



eMbedded
Information
Systems

Le bus I2C

Inter Integrated Circuit



Bus I2C

- ▶ Bus sériel 2 fils pour données et contrôle
- ▶ Une ligne de données SDA
- ▶ Une ligne d'horloge SCL
- ▶ Conditions uniques de start et stop
- ▶ Possibilité d'avoir plusieurs maîtres (un seul à la fois)
- ▶ Plusieurs esclaves, sélectionnés par une adresse 7-Bit
- ▶ Transfert bidirectionnel de données
- ▶ Acknowledge après chaque octet transféré
- ▶ Pas de limite au nombre d'octet transférés

Familles de circuits avec I2C

- ▶ Microcontrôleurs
- ▶ Microprocesseurs
- ▶ Périphériques généraux
 - ▶ Interfaces I2C-parallèle, I/O
 - ▶ Mémoires, Afficheur, DAC, ADC, Horloges (RTC)
- ▶ Périphériques pour applications spécifiques
 - ▶ Audio, Telephonie, Vidéo

Caractéristiques

- ▶ Synchronisation d'horloge
- ▶ Procédure d'arbitrage
- ▶ Vitesse de transmission jusqu'à 100Khz
- ▶ Longueur de bus jusqu'à 4 mètres
- ▶ Capacité pilotable jusqu'à 400pF
- ▶ Possibilité de placer des résistances en série pour des questions de protection des circuits
- ▶ Compatible avec la plupart des technologies (TTL, CMOS, etc...).

Définitions

▶ Maître :

- ▶ Initie un transfert en générant des conditions de start et stop
- ▶ Génère l'horloge
- ▶ Transmet l'adresse de l'esclave
- ▶ Détermine le sens du transfert de données

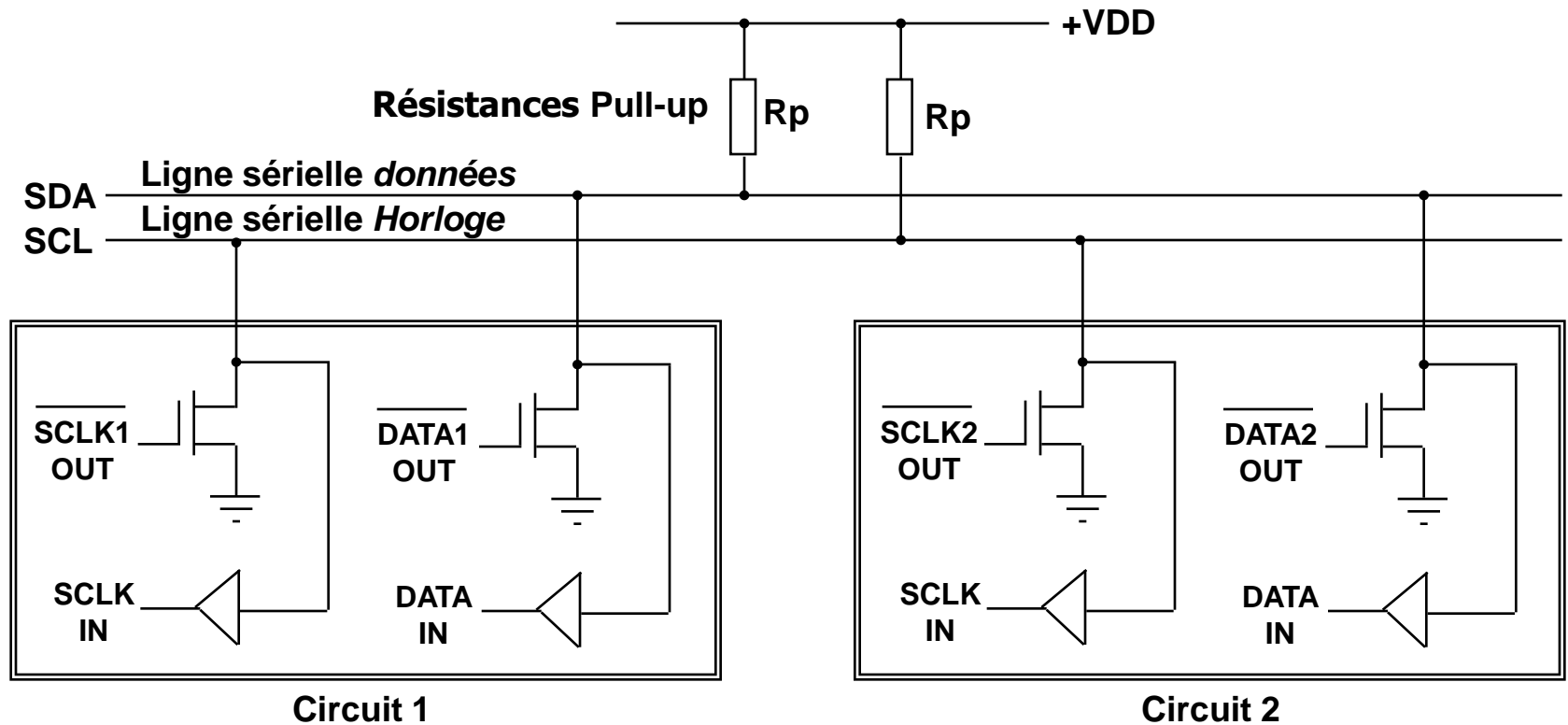
▶ Esclave :

