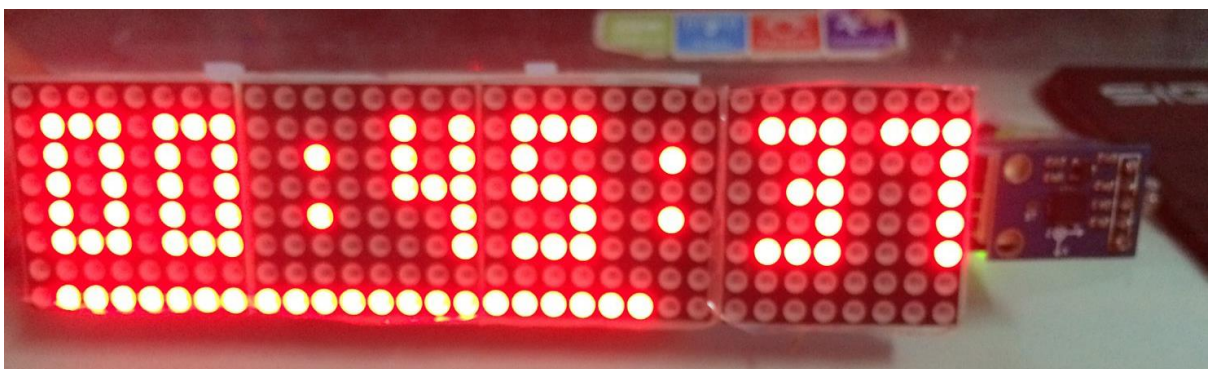
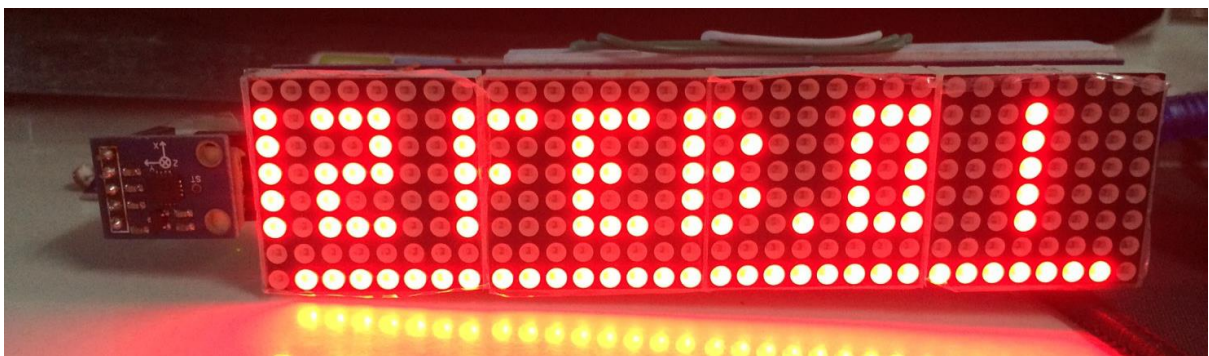
  a = map(x_raw,x_raw_min,x_raw_max,-90,90);
  if( a / b < 0 ) {
    test_all_led();
    rotate *= -1;
  }
  b = a;

  if( rotate == -1 ) { i=0; count = 1; y = 1; }
  else { i=-8; count = -1; y = -6; }

  Serial.print(x_raw); Serial.print(" "); Serial.println(analogRead(A5));
}
```

จะเป็นการ set ค่าให้มุมมีค่าตั้งแต่ -90 ถึง 90 องศา เมื่อมีการหมุนจะนำค่ามุมปัจจุบัน a ไปหารกับ b ค่ามุมก่อนหน้าถ้าเกิดน้อยกว่า 0 แสดงว่ามีการพลิกเกิดขึ้น

แล้วจะมีการเก็บค่า i count และ y เพื่อนำไปใช้ในการแสดงผลบน dot matrix



## ฟังก์ชัน LIGHT สำหรับการปรับความสว่างของ LED

```
void LIGHT() {  
  light = map(analogRead(A5), 100, 1000, 1, 10);  
  for (int address = 0; address < 4; address++) {  
    lc.setIntensity(address, 10 - (analogRead(A5)/100));  
  }  
}
```

จะเป็นการ set ค่าความสว่างของ LED ตามค่าที่วัดได้แล้วนำแปลงค่าเพื่อใช้ในการแสดงผล

## ฟังก์ชัน BLINK สำหรับการกระพริบเพื่อใช้แสดงผลเวลาตั้งค่าต่างๆ

```
void BLINK(int mode) {  
  ROTATE();  
  
  int k=0, m;  
  switch(mode) {  
    case 0: k=4*i; m=k+9; break;  
    case 1: i+=2; k=4*i+1; m=k+13; break;  
    case 2: i+=6; k=4*i-1; m=k+8; break;  
  }  
  while(k!=m) {  
    int n = 0;  
    while(n<5) {  
      plot(abs(k+1) , abs(y+n) , 0); n++;  
    }  
    k++;  
  }  
}
```

สำหรับฟังก์ชันนี้จะเป็นการกระพริบ mode คือตำแหน่งที่จะทำการกระพริบ

