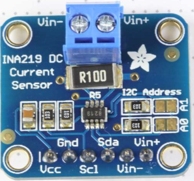
|  |  |  |
| --- | --- | --- |
| SSI | Mesures et traitements des données d’un capteur de courant numérique | TD |

1. Matériel utilisé

Pour faire ce TD, nous mettons en œuvre une carte Arduino UNO et un capteur de courant numérique équipé d’un amplificateur d’instrumentation INA219B.

Le capteur, baptisé INA219B utilise une résistance de shunt (Rsh) comme corps d’épreuve.

Ce capteur fournit les informations de courant, de tension de shunt et de puissance dissipée dans le shunt à une carte Arduino (ou compatible) via une interface i2c.

**Rappel sur principe de la mesure de l’intensité avec une R sh**

R Charge

Rsh

V Alim

U Charge

Ush

**I**

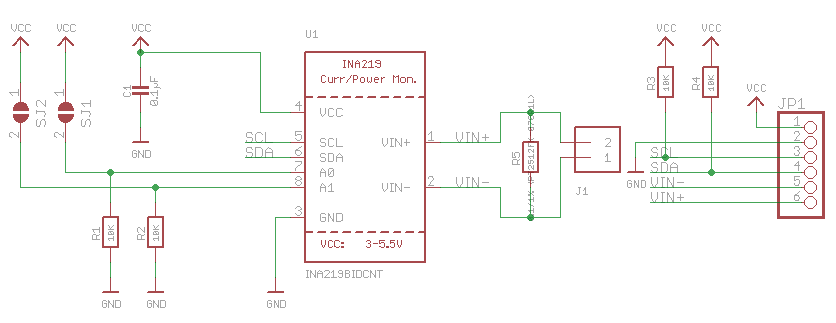
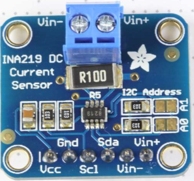
Un shunt est une résistance calibrée et connue utilisée pour la mesure de courants forts. Pour ce faire on mesure la tension à ses bornes à l'aide d'un [voltmètre](https://fr.wikipedia.org/wiki/Voltm%C3%A8tre) branché en parallèle, l'utilisation de la [loi d'Ohm](https://fr.wikipedia.org/wiki/Loi_d%27Ohm) permet de déduire le courant traversant le shunt.

**I** = Ush / Rsh

Afin de limiter la chute de tension provoquée par son utilisation et afin de limiter les pertes par [effet joule](https://fr.wikipedia.org/wiki/Effet_joule) dans le shunt, il doit avoir une valeur de résistance très faible, de l'ordre de quelques [m](https://fr.wikipedia.org/wiki/Milli)[Ω](https://fr.wikipedia.org/wiki/Ohm_(unit%C3%A9)). Toutefois elle ne doit pas non plus être trop faible afin que la tension puisse être transmise et mesurée sans perturbation auxiliaire.

Les shunts sont employés pour la mesure de [courants continus](https://fr.wikipedia.org/wiki/Courant_continu) et [alternatifs](https://fr.wikipedia.org/wiki/Courant_alternatif) qu'ils soient de basses ou de hautes fréquences.

**Schéma électrique du capteur**



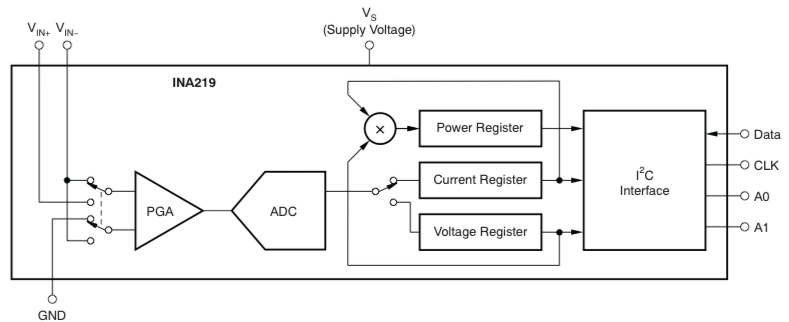
Bornier de raccordement au circuit à mesurer : Vin+ et Vin-

Rsh = 0,1Ω

Bornier de raccordement :

* Alim : Vcc & Gnd
* I2c : scl & sda
* Vin+ et Vin-

Note : Ush = Vin+ - Vin-



CAN ajustable

(12 bits par défaut)

AOP

Gain ajustable

**Zoom sur l’amplificateur INA219B**

L’échelle pleine de mesure de Ush (aux bornes de Rsh) est ajustée par le gain du PGA, la résolution du CAN est ainsi modifiable. La valeur de Ush numérisée est transmise au registre de tension ou/et au registre de courant pour être traitée puis les résultats communiqués par i2c.