- ▶ Répond seulement quand il est adressé
- ▶ Le timing est contrôlé par la ligne d'horloge SCL

Aspects matériels

- ▶ Les circuits connectés sur le bus doivent avoir un accès “Drain ouvert” pour les deux lignes SDA (données) et SCL (horloge)
- ▶ Les circuits doivent être capables de détecter le niveau logique
- ▶ Tous les circuits doivent avoir une référence GND commune
- ▶ Les lignes SCL et SDA sont reliées à VCC via des résistances de pull-up
- ▶ A tout moment, le bus I2C est dans l'un des états suivants :
 - ▶ Au repos
 - ▶ En mode de transmission maître
 - ▶ En mode de réception esclave

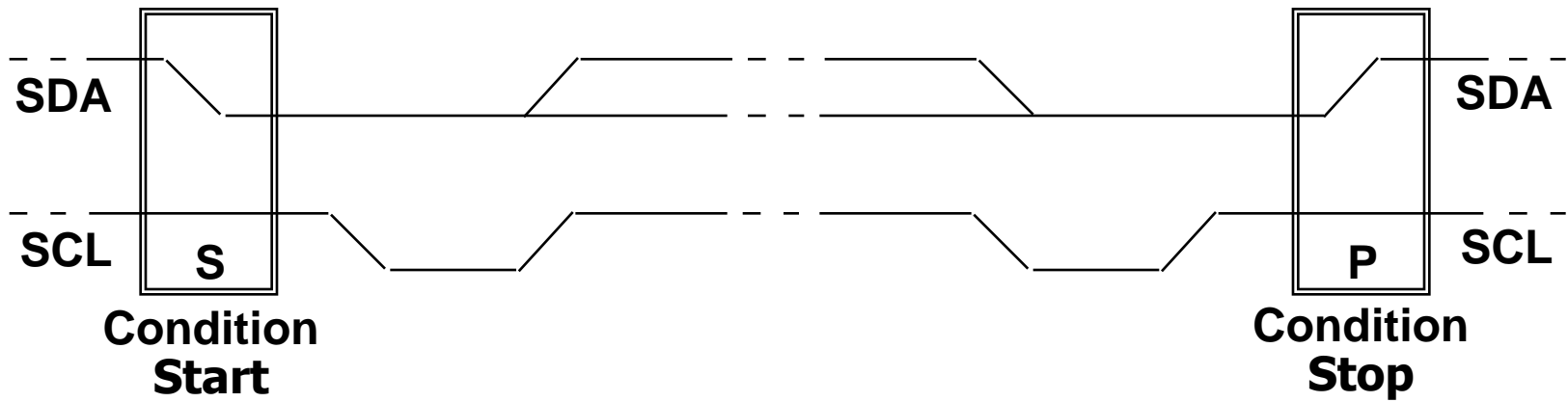
Configuration « Drain ouvert »



