



# Le BUS 1 WIRE

*Documentation interne ELE  
D. MENESPLIER 2004*

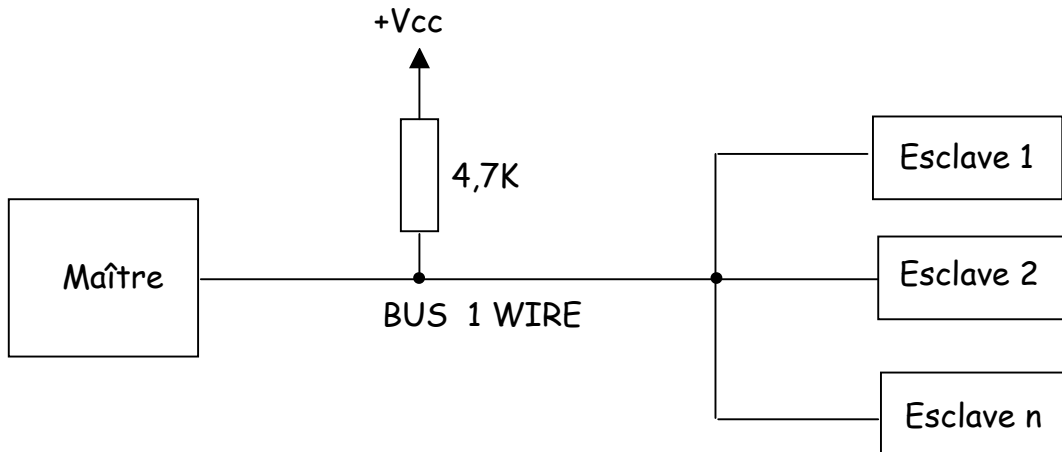
Le bus 1 WIRE de DALLAS, permet de connecter et de faire dialoguer entre eux des circuits sur un seul fil.

Ce système de bus utilise un seul maître, qui pourra dialoguer avec un ou plusieurs esclaves.

Toutes les commandes et données sont envoyées avec le bit LSB en tête.

Le fil unique du bus doit être tiré au +Vcc par une résistance de 4,7K $\Omega$ .

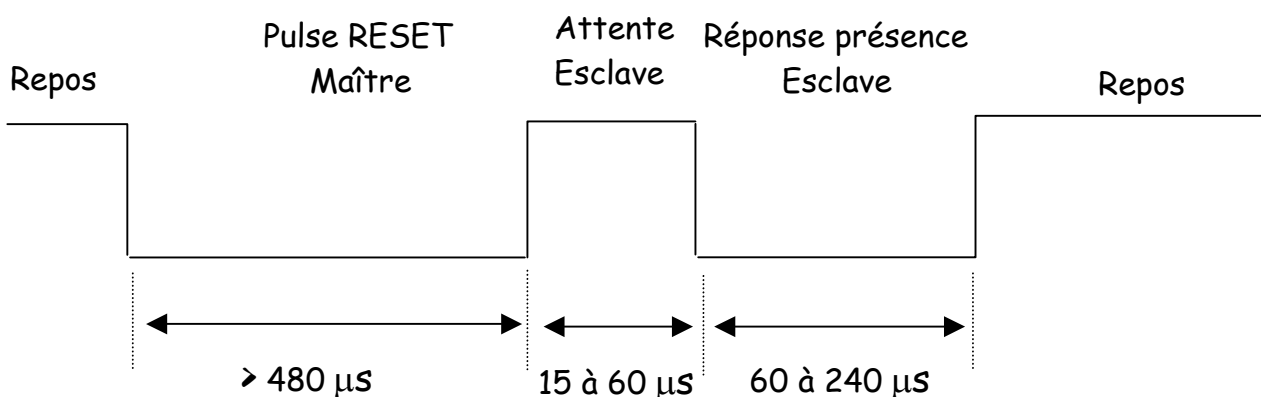
L'état repos du bus est donc un état haut.



Si le bus est maintenu à l'état bas plus de 480  $\mu$ s par le maître, tous les composants sur le bus sont remis à zéro.