Exemple :

* Par défaut, le gain est de 320mV (PGA8). L’échelle pleine de Ush (analogique) est 0-320 mV. La plage de mesure de l’intensité est donc de 0-3,2 A

I max = Ush max / Rsh = 0,32 / 0,1 = 3,2 A

* La précision de mesure de l’intensité est donc de 0,8 mA

q = Upe / (2Nb – 1) = 3,2 / (212 -1) = 0,78 mA/LSB

1. Consultez les différentes valeurs de gain (PGA,) page 3 du datasheet de INA219B, et complétez le tableau ci-dessous faisant la synthèse des différentes configurations possible. (1 pt)

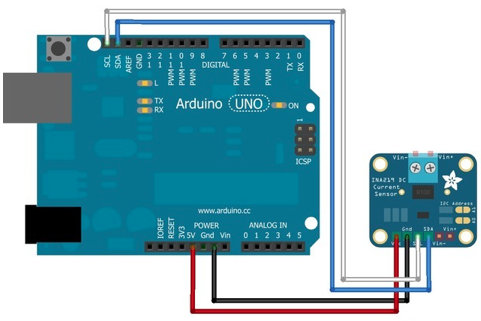
|  |  |  |  |
| --- | --- | --- | --- |
| Gain (PGA) | Echelle pleine de Ush en mV | I max mesurable en A | Précision de mesure de l’intensité en mA/LSB |
| PGA8 | 0 - 320 | 3,2 | 0,8 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Quel est l’intérêt de pouvoir ajuster le gain ? (2 pts)
2. Mise en œuvre

L’étude a pour but de faire découvrir :

* La mise en œuvre de ce type de capteur,
* Le décodage d’une trame i2c
* Le traitement des données transmises par bus i2c et les relations entre programme ino et les bibliothèques rattachées.

**Câblage**



R Charge

V Alim

I

**Expérimentation**

La charge R est connue, la tension d’alimentation est ajustée afin d’obtenir un courant d’environ 300mA. On souhaite mesurer avec précision ce courant à l’aide du capteur.

A l’aide d’un oscilloscope, les trames échangées entre le capteur et l’Arduino sont relevées.

* 1. Décodage de la trame i2c



**Séquence 1**

**Séquence 2**

Ci-contre les relevés des trames échangées entre le capteur et l’Arduino lors de la mesure.

**En bleu**(Var) : la broche SCL ou Clk de l’i2c.

**En rouge** (EA1) : la broche SDA ou Data de l’i2c.

Il est possible de différencier deux séquences distinctes.

1. En vous aidant de votre cours et du document « Protocole I2C.docx », décodez la trame du document « Trame I2C.docx ». Puis complétez le résumé dans les tableaux ci-dessous. (6 pts)

**Séquence 1:**

* **Partie 1:** Timing Diagram for **Write** Word Format

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Trame 1 | Trame 2 | Trame 3 | Trame 4 |
|  | Adresse esclave | Pointeur registre | MSByte Data | LSByte Data |
| Binaire | 1000000/W | 00000101 |  |  |
| Hexa | 0x40 |  |  |  |

* **Partie 2:** Typical Register Pointer Set

|  |  |  |
| --- | --- | --- |
|  | Trame 1 | Trame 2 |
|  | Adresse esclave | Pointeur registre |
| Binaire |  |  |
| Hexa |  |  |

**Séquence 2:**

* Timing Diagram for **Read** Word Format

|  |  |  |  |
| --- | --- | --- | --- |
|  | Trame 1 | Trame 2 | Trame 3 |
|  | Adresse esclave | MSByte Data | LSByte Data |
| Binaire | 1000000/R |  |  |
| Hexa | 0x40 |  |  |

* 1. Lecture de la trame

1. Introduction

Les trames échangées entre le capteur et l’Arduino sont liés aux instructions du programme « courrant\_seul.ino » et des bibliothèques associées à ce capteur (Adafruit\_INA219.cpp et Adafruit\_INA219.h).