Les circuit I2C sont câblés en ET logique

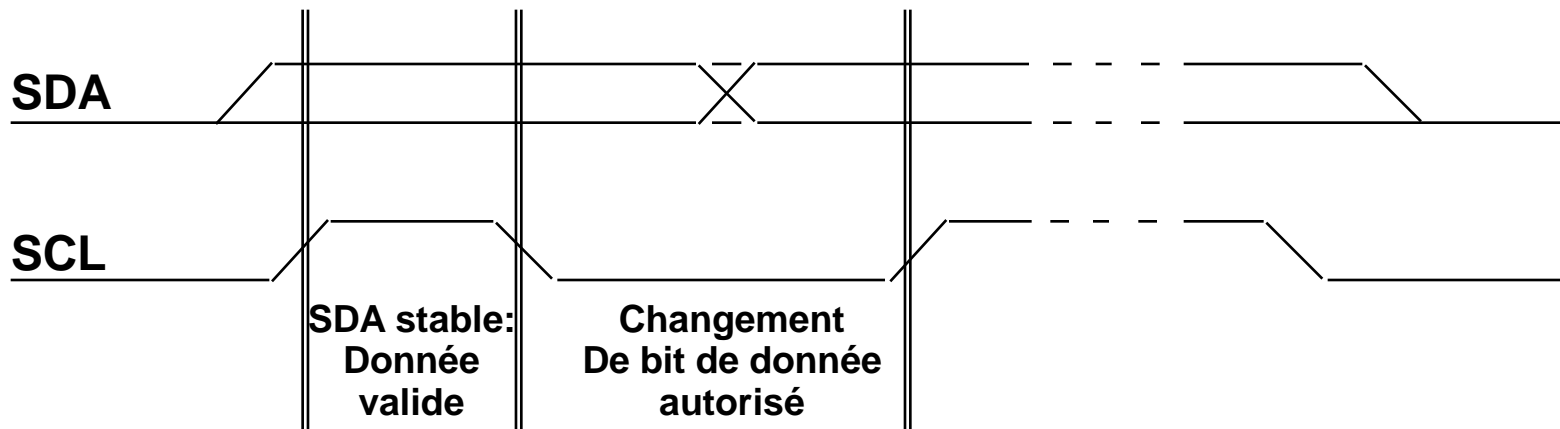
Conditions de Start et Stop

- ▶ Une transition sur SDA, alors que SCL est à '1', est définie comme une condition de Start ou Stop
- ▶ Ces deux conditions (Start et Stop) sont générées par le maître du bus
- ▶ Le Bus est occupé (busy) après une condition de Start



Transferts de données

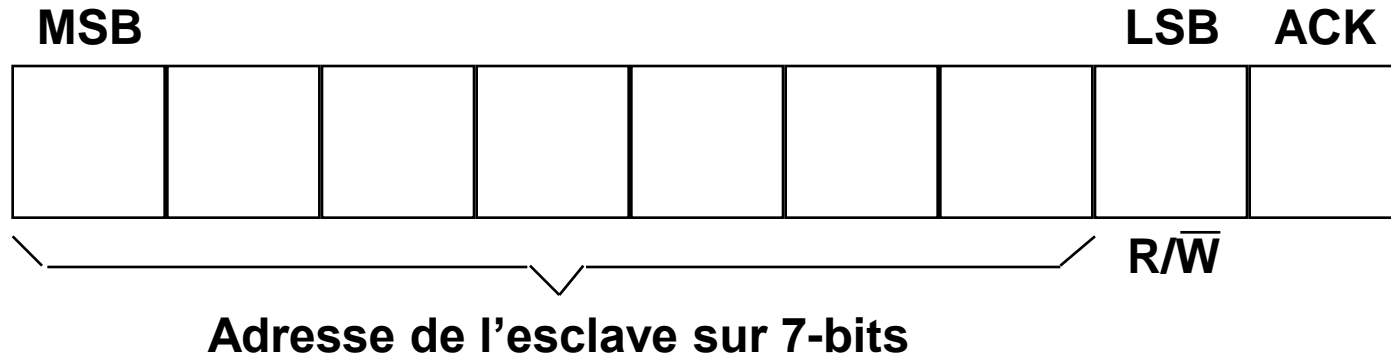
- ▶ En mode de transfert normal, la ligne de données SDA change d'état seulement lorsque SCL est à '0'



