Bus I2C

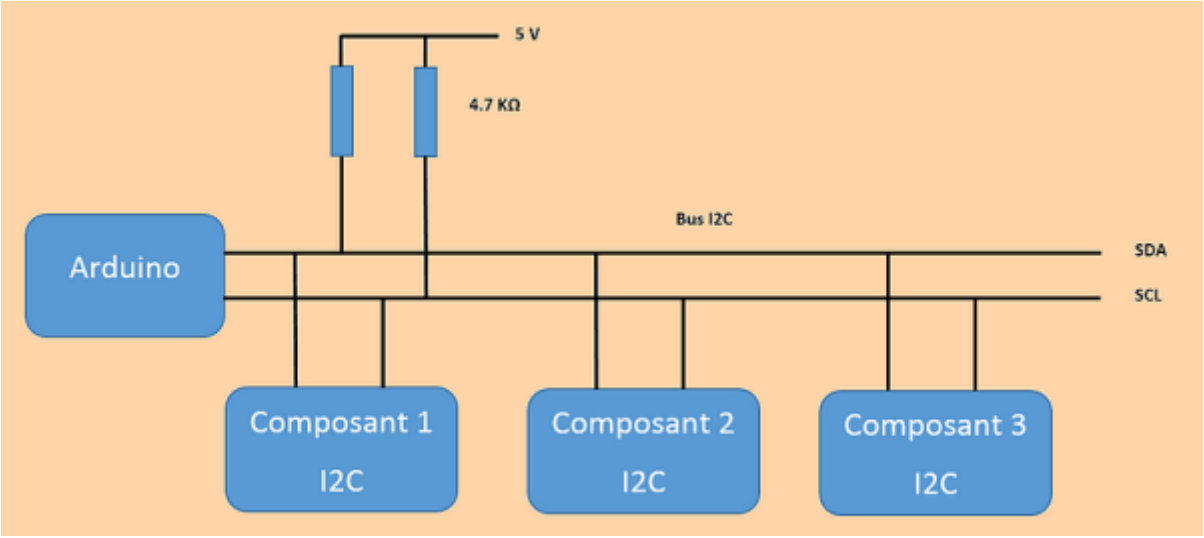


Le protocole I2C

I2C, pour Inter Integrated Circuit , est un protocole de communication crée à la base par Philips en 1982 pour standardiser l'échange de données entre différents circuits intégrés d'un même système. Elle est aujourd'hui assurée par le fabricant NXP, spin-off de l'entreprise.

Ce protocole est basé sur un bus de communication sériel, ce qui signifie qu'un seul câble est utilisé pour le transfert de données.

En pratique, le bus I2C est constitué de deux câbles, un donc pour les données, nommé SDA (Serial Data) et l'autre faisant office d'horloge pour déterminer la fréquence de la communication, nommé SCL (Serial Clock).



Le nombre maximum d'équipements est limité par le nombre d'adresses disponibles, 7 bits

d'adressage et un bit R/W (lecture ou écriture), soit 128 périphériques, mais il dépend également de la capacité (CB) du bus (dont dépend la vitesse maximale du bus).

Il faut savoir que des adresses sont réservées pour diffuser des messages en broadcast et que de nombreuses adresses sont déjà attribuées par les fabricants ce qui limite grandement le nombre d'équipements (une variante d'adressage sur 10 bits existe également).

Temps et vitesses

Il existe cinq vitesses de transmission :

- Standard mode (Sm) » ≤ 100 kbit/s,
- Fast mode (Fm) » ≤ 400 kbit/s,
- Fast plus mode (Fm+) » ≤ 1 Mbit/s,
- High-speed mode (Hs-mode) » ≤ 3,4 Mbit/s,
- Ultra-fast mode (UFm) » ≤ 5 Mbit/s, unidirectionnel uniquement.

Description

Cette librairie vous permet de communiquer avec les composants utilisant le protocole I2C / TWI (communication série sur 2 fils).

Pour plus de détails sur le bus I2C : http://fr.wikipedia.org/wiki/I%C2%B2C

Connexions

Sur la plupart des cartes Arduino :

- la ligne SDA (ligne de données) est sur la broche analogique 4
- la ligne SCL (ligne d'horloge) est sur la broche analogique 5

Sur la carte Arduino Mega :

- SDA est sur la broche numérique 20
- SCL est sur la broche numérique 21

Les fonctions de la librairie Wire

Fonctions d'initialisation

- begin() : initialise communication avec Arduino “maître”
- begin(adresse) : initialise communication avec Arduino “esclave”

Fonctions “mode maître”

- requestFrom(adresse, quantite) : demande de données à un esclave
- beginTransmission(adresse) : débute communication avec un esclave (ouvre stockage données à envoyer avec write)
- endTransmission() : envoi des données vers esclave

- write() : écrit les données à envoyer vers esclave
- available() : test si données disponibles en provenance esclave (cf requestFrom)
- read() : lit les données en provenance de l'esclave

Fonctions “mode esclave”

```
  write() : envoie les données vers le maître après requête
  available() : test si données disponibles en provenance du maître (cf
onReceive)
  read() : lit données en provenance maître
  onReceive(fonction) : définit la fonction à appeler sur réception de
données en provenance du maître
  onRequest(fonction) : définit la fonction à appeler sur requête du maître
```

Fonctions obsolètes

```
  send() - obsolète
  receive() - obsolète
```

Principe d'utilisation

Pour inclure la librairie Wire dans un programme, on ajoutera au début du programme la ligne suivante :

#include <Wire.h> [$[Get Code]]

Pour ce faire, le plus simple est d'utiliser le menu du logiciel Arduino Programme > Importer Librairie > Nom_librairie

Voir également l'instruction #include

Note

Il existe deux versions d'adresses I2C en 7 et en 8 bits. Les adresses 7 bits identifient le composant, les adresses 8 bits déterminent si on écrit ou on lit le composant. La librairie Wire utilise les adresses 7 bits. Si vous avez une fiche technique ou un exemple de code qui utilise des adresses 8 bits, vous devrez éliminer le bit de poids faible (en décalant la valeur de l'adresse d'un bit vers la droite - voir notamment l'opérateur de décalage à droite »), pour obtenir une adresse comprise entre 0 et 127.

Liens utiles

I2C Principes

I2C sur ESP32

Bibliothèque Wire : I2C

Reference Librairie I2C Arduino EN

Plusieurs cartes Arduino Pilotées par Bus I2C

From:
https://chanterie37.fr/fablab37110/ - **Castel'Lab le Fablab MJC de Château-Renault**

Permanent link:
https://chanterie37.fr/fablab37110/doku.php?id=start:arduino:i2c&rev=1607105073

Last update: **2023/01/27 16:08**