

# BUS I2C

## M1 Electronique

Haute Ecole Roi Louvain en Hainaut

Ref : Electronique Numérique

Date : 31/10/2013

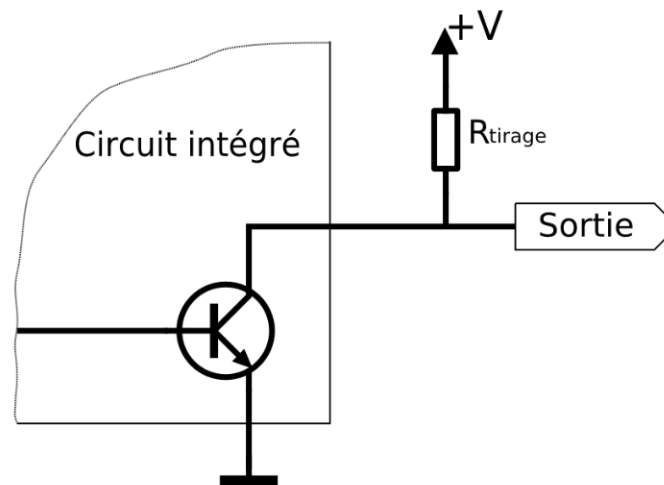
**HELHa** | Haute École  
Louvain en Hainaut

- Le bus I<sup>2</sup>C permet d'établir une liaison série synchrone entre 2 ou plusieurs composants.
- Il a été créé dans le but d'établir des échanges d'informations entre circuits intégrés se trouvant sur une même carte. Son nom, d'ailleurs, traduit son origine : Inter Integrate Circuit, ou I.I.C., ou plus communément I<sup>2</sup>C (I carré C).
- L' I<sup>2</sup>C permettait, à ses débuts, de travailler à des fréquences maximales de 100 Kbits/seconde, vitesses assez rapidement portées à 400 Kbits/seconde. Il existe maintenant des familles de circuits pouvant atteindre des vitesses de 3.4 Mbits/seconde.
- Le bus I<sup>2</sup>C est constitué de 2 uniques lignes bidirectionnelles :
  - La ligne SCL (Serial Clock Line), qui, comme son nom l'indique, véhicule l'horloge de synchronisation;
  - La ligne SDA (Serial DAta line), qui véhicule les bits transmis.

- Il est important de savoir que :
  - La ligne SCL est gérée par le maître (nous verrons que par moment, l'esclave peut prendre provisoirement le contrôle de la ligne).
  - La ligne SDA, à un moment donné, est pilotée par celui qui envoie une information (maître ou esclave).
  - Tous les circuits sont connectés sur ces 2 lignes. Il existe 2 sortes de circuits pouvant être connectés :
    - Les circuits maîtres, qui dirigent le transfert et pilotent l'horloge SCL.
    - Les circuits esclaves, qui subissent l'horloge et répondent aux ordres du maître.
- Une particularité est qu'on peut placer plusieurs maîtres sur le même bus I<sup>2</sup>C. Ceci implique que, sans précautions, si 2 maîtres désirent prendre le contrôle du bus en même temps, on encourrait, sans précautions particulières, 2 risques :
  - La destruction de l'électronique des circuits, pour le cas où l'un d'entre eux impose un niveau haut et l'autre un niveau bas (court-circuit).
  - La corruption des données.