Adressage I2C

- ▶ Chaque noeud (périphérique) a une adresse unique sur 7 bits
- ▶ Généralement, les noeuds ont une partie fixe et une partie programmable de leur adresse
- ▶ Les adresses commençant par 0000 ou 1111 ont des fonctions spéciales
 - ▶ 0000000 est une adresse globale
 - ▶ 0000001 est une adresse sans objet (Null)
 - ▶ 1111xxx est réservée pour les expansions du bus

Premier octet transmis



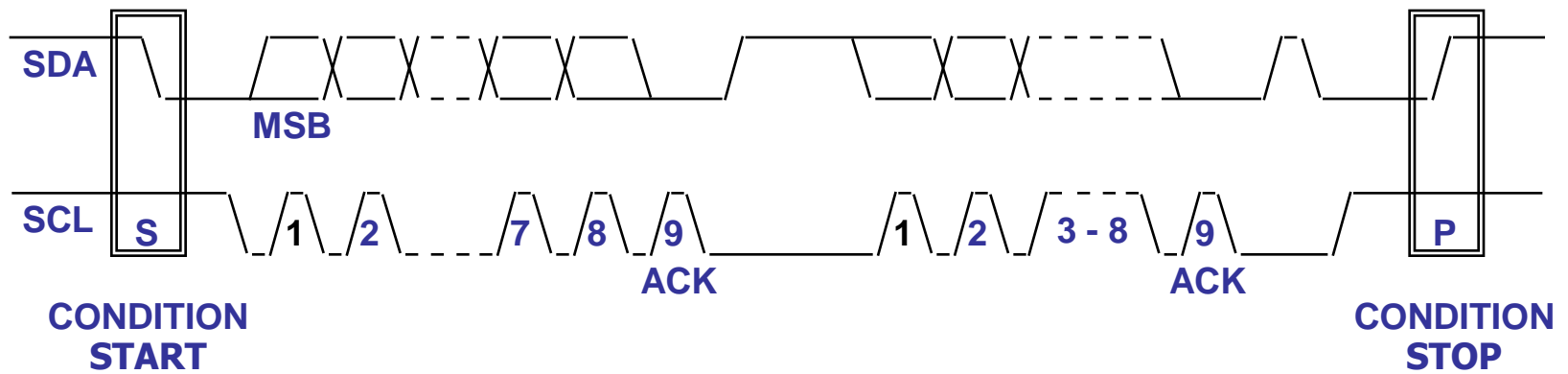
▶ R/W :

- ▶ 0 : Le maître écrit vers l'esclave
- ▶ 1 : Le maître lit l'esclave

Acknowledgement

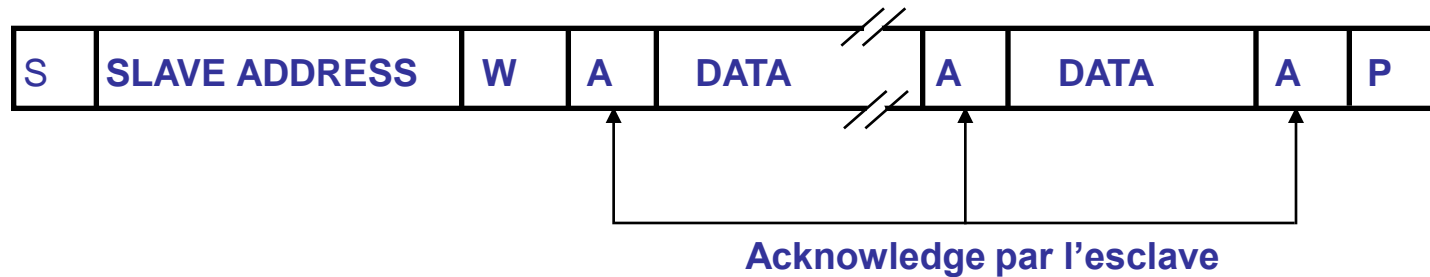
- ▶ Les récepteurs (Maître ou Esclave) tirent la ligne de donnée SDA à '0' durant une période d'horloge après réception d'un octet.
- ▶ Les récepteurs Maître laissent la ligne SDA à '1' après réception du dernier octet ***demandé***.
- ▶ Les récepteurs Esclave laissent la ligne SDA à '1' sur l'octet qu'ils ***peuvent accepter***.