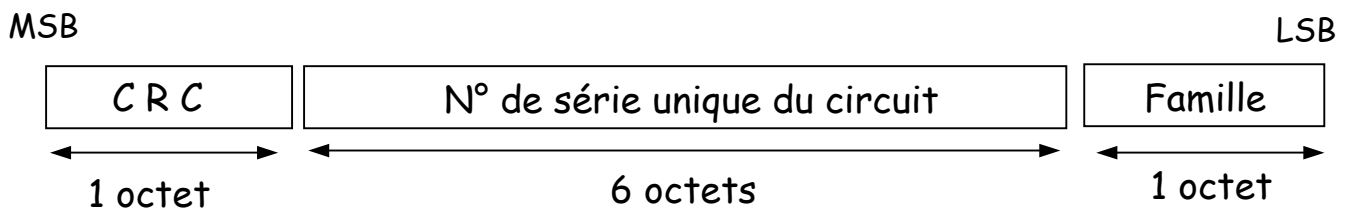
C'est le pulse d'initialisation ou de Reset.

Après un délai de 15 à 60  $\mu$ s, le ou les esclaves raccordés, forcent le bus à l'état bas pendant 60 à 240  $\mu$ s pour signaler leur présence.



Chaque circuit possède une adresse physique unique, gravée dans la puce à la fabrication.

Cette adresse est constituée de 64 bits soit 8 octets. Le premier octet détermine le type de famille auquel appartient le circuit. Les 6 octets suivants, constituent le code propre du circuit. Le dernier octet est le CRC. C'est un octet de contrôle calculé à partir des 56 bits précédents.



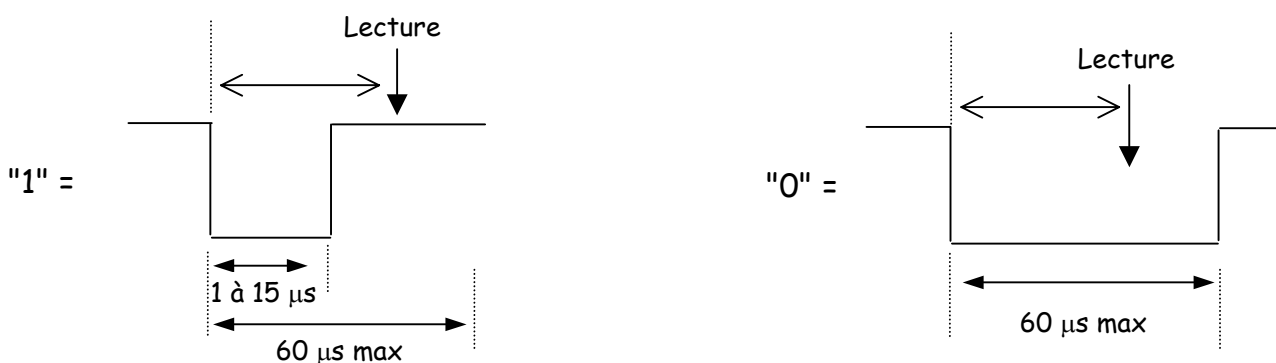
Toute transaction entre un maître et un ou plusieurs esclaves, débute par une initialisation, constituée par l'envoi du pulse de Reset par le maître. Le maître doit ensuite envoyer une commande de type ROM qui est propre au protocole 1 Wire, et que tous les circuits de ce type vont reconnaître. Cela va permettre entre autre de sélectionner un circuit parmi les différents esclaves qui ont répondu présents au pulse de Reset. Le dialogue et l'échange de données pourra ensuite commencer, entre le maître et l'esclave sélectionné.

### Emission d'un bit du maître vers l'esclave:

Le maître force le bus à "0" pendant 1 à 15  $\mu$ s. L'esclave va lire le bus entre 15 et 45  $\mu$ s après le front descendant ( valeur typique 30  $\mu$ s).

Si on veut émettre un "1", il faut repasser le bus à "1" immédiatement, et ne plus rien faire jusqu'à  $t = 60 \mu$ s. Pour émettre un "0" il faut laisser le bus à "0" jusqu'à  $t = 60 \mu$ s, puis repasser le bus à "1".