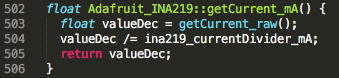
Pour la suite de l’activité, nous utiliserons le logiciel « Sublime Text » pour éditer les codes (en ressources de l’activité). Ce logiciel nous permettra d’utiliser l’outil de « recherche texte » (ctrl + f).

1. Dans le programme « courant\_seul.ino », quelle instruction demande au capteur de fournir l’intensité en mA ? (0.5 pt)

Cette instruction est traitée dans la bibliothèque « Adafruit\_INA219.cpp ».

1. getCurrent\_mA

Faite une recherche de l’instruction « getCurrent\_mA() » dans la bibliothèque « Adafruit\_INA219.cpp »



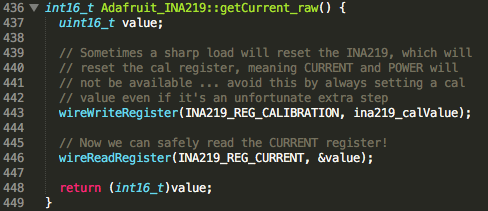
503 : La valeur brute retournée par le sous-programme « getCurrent\_raw() » sera transformée en valeur brute décimale

504 : La valeur brute décimale est divisée par « ina219\_currentDivider\_mA »

505 : Valeur en mA est retournée au programme « .ino » pour traitement

1. Quelle est la valeur du coefficient « ina219\_currentDivider\_mA » ? (1 pt)
2. getCurrent\_raw

Faite une recherche de l’instruction « getCurrent\_raw() » dans la bibliothèque « Adafruit\_INA219.cpp »



437 : Création de la variable à transmettre sous forme brute

443 : Le maître (Arduino) écrit (en i2c) à l’esclave (le capteur) dans son registre « Calibration » la valeur « ina219\_calValue » : il s’agit de la **Séquence1-Partie1** de la trame i2c

446 : Le maître (Arduino) lit le registre « Current » de l’esclave et stocke la valeur dans la variable « value » : le souhait du maître de lire le registre « Current » est **Séquence1-Partie2** de la trame i2c, la réponse de l’esclave est la **Séquence2**

448 : Le sous-programme « getCurrent\_raw() » retourne au sous-programme « getCurrent\_mA() » la variable lue.

1. Pourquoi est-il nécessaire de « calibrer » le capteur avant la mesure ? (2 pts)

Faire une recherche dans la bibliothèque « Adafruit\_INA219.h » du pointeur de registre « calibration » (INA219\_REG\_CALIBRATION) et relever son numéro en hexadécimale.

1. Est-ce que cette valeur hexadécimale correspond à votre trame 2 Séquence1-Partie1 ? Oui ou Non (0.5 pt)

Faire une recherche dans la bibliothèque « Adafruit\_INA219.cpp » de la valeur décimale de la calibration (ina219\_calValue)

1. Est-ce que cette valeur décimale correspond à votre trame 3 et 4 Séquence1-Partie1 ? Oui ou Non (0.5 pt)
2. Quelle est la résolution (en bits) du CAN ainsi programmé ? (1 pt)

Faire une recherche dans la bibliothèque « Adafruit\_INA219.h » du pointeur de registre « Current » (INA219\_REG\_CURRENT) et relever son numéro en hexadécimale.

1. Est-ce que cette valeur hexadécimale correspond à votre trame 2 Séquence1-Partie2 ? Oui ou Non (0.5 pt)

Reste à lire la variable brute « value » qui sera retournée au sous-programme « getCurrent\_mA() », trame 2 et 3 Séquence2

1. Retour à getCurrent\_mA
2. Donnez la valeur de la variable « value » en décimale (valueDec), comme elle sera reçue par le sous-programme « getCurrent\_mA ». (2 pts)
3. Calculez comme le fait le sous-programme « getCurrent\_mA » la valeur de l’intensité du courant ? (3 pts)

Ouf…

1. Bonus

A partir de la ligne 121 de la bibliothèque « Adafruit\_INA219.h » vous trouverez INA219\_CONFIG\_SADCRES\_MASK

1. Quelles est la configuration par défaut ? Expliquez comment se fait la mesure de la tension de shunt dans cette configuration par défaut ? (2 pts)