Transferts de données

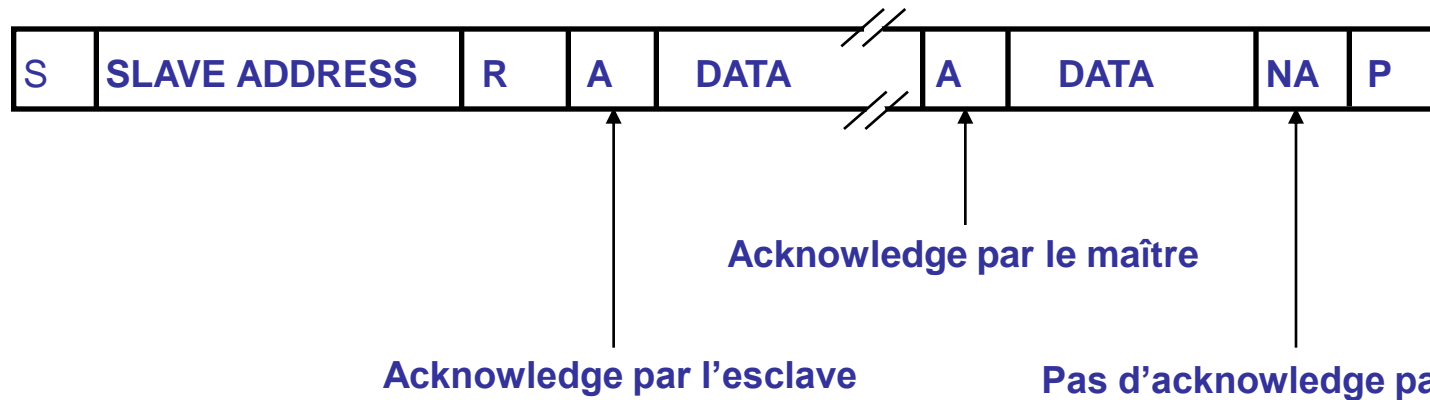


Formats de données possibles

Ecriture depuis le maître :



Lecture depuis le maître :

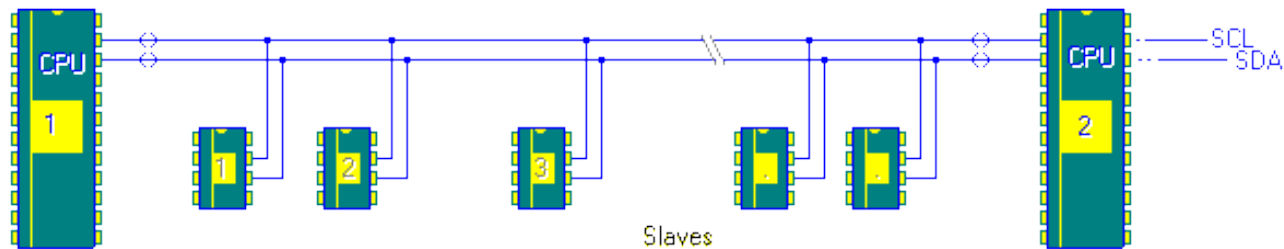


Synchronisation d'horloge

- ▶ La synchronisation d'horloge est utilisée pour synchroniser des maîtres en cas d'arbitration nécessaire entre plusieurs maîtres
- ▶ Elle peut aussi être utilisée comme “handshake” par un circuit esclave pour ralentir le transfert de données depuis un maître
- ▶ La procédure de synchronisation d'horloge est possible de deux façons :
 1. Si la ligne SCL descend lorsqu'un maître génère un '1', le maître génère un '0' et allonge la durée de son alternance à '0'
 2. Quand un maître cesse de générer un '0' sur SCL, il attend jusqu'à ce que la ligne monte effectivement avant de compter l'alternance à '1' (ceci est permis par le fait que la ligne SCL est tirée à '1' par la résistance pull-up)

Systemes multi-maitres

- ▶ La mise en place de systemes multi-maitres necessite des caracteristiques supplementaires



Arbitration

- ▶ L'arbitration est le mecanisme par lequel des maitres concurrents decident lequel d'entre eux aura le controle du bus
- ▶ L'arbitration I2C ne corrompt pas les donnees transmises par le maitre gagnant
- ▶ L'arbitration est perdue par un maitre quand il tente de generer un '1' sur la ligne de donnees SDA, et echoue
- ▶ Ce mecanisme d'arbitration est aussi utilise dans des bus plus sophistiques comme le bus de terrain CAN (Controller Area Network)



Communication I2C avec le MSP430

Le module du microcontrôleur utilisé pour la communication est le USCI_B0.