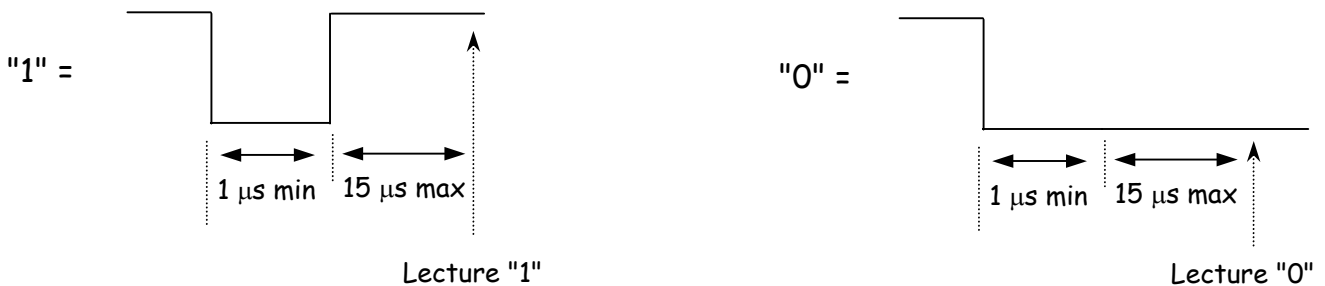
La durée du bit est donc de 60  $\mu$ s, ce qui donne un débit de 16 kbits/sec.



## Réception d'un bit par le maître:

Le maître force le bus à "0" pendant au moins 1  $\mu$ s. Si l'esclave veut émettre un "1", il laisse le bus libre donc tiré à "1". Pour émettre un "0", l'esclave doit tirer le bus à "0" pendant 15  $\mu$ s au minimum.

Le maître devra donc dans tous les cas lire le bus 15  $\mu$ s maximum après avoir tiré le bus à "0" pendant 1  $\mu$ s. L'état du bus donnera alors le bit transmis par l'esclave.



## COMMANDES ROM:

Ces commandes sont constituées d'un octet que le maître devra envoyer après avoir fait un reset.

### h'33' = READ ROM

Cette commande ne peut être utilisée que s'il n'y a qu'un seul esclave sur le bus. Celui ci répond alors ces 64 bits de code.

### h'55' = MATCH ROM

Cette commande suivi de 64 bits de code, va permettre au maître de sélectionner un esclave particulier.

### h'CC' = SKIP ROM

Commande d'appel général, pour adresser tous les esclaves. Cette fonction est utile pour adresser un esclave qui est seul sur le bus, sans avoir à envoyer les 64 bits de son code.

## h'FO' = SEARCH ROM

Cette commande va permettre de rechercher bit à bit les codes de tous les esclaves raccordés au bus 1 Wire.

En réponse à cette commande, les esclaves envoient leur premier bit, puis ce même bit inversé. Le maître émet à son tour ce premier bit. Les esclaves qui reconnaissent leur 1<sup>er</sup> bit restent à l'écoute, et les autres s'éliminent et ne répondront plus. Les esclaves toujours présents vont maintenant envoyer leur 2<sup>eme</sup> bit, puis ce même 2<sup>eme</sup> bit mais inversé. Le maître comme précédemment va émettre ce 2<sup>eme</sup> bit. Les esclaves qui ne reconnaissent pas leur 2<sup>eme</sup> bit vont s'éliminer.

Quand le maître reçoit le bit et son inverse à : 1 1 c'est qu'il n'y a pas de circuit sur le bus 1Wire.

Quand le maître reçoit le bit et son inverse à : 0 0 c'est qu'il y a conflit, car des esclaves ont un "1" et des autres un "0" à cette position. Dans ce cas il enverra en réponse un bit à "0" pour ne garder que les circuits ayant un "0" à cette position et éliminer ceux qui ont un "1".

Quand le maître reçoit le bit et son inverse à : 0 1 c'est qu'il n'y a que des circuits ayant un bit à "0" à cette position. Il enverra un "0" pour garder tous ces circuits. Et s'il a reçu 1 0 il enverra "1", car le bit de cette position est à "1" et on gardera les circuits.

Le principe général de la recherche est de désélectionner les uns après les autres les circuits à chaque conflit sur les différentes positions des bits.

A la fin de chaque étape de recherche, le maître connaît un nouveau code de 64 bits complet d'un circuit. L'étape suivante est identique jusqu'au niveau de la dernière décision après le conflit. Le maître part alors dans la direction opposée, il enverra un "1" alors qu'il n'avait gardé que les circuits ayant un "0" à cette position. Ainsi, bit par bit, on va arriver à lire les 64 bits de tous les esclaves. Le maître va ainsi savoir combien il y a d'esclaves sur le bus et quelles sont leurs codes propres.