# Principe du bus au niveau hardware

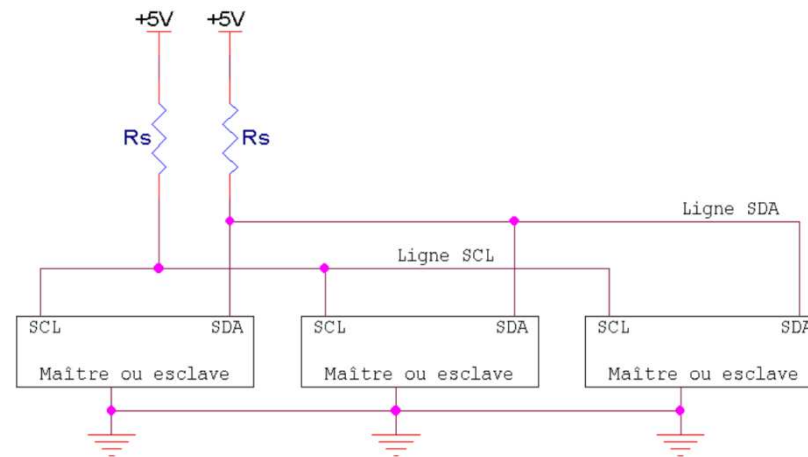
- La parade trouvée par la Société Philips (l'inventeur de l'I<sup>2</sup>C) est :
  - De travailler avec des étages de sortie qui ne peuvent imposer qu'un niveau 0, ou relâcher la ligne, qui remonte d'elle-même au niveau 1 via des résistances de rappel.
  - De cette façon, on n'aura jamais de court-circuit, puisque personne ne peut placer la tension d'alimentation sur la ligne.
  - Ces étages sont des montages appelés « collecteur ouvert » ou « drain ouvert » suivant la technologie utilisée (Bipolaire ou MOS).



# Principe du bus au niveau hardware

## BSI Orientation Génie Electrique

- Chaque circuit peut forcer la ligne SCL ou SDA à 0, mais aucun circuit ne peut la forcer à l'état haut.
- Elle repassera à l'état haut via les résistances de rappel (pull-up) si tous les circuits connectés ont libéré la ligne.
- Le schéma d'interconnexion des circuits sur un bus I<sup>2</sup>C est donc le suivant :