LEDs dimmer

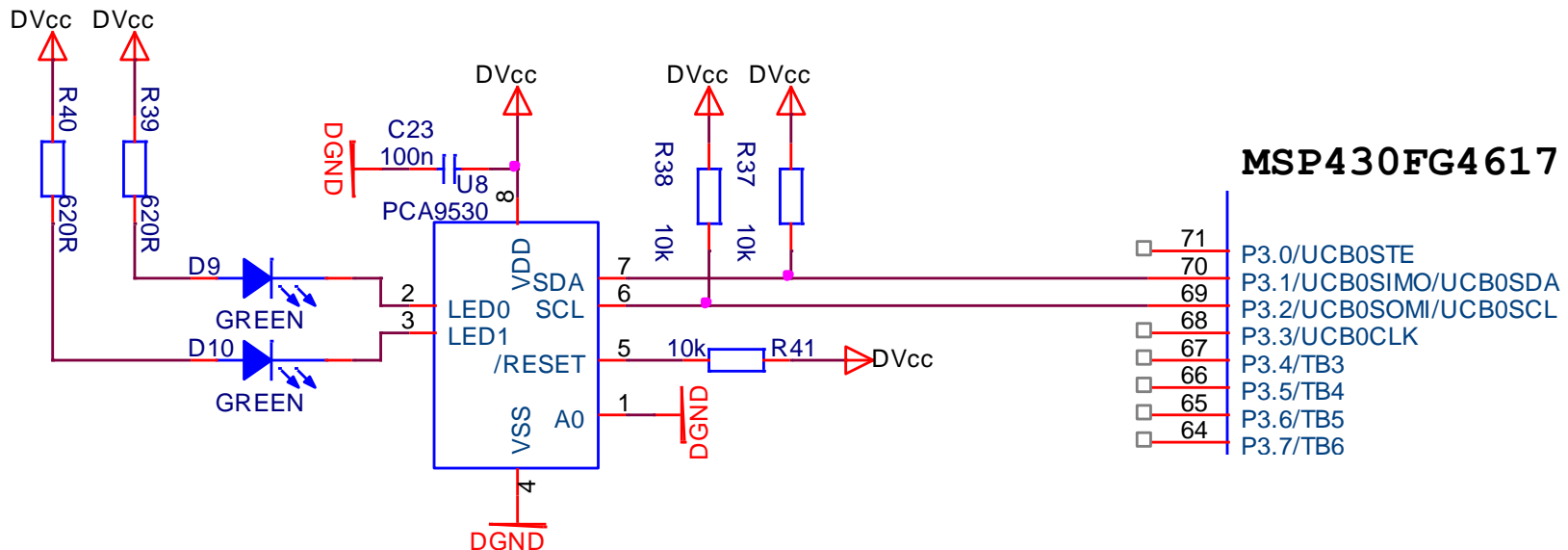


Schéma du branchement du LEDs dimmer I2C de la carte IAI

Registres pour le MSP430F4617

No	Registre	Nom court	Bits à configurer
1	Port 3 select register	P3SEL	Bits 1 et 2
2	USCI_B1 control register 1	UCB0CTL1	UCSWRST, UCSSELx
3	USCI_B1 control register 0	UCB0CTL0	UCMST + UCMODE_3 + UCSYNC
4	USCI_B0 bit rate control register 1 & 0	UCB0BR1, UCB0BR0	tous
5	USCI_B0 I2C slave address register	UCB0I2CSA	tous

Registres à initialiser

No	Registre	Nom court	Bits à utiliser
6	USCI_B1 transmit buffer register	UCB0TXBUF	tous
2	USCI_B1 control register 1	UCB0CTL1	UCTR, UCTXSTT, UCTXSTP
7	SFR interrupt enable register 2	IE2	UCB0TXIE