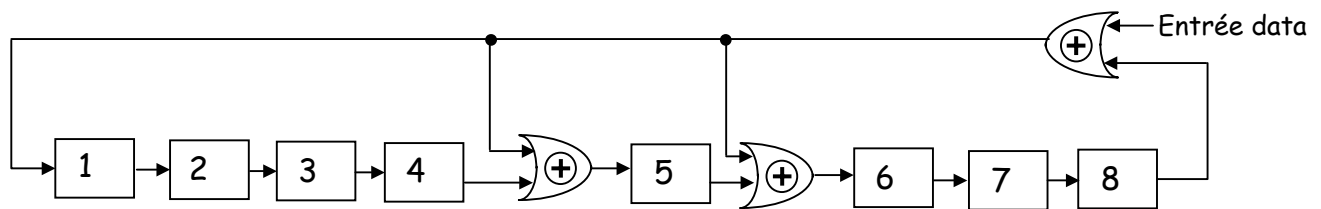
## h'EC' = CONDITIONAL SEARCH

Cette commande fonctionne comme la commande SEARCH ROM, à la différence que seul les circuits ayant une condition bien spécifiée participent à la recherche. Par exemple les circuits de mesure de la température qui ont le Flag d'alarme actif ou les port E/S qui ont leur sortie à "1".

## Le CRC:

Le polynôme générateur du CRC est :  $X^8 + X^5 + X^4 + 1$

Sa représentation est la suivante:



## Calcul pratique du CRC:

Quand on aura reçu les 56 premiers bits du circuit, soit 7 octets, on devra calculer l'octet de CRC pour le comparer à celui que l'on va recevoir avec les 8 bits restants.

Pour éviter les calculs complexes du polynôme, on va utiliser une table indexée de 256 valeurs décimales.

index																
0	0	94	188	226	97	63	221	131	194	156	126	32	163	253	31	65
16	157	195	33	127	252	162	64	30	95	1	227	189	62	96	130	220
32	35	125	159	193	66	28	254	160	225	191	93	3	128	222	60	98
48	190	224	4	92	223	129	99	61	124	34	192	158	29	67	161	255
64	70	24	250	164	39	121	155	197	132	218	56	102	229	187	89	7
80	219	133	103	57	186	228	6	88	25	71	165	251	120	38	196	154
96	101	59	217	135	4	90	184	230	167	249	27	69	198	152	122	36
112	248	166	68	26	153	199	37	123	58	100	134	216	91	5	231	185
128	140	210	48	110	237	179	81	15	78	16	242	172	47	113	147	205
144	17	79	173	243	112	46	204	146	211	141	111	49	178	236	14	80
160	175	241	19	77	206	144	114	44	109	51	209	143	12	82	176	238
176	50	108	142	208	83	13	239	177	240	174	76	18	145	207	45	115
192	202	148	118	40	171	245	23	73	8	86	180	234	105	55	213	139
208	87	9	235	181	54	104	138	212	149	203	41	119	244	170	72	22
224	233	183	85	11	136	214	52	106	43	117	151	201	74	20	246	168
240	116	42	200	150	21	75	169	247	182	232	10	84	215	137	107	53

**Exemple:** On vient de recevoir les 7 octets suivants :

**00 00 04 0C 38 F0 01**

Le CRC que l'on va calculer est initialisé au départ à la valeur 00.

On part du code de famille qui est le 1<sup>er</sup> octet: 01

On fait le OU Exclusif entre le CRC et l'octet soit : h'00' XOR h'01'.

Le résultat est h'01' soit d'01'. Cette valeur est l'index de la table qui va donner le CRC . On trouve à l'index 01 de la table la valeur décimale 94 soit h'5E'. Donc CRC = 5E.

On fait maintenant le OU Exclusif entre le nouveau CRC et le 2eme octet. h'5E' XOR h'F0' donne : h'AE' soit 174 en décimal. Cette valeur d'index nous donne dans la table un nouveau CRC calculé de 176 soit h'B0'.

Le OU Exclusif avec le 3eme octet donne : h'B0' XOR h'38' = h'88' soit l'index décimal de 136 qui donne dans la table: CRC = d'78' ou h'4E'.

Le OU Exclusif avec le 4eme octet donne : h'4E' XOR h'0C' = h'42' soit l'index décimal de 66 qui donne dans la table: CRC = d'250' ou h'FA'.

Le OU Exclusif avec le 5eme octet donne : h'FA' XOR h'04' = h'FE' soit l'index décimal de 254 qui donne dans la table: CRC = d'107' ou h'6B'.

Le OU Exclusif avec le 6eme octet donne : h'6B' XOR h'00' = h'6B' soit l'index décimal de 107 qui donne dans la table: CRC = d'69' ou h'45'.