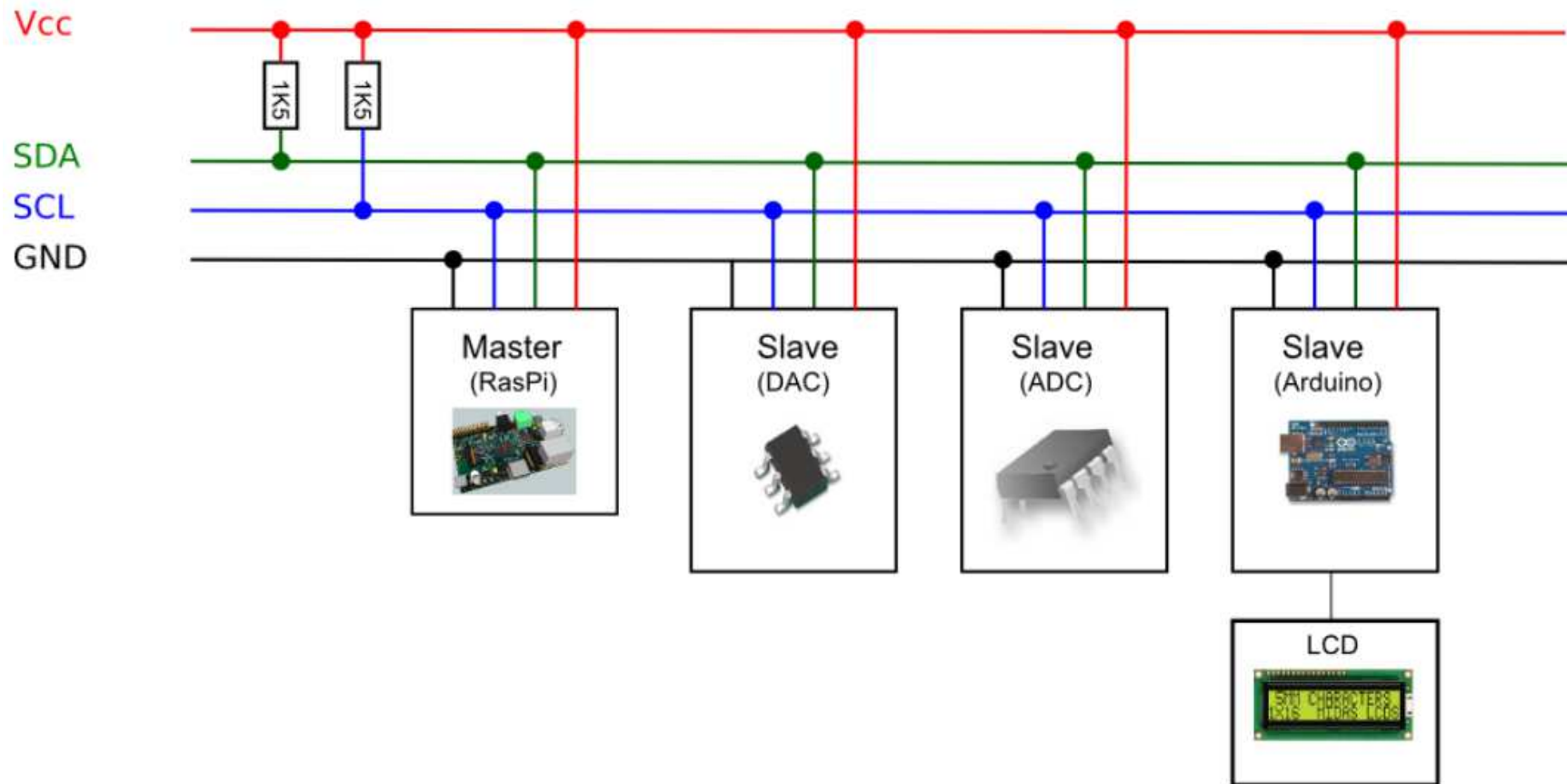


- **ATTENTION** : La masse de tous ces circuits, qui sert de référence, doit évidemment être commune. De ce fait, si ces circuits se situent sur des cartes différentes, ou ne partagent pas la même référence, il vous faudra une ligne supplémentaire dans votre connexion, afin de transmettre la tension de référence. On parle donc couramment de liaison à 2 lignes, mais en réalité 3 lignes sont nécessaires pour communiquer.

# Principe du bus au niveau hardware

## Exemple

BSI Orientation Génie Electrique

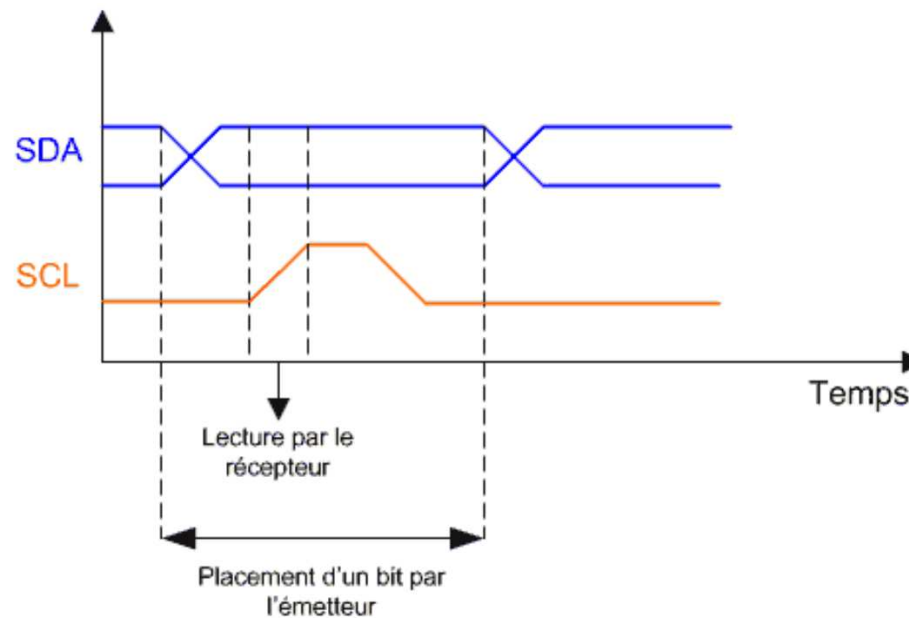


Fabrice Triquet

# Les différents types de signaux

## Le bit « ordinaire »

- Le bit est d'abord placé sur la ligne SDA, puis la ligne SCL est placée à 1 (donc libérée) durant un moment puis forcée de nouveau à 0. L'émission du bit s'effectue donc sans aucune correspondance d'un flanc d'horloge. Il est par contre lu lors du flanc montant de cette horloge.



