

```
/*
```

```
BH1750モジュールを使った照度計  
表示は小型LCD(AQM0802)を使用
```

```
-----
```

```
12.Sep.2016 S.S
```

```
ATtiny85接続:
```

```
* ATtiny Pin 1 = (RESET) N/U  
* ATtiny Pin 3 = (D4) to LED1  
* ATtiny Pin 5 = SDA on DS1621 & GPIO  
* ATtiny Pin 7 = SCK on DS1621 & GPIO  
(2.7-5.5V)今回は3.3V  
ATtiny Pin 2 = (D3) N/U  
ATtiny Pin 4 = GND  
ATtiny Pin 6 = (D1) N/U  
ATtiny Pin 8 = VCC
```

```
BH1750接続:
```

```
VCC-3.3v  
GND-GND  
SCL-SCL(analog pin 5)  
SDA-SDA(analog pin 4)  
ADD-NC or GND
```

```
*/
```

```
#include <TinyWireM.h>  
#include <ST7032TinyWireM.h>
```

```
#define LED1 3 // ATtiny Pin 2
```

```
byte buff[2];  
int Adresse = 0x23; // BH1750のI2Cアドレス  
float lux = 0.0;
```

```
ST7032 lcd;
```

```
void setup()
```

```
{  
  pinMode( LED1, OUTPUT ); // ステータスLED  
  Blink( LED1, 2 ); // 起動表示  
  
  TinyWireM.begin();  
  lcd.begin(8, 2); // AQM0802なので、8文字×2行  
  lcd.setContrast(30); // 3.3V時のコントラストはこの値  
  // Print a message to the LCD.  
  lcd.print("Illumine"); // 1行目にメッセージを表示  
}
```

```
void loop()
```

```

{

    lcd.setCursor(0, 1);          // 照度は2行目に表示

// Check I2C Adresse
TinyWireM.beginTransmission(Adresse);
if(TinyWireM.endTransmission()!=0)
{
    digitalWrite( LED1,HIGH );    // I2Cアドレス設定失敗
} else {
    digitalWrite( LED1,LOW );     // I2Cアドレス設定OK
}

// 照度センサ設定
TinyWireM.beginTransmission(Adresse);
TinyWireM.send(0x21);            // One Time H-Resolution Mode2(0.5 lux
読み取り後Power down)
TinyWireM.endTransmission();

byte i=0;

TinyWireM.requestFrom(Adresse, 2);

while(TinyWireM.available())
{
    buff[i] = TinyWireM.receive();
    i++;
}

lux = ((buff[0]<<8)|buff[1]) / 1.2; // 読み取った値を16bitに変換
// 1.2はROHMの仕様書(P.17)にある補正值

if(lux < 0) {
    lcd.print("Overflow");
} else {
    lcd.print(lux); lcd.print("Lx   ");
}

delay(1000);
}

//
// LEDをブリンク
// ブリンク回数は引数から得る
//
void Blink(byte led, byte times){
    for (byte i=0; i< times; i++){

```

```
digitalWrite(led,HIGH);  
delay ( 400 );  
digitalWrite(led,LOW);  
delay ( 200 );  
}  
}
```