Le OU Exclusif avec le 7eme octet donne : h'45' XOR h'00' = h'45' soit l'index décimal de 69 qui donne dans la table: CRC = d'121' ou h'79'.

Le CRC des 7 octets reçu est donc h'79' qu'il faudra comparer au 8<sup>eme</sup> octet que l'on va recevoir.

## CLE iBUTTON DS1990

Ces circuits sont utilisés comme clé d'accès. Ils possèdent un code unique sur 64 bits comme tous les circuits 1 Wire.

Le code famille est h"01", suivi de 6 octets propre au circuit et d'un octet de CRC.

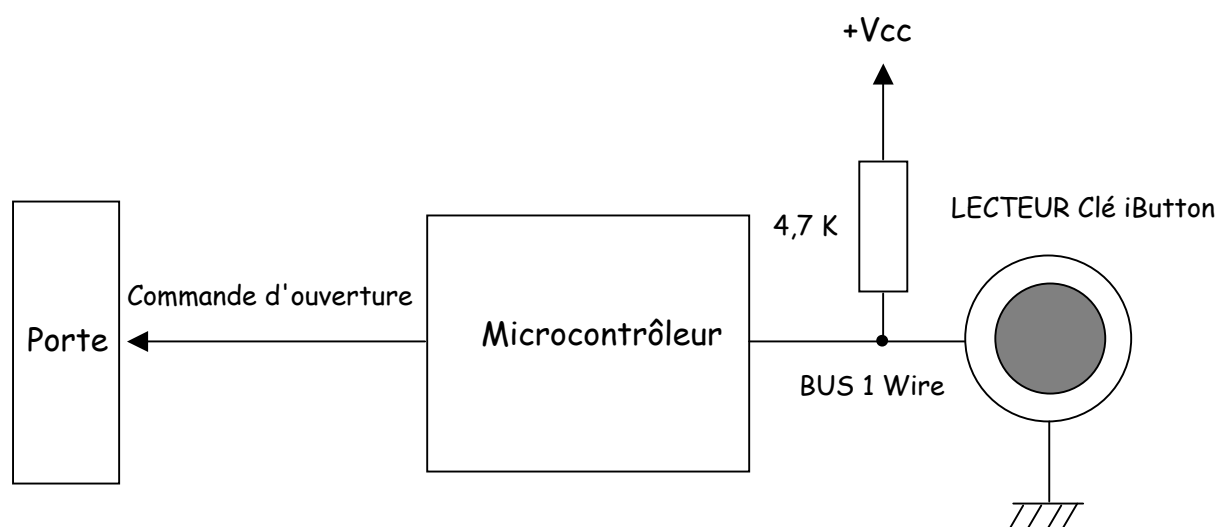
Le lecteur de clé est raccordé au bus 1 Wire.

La détection de présence de ce circuit se fait en envoyant le pulse de Reset, qui est un état bas pendant au moins 480  $\mu$ s.

Quand une clé est présente sur le bus elle le signale en maintenant le bus à l'état bas pendant 60 à 240  $\mu$ s.

Le maître devra donc en permanence chercher à détecter la présence d'une clé. Quand il aura trouvé une clé sur le bus 1 Wire, il enverra la commande Read ROM ( h'33'). En réponse la clé enverra ces 64 bits. Le maître devra calculer le CRC à partir des 56 premiers bits reçus et le comparer au CRC que la clé a envoyé avec les 8 derniers bits. En cas d'erreur une nouvelle lecture s'impose.

Quand la réception des 8 octets sera bonne, c'est à dire que le CRC calculé sera le même que le CRC reçu, le maître pourra comparer ces 64 bits à ceux contenus dans une table, et autoriser ou non l'ouverture de la porte à accès contrôlé.





## CAPTEUR de TEMPERATURE DS 18B20

Ces circuits possèdent un code unique sur 64 bits comme tous les circuits 1 Wire.

Le code famille est h"28", suivi de 6 octets propre au circuit et d'un octet de CRC.

La détection de présence de ce circuit se fait en envoyant le pulse de Reset, qui est un état bas pendant au moins 480  $\mu$ s.

Quand un circuit DS 18B20 est présent sur le bus il le signale en maintenant le bus à l'état bas pendant 60 à 240  $\mu$ s.