- Le bit présent sur la ligne SDA doit être présent avant la montée du signal SCL, et continuer un certain temps avant sa redescente.

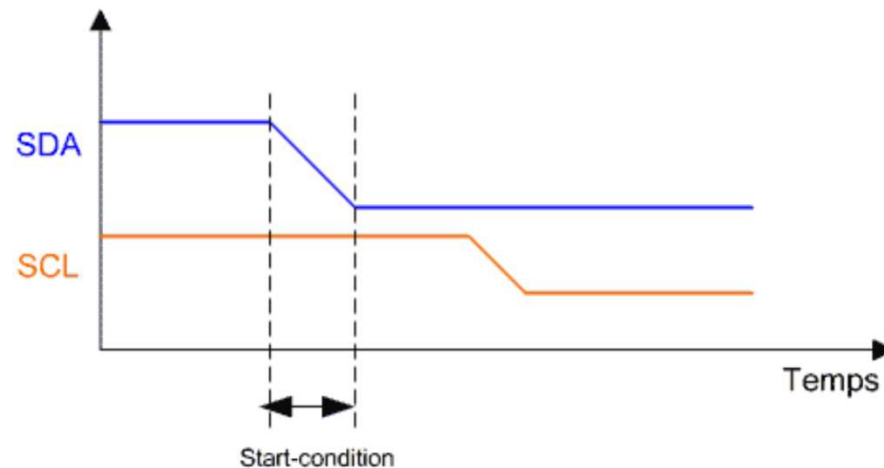


# Les différents types de signaux

## Le start-condition

BSI Orientation Génie Electrique

- Comme pour tout signal synchrone, le protocole ne définit pas de start et de stop-bit. Mais il intègre toutefois les notions de « start-condition » et de « stop-condition ». Ces séquences particulières, obtenues en modifiant la ligne SDA alors que la ligne SCL est positionnée à l'état haut permettent de définir un début et une fin des messages. Ceci est indispensable pour repérer le premier octet du message, qui a un rôle particulier.
- Au repos, SCL et SDA se trouvent relâchés (état haut), le start-condition (symbole conventionnel : S) est réalisé simplement en forçant la ligne SDA à 0, tout en laissant la ligne SCL à 1.



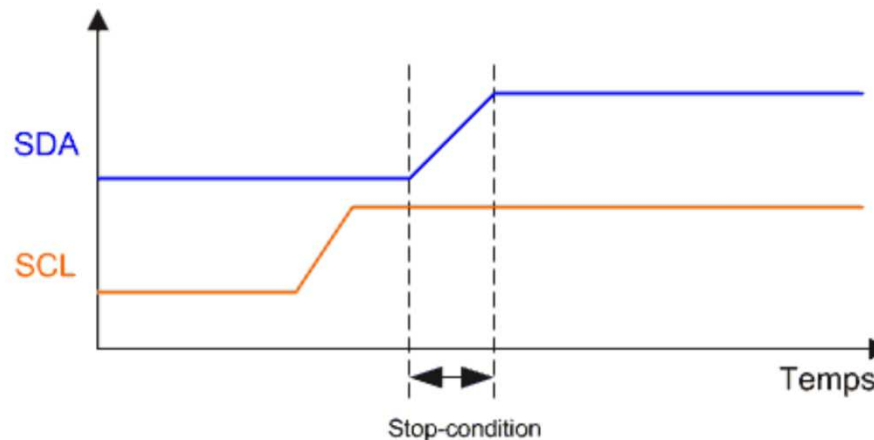


# Les différents types de signaux

## Le stop-condition

BSI Orientation Génie Electrique

- Cette condition indique la fin du message en cours. Elle remet les lignes SDA et SCL au repos, mais en respectant la chronologie suivante :
  - La ligne SDA est ramenée à 1, alors que la ligne SCL se trouve déjà à 1.