Registres à utiliser lors de communication

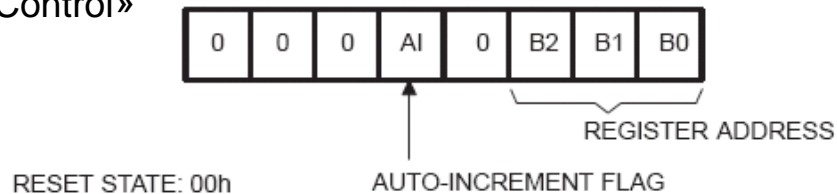
En pratique ...

- ▶ Il faut toujours garder à l'esprit que le bus I2C est un bus multi-master, et que les esclaves peuvent mettre plus ou moins de temps à réagir. Il faut donc:
 - ▶ Attendre que le bus soit libre avant de communiquer
 - ▶ Que l'esclave quitte les communications.

- ▶ Voici les étapes d'une communication:
 1. Dans le préambule du programme:
 2. Configuration des pins I/O utilisées pour la communication I2C.
 3. Configuration des registres UCB0CTL0 et UCB0CTL1 du MSP430 régissant la communication I2C par le module USCI. Il faut mettre le module USCI_B0 en reset pendant sa configuration à l'aide de UCSWRST. Enlever le reset à la fin de la configuration.
 4. Configuration de l'adresse de l'esclave par le registre UCB0I2CSA.
 5. Le maître initie la communication par une « start condition ».

-
6. Le maître envoie l'adresse du périphérique esclave avec qui il veut communiquer (7 bits), ainsi qu'un bit R/W indiquant le sens des données à suivre. L'esclave doit quitter la bonne réception de ceci en tirant la ligne de donnée en bas.
 7. Les données sont transmises. Il peut y avoir plusieurs octets.

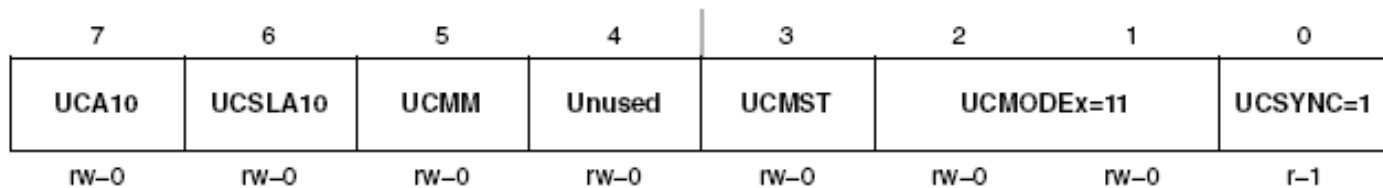
- ▶ Envoi du byte de «Control»



Composition du byte de contrôle

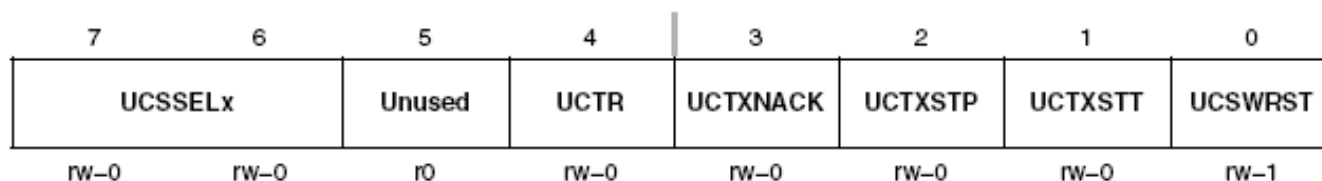
- ▶ L'« auto-increment » sert à indiquer qu'on veut écrire dans plusieurs registres successivement. L'adresse (« B2, B1, B0 ») indique dans quel registre du PCA9530 sera chargée la valeur qui suit.
 - ▶ Envoi des autres valeurs, selon l'application programmée.
8. Le maître termine la communication par une « stop condition ».

