

**Ce tp a pour but de créer une librairie pour utiliser un capteur de lumière I2C de type BH1750.**

## 1. Câblage du circuit et vérification de l'adresse

Câbler le module BH1750 *en faisant bien attention à la polarité*.

Utiliser le programme ic2Scanner et vérifier que l'adresse trouvée correspond bien à l'adresse indiqué dans la documentation du circuit.

## 2. Une première lecture de la luminosité

Décrire *très précisément* la suite de transactions I2C nécessaires pour :

- Démarrer le circuit (PowerOn)
- Faire un reset du circuit
- Placer le circuit en mode continu haute résolution
- Lire une mesure

En utilisant la librairie Wire, écrire un programme qui exécute les actions ci dessus pour lire et afficher la luminosité.

## 3. Fonctions C pour lire et écrire sur le bus I2C

Comme il y a beaucoup d'actions identiques, il est judicieux d'écrire un fonction pour écrire un octet et un fonction pour lire les deux octets de la réponse.

Les prototypes seront :

```
void i2cWrite8( uint8_t address , uint8_t data ) ;  
uint16_t i2cRead16( uint8_t address ) ;
```

Modifier votre programme pour utiliser ces fonctions.

## 4. Conception d'une librairie pour accéder au circuit BH1750

Le but est de disposer d'une classe C++ qui permet d'accéder facilement au fonctionnalité du circuit sans avoir à connaître les détails de son fonctionnement.

La classe devra au moins comporter les membres suivant :

```
#ifndef BH1750_h
#define BH1750_h

#include "Arduino.h"

#include "Wire.h"

class BH1750 {
public:
    BH1750(); // constructeur (rien à faire mais doit exister!)
    void begin(void); // initialisation du circuit
    void setAddress( uint8_t address = 0x23) ; // fixe l'adresse du
circuit
    void setMode(uint8_t mode ); // change le mode de fonctionnement
    void reset(void); // reset du circuit
    uint16_t getRawLight(void); // lit le résultat brut d'une mesure
private:
    void i2cWrite8( uint8_t data ) ; // écrit un octet
    uint16_t i2cRead16( ) ; // lit 16 bits
    uint8_t address ;
    uint8_t mode ;
};
#endif
```

Ensuite vous pouvez rajouter des méthodes pour utiliser les autres fonctionnalités du circuit et une méthode pour retourner la valeur de la luminosité en lux (conversion fonction du mode).

Vous pouvez partir du programme BH1750\_lib\_ETD fournit.