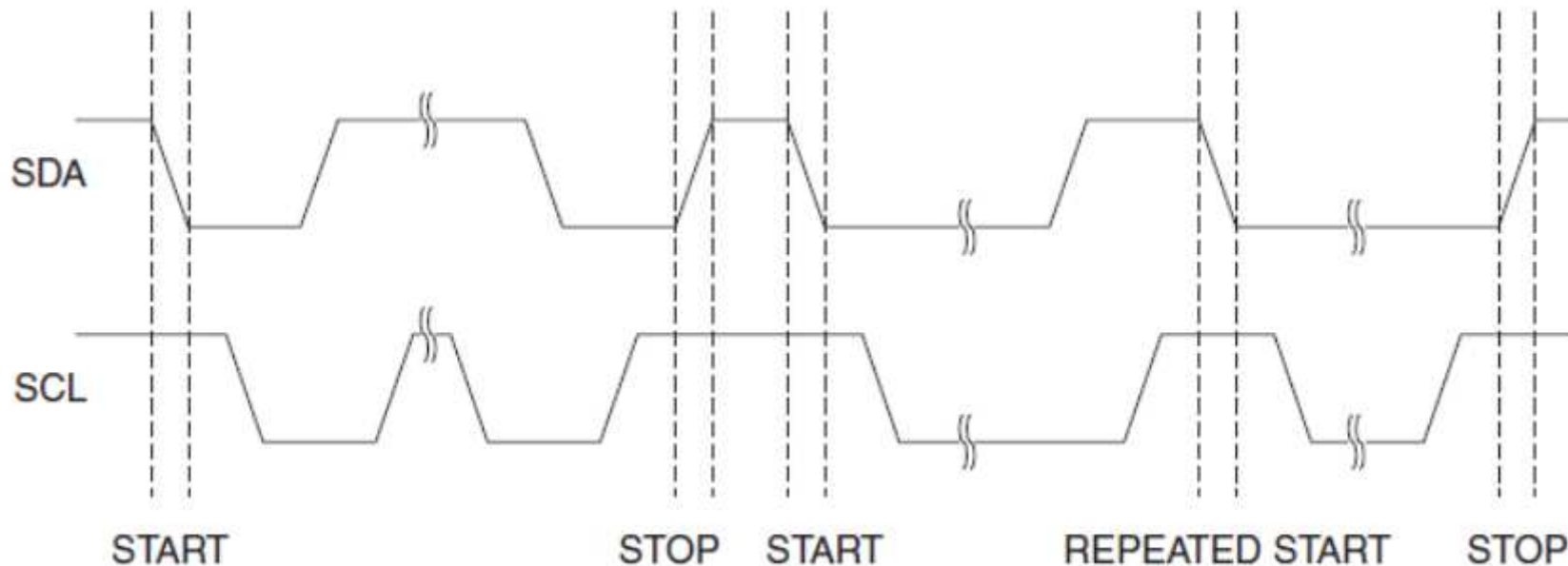
Fabrice Triquet

# Les différents types de signaux

## Le repeat start-condition

BSI Orientation Génie Electrique

- Il peut arriver qu'une condition start soit répétée avant une condition stop, on parle alors de repeated start. Ce dernier s'obtient de la même façon.