UCBxCTL0, USCI_Bx Control Register 0



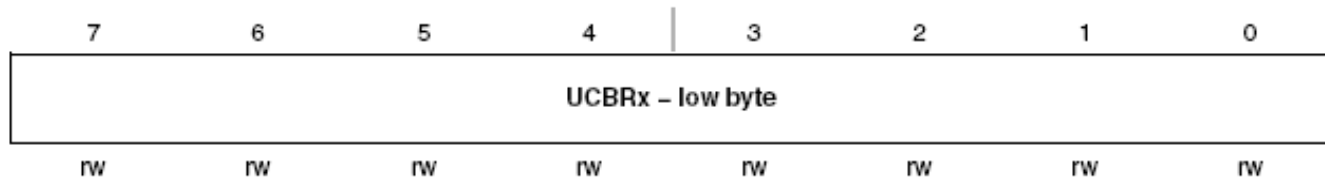
UCA10	Bit 7	Own addressing mode select 0 Own address is a 7-bit address 1 Own address is a 10-bit address
UCSLA10	Bit 6	Slave addressing mode select 0 Address slave with 7-bit address 1 Address slave with 10-bit address
UCMM	Bit 5	Multi-master environment select 0 Single master environment. There is no other master in the system. The address compare unit is disabled. 1 Multi master environment
Unused	Bit 4	Unused
UCMST	Bit 3	Master mode select. When a master loses arbitration in a multi-master environment (UCMM = 1) the UCMST bit is automatically cleared and the module acts as slave. 0 Slave mode 1 Master mode
UCMODEx	Bits 2-1	USCI Mode. The UCMODEx bits select the synchronous mode when UCSYNC = 1. 00 3-pin SPI 01 4-Pin SPI (master/slave enabled if STE = 1) 10 4-Pin SPI (master/slave enabled if STE = 0) 11 I ² C mode
UCSYNC	Bit 0	Synchronous mode enable 0 Asynchronous mode 1 Synchronous mode

UCBxCTL1, USCI_Bx Control Register 1

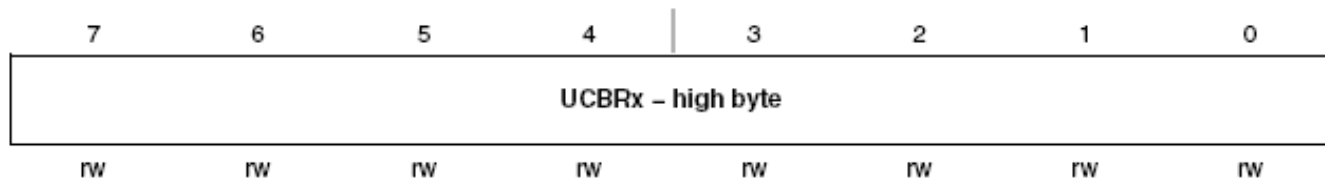


UCSSELx	Bits 7-6	USCI clock source select. These bits select the BRCLK source clock. 00 UCLKI 01 ACLK 10 SMCLK 11 SMCLK
Unused	Bit 5	Unused
UCTR	Bit 4	Transmitter/Receiver 0 Receiver 1 Transmitter
UCTXNACK	Bit 3	Transmit a NACK. UCTXNACK is automatically cleared after a NACK is transmitted. 0 Acknowledge normally 1 Generate NACK
UCTXSTP	Bit 2	Transmit STOP condition in master mode. Ignored in slave mode. In master receiver mode the STOP condition is preceded by a NACK. UCTXSTP is automatically cleared after STOP is generated. 0 No STOP generated 1 Generate STOP
UCTXSTT	Bit 1	Transmit START condition in master mode. Ignored in slave mode. In master receiver mode a repeated START condition is preceded by a NACK. UCTXSTT is automatically cleared after START condition and address information is transmitted. Ignored in slave mode. 0 Do not generate START condition 1 Generate START condition
UCSWRST	Bit 0	Software reset enable 0 Disabled. USCI reset released for operation. 1 Enabled. USCI logic held in reset state.

UCBxBR0, USCI_Bx Baud Rate Control Register 0



UCBxBR1, USCI_Bx Baud Rate Control Register 1



UCBRx

Bit clock prescaler setting.

The 16-bit value of $(UCBxBR0 + UCBxBR1 \times 256)$ forms the prescaler value.