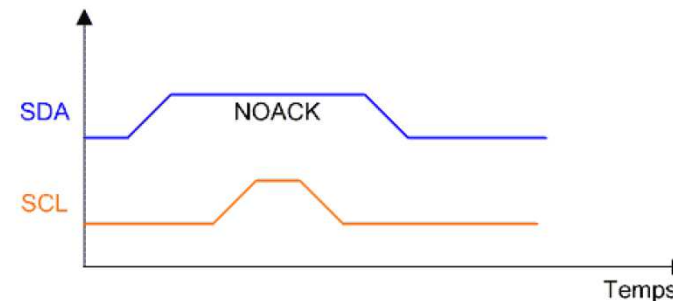
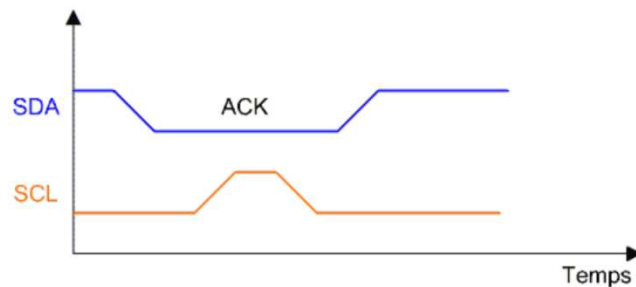
Fabrice Triquet

# Les différents types de signaux

## L'acknowledge

BSI Orientation Génie Electrique

- L'acknowledge (ACK) est en fait l'accusé de réception de l'octet envoyé.
- C'est donc le récepteur qui émet ce bit pour signaler qu'il a bien lu l'octet envoyé par l'émetteur.
- Cet accusé de réception est lu comme un bit classique. Il vaudra 0 si l'accusé de réception signifie « OK », et 1 pour toute autre raison (récepteur dans l'impossibilité de répondre, par exemple).



- NOACK équivaut à une absence de réaction, puisque seul le niveau bas est imposé, le niveau haut étant lié à la libération de la ligne (ou à sa non appropriation).

Fabrice Triquet

# Les différents types de signaux

## Le bit read/write

BSI Orientation Génie Electrique

- Le bit R/W indique à l'esclave si les bits de données contenus dans la trame sont destinés à être écrits ( $R/W = 0$ ) ou lus ( $R/W = 1$ ) par le maître.
- Dans le cas d'une écriture, le maître enverra les données à l'esclave, dans le cas d'une lecture, c'est l'esclave qui enverra ses données au maître.

Fabrice Triquet

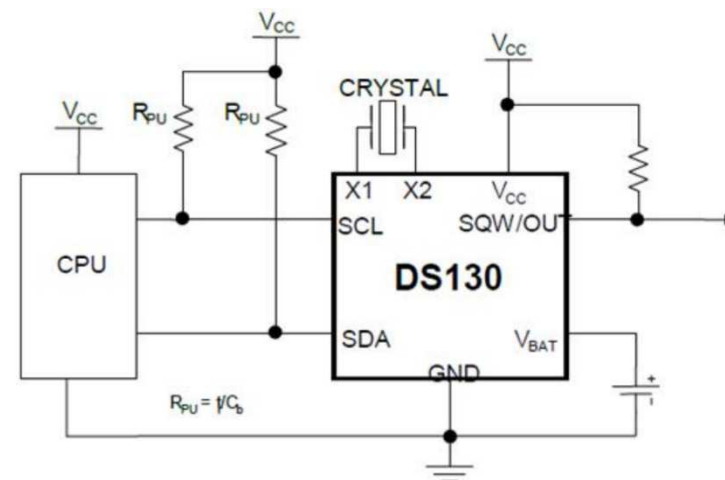
- Le but recherché est de pouvoir connecter un grand nombre de composants I<sup>2</sup>C sur le même bus.
- Comme aucune ligne de sélection existe, c'est de façon logicielle que le destinataire va être sélectionné. Ceci fait naturellement appel à la notion d'adresse.
- La première norme I<sup>2</sup>C limitait la taille des adresses à 7 bits. Au fil de son évolution, on a vu apparaître la possibilité d'utiliser des adresses codées sur 10 bits.
- Seul l'esclave dont l'adresse correspond à celle envoyée par le maître pourra répondre.
- L'adresse '0000000' est utilisée pour adresser simultanément tous les périphériques (general call address).

- Corollaires de tout ce qui précède :
  - Seuls les esclaves disposent d'adresses;
  - Ce sont toujours les maîtres qui pilotent le transfert;
  - Un maître ne peut parler qu'à un esclave (ou à tous les esclaves), jamais à un autre maître;
  - Rien n'interdit cependant qu'un composant passe du statut de maître à celui d'esclave;
  - réciproquement. Le bus I<sup>2</sup>C est donc d'une complète souplesse à ce niveau;

# Exemple de protocole

## Communication avec la RTC (Real Time Clock)

- La RTC employée, real time clock, est la Maxim DS1307. Elle permet de compter les secondes, minutes, heure, le jour de la semaine, la date, le mois et l'année. Sa communication est réalisée par une interface I2C. Les informations sont codées en BCD.
- Le composant est alimenté en 5V et l'alimentation VBat, permettant d'assurer la cohérence des données temporelles en cas de défaillance de l'alimentation, est de 3V.
- La fréquence de la clock I2C (SCL) doit être inférieure ou égale à 100kHz. Le quartz à adjoindre à la RTC résonne à la fréquence de 32,768 kHz et ne requiert pas de condensateurs externes (capacité de 12,5 pF internes).
- La sortie SQW est une sortie square wave/output driver, qui permet, si on l'active via une écriture I2C, de générer des signaux carrés aux fréquences de 1 Hz, 4 kHz, 8 kHz ou 32 kHz.





# Exemple de protocole

## Communication avec la RTC (Real Time Clock)

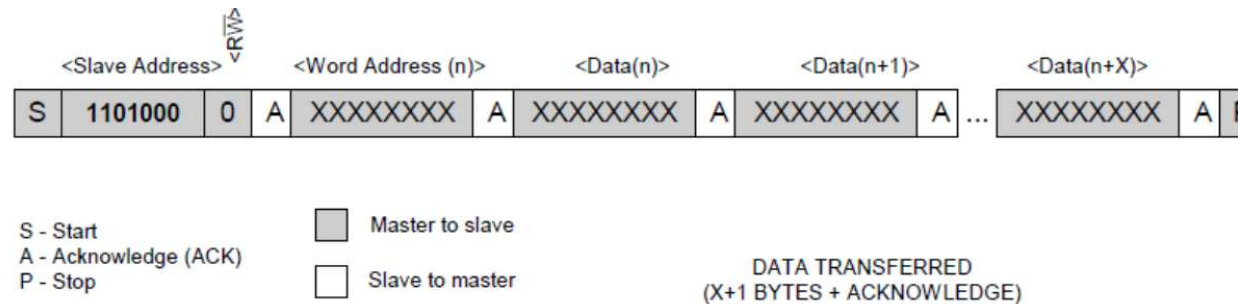
- Le tableau ci-dessous représente les différents registres de la RTC accessibles en I2C. Pour la plupart des registres, les 4 bits de poids faible donnent la valeur des unités tandis que les bits 4, 5 et 6 voire 7 donnent le nombre de dizaines. Dans le cas des heures, le bit 5 détermine le format 12 ou 24h.
- Le registre Control permet d'opérer sur la sortie de la RTC : mettre la sortie à 1 ou générer un signal carré de fréquence 1, 4 000, 8 000 ou 32 000 Hz.

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	FUNCTION	RANGE
00h	CH	10 Seconds			Seconds				Seconds	00–59
01h	0	10 Minutes			Minutes				Minutes	00–59
02h	0	12	10 Hour	10 Hour	Hours				Hours	1–12 +AM/PM 00–23
		24	PM/ AM							
03h	0	0	0	0	0	DAY			Day	01–07
04h	0	0	10 Date		Date				Date	01–31
05h	0	0	0	10 Month	Month				Month	01–12
06h	10 Year				Year				Year	00–99
07h	OUT	0	0	SQWE	0	0	RS1	RS0	Control	—
08h–3Fh									RAM 56 x 8	00h–FFh

## Communication avec la RTC (Real Time Clock) ECRITURE

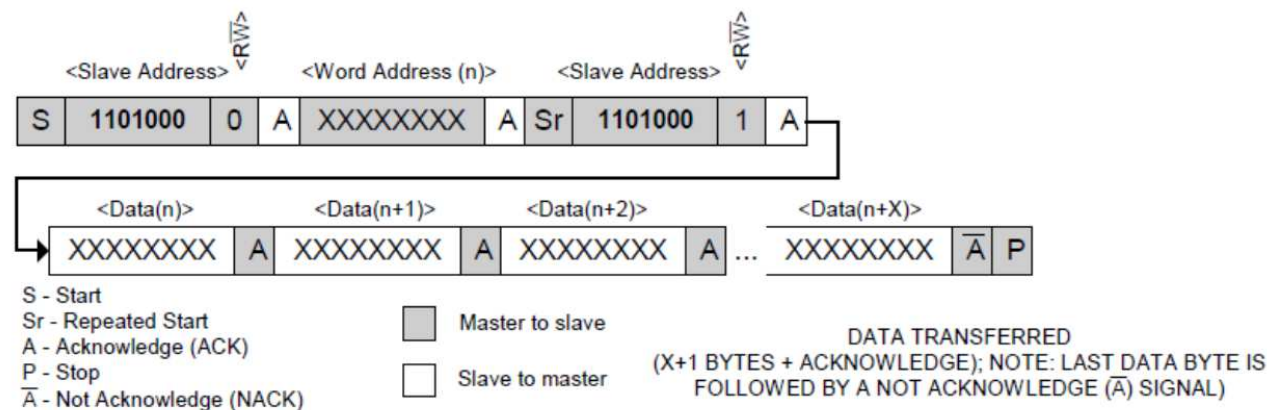
BSI Orientation Génie Electrique

- L'écriture d'une data débute par l'envoi d'un start. Lorsque la ligne SDA est tombée à 0 avant la clock, on envoie l'adresse de l'esclave, ici l'adresse de la RTC, codée sur 7 bits.
- Le huitième bit (read/write) permet de connaître le type d'opération que l'on va effectuer : lecture ou écriture.
- Si l'esclave a bien reçu l'information, il répond par un ACK correspondant à une mise à 0 de la ligne SDA avant que la clock ne passe à 1
- Ensuite, le maître envoie l'adresse du registre de la RTC qu'il souhaite modifier (word address).
- Après réception d'un nouvel ACK, le maître envoie les modifications à apporter aux registres de la RTC. Toutes les données envoyées sont entrecoupées d'un acquittement par l'esclave et l'adresse du registre écrit est incrémentée avant chaque nouvel envoi de données.
- **Le maître termine la communication en envoyant une condition de stop.**



Fabrice Triquet

- La lecture d'une information débute par l'envoi de l'adresse de l'esclave concaténé au bit d'écriture suivi de l'envoi du registre dont on souhaite extraire les données.
- Après réception de l'ACK de la RTC suite à la transmission du Word Address, un repeated start est généré par le maître suivi de l'envoi de l'adresse de l'esclave adressé à laquelle le bit read/write fixé à 1 est ajouté.
- L'esclave confirme la réception de l'octet par l'envoi d'un acquittement au maître.
- Ensuite, les données des différents registres de la RTC sont envoyées, l'adresse de la donnée est incrémentée avant chaque envoi. Les datas sont acquittées par le maître.
- Un NACK du récepteur signifie la fin de la lecture d'informations. Un stop est ensuite généré par le maître pour libérer la ligne I2C.



- Chaque périphérique Maître ou Esclave a son propre protocole.
- Ce protocole est définit dans la data sheet du fabricant.
- Dans le cadre du projet, on se limitera a considérer le  $\mu C$  comme maître et les autres périphériques comme esclaves.
- L'utilisation des interruptions doit être exploitée (lire la note d'application AVR315).