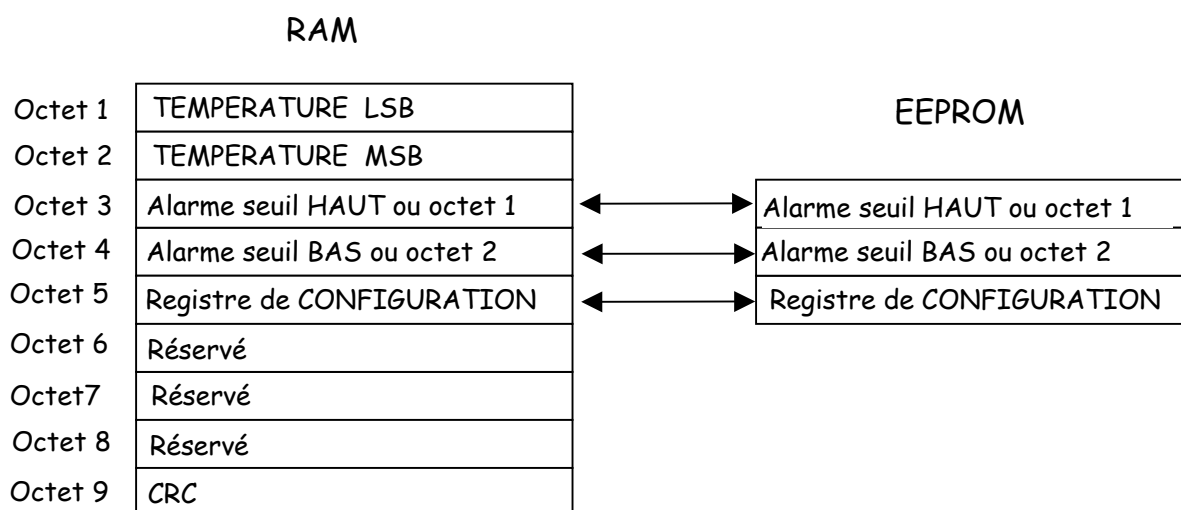
Toute transaction avec un tel circuit doit démarrer par un pulse de Reset suivi de l'envoi d'une commande ROM. On pourra après envoyer une commande de fonction propre à ce type de circuit.

Si le circuit est seul sur le bus 1 Wire, la commande ROM peut être l'appel général SKIP ROM = h"CC". Si ce n'est pas le cas, il faudra connaître les 64 bits propre du circuit que l'on veut atteindre et utiliser la commande MATCH ROM = h'55" suivi des 8 octets du code.

Une recherche préalable des 8 octets de code sera faite par la commande READ ROM = h'33' si le circuit est seul ou bien par SEARCH ROM = h"FO" s'il y a plusieurs circuits sur le bus.

### MEMOIRE INTERNE:

Elle est constituée d'une zone RAM de 9 octets et d'une zone EEPROM non volatile de 3 octets.



## OCTET 1 et 2: TEMPERATURE LSB et MSB

La température est donnée sur 16 bits en complément à 2 entre -55 °C et + 125 °C.

**S** = Signe de la température. Ce bit est à "1" si elle est négative et à "0" si elle est positive.

Si le signe est positif (bit S=0) la valeur absolue de la température sera donnée par les bits significatifs de LSB et MSB .

Par contre si la température est négative ( bit S=1), la valeur absolue sera obtenue en complémentant la valeur des bits significatifs de LSB et MSB et en ajoutant 1 au résultat.

	Bit 7							Bit 0
LSB	$2^3=8$	$2^2=4$	$2^1=2$	$2^0=1$	$2^{-1}=0,5$	$2^{-2}=0,25$	$2^{-3}=0,125$	$2^{-4}=0,0625$
	Bit 15							Bit 8
MSB	S	S	S	S	S	$2^6=64$	$2^5=32$	$2^4=16$

## OCTET 5: Registre de CONFIGURATION

	Bit 7							Bit 0
Config	0	X1	X0	1	1	1	1	1

Seuls deux bits sont significatifs dans ce registre: X1 et X0. Ces bits permettent de choisir la résolution.

X1	X2	RESOLUTION	Temps conversion
0	0	9 bits	94 ms
0	1	10 bits	188 ms
1	0	11 bits	375 ms
1	1	12 bits	750 ms

En résolution 11 bits, le bit 0 n'est pas défini. Le bit de poids min est le bit 1 de valeur : 0,125 °C.

En résolution 10 bits, les bit 0 et 1 ne sont pas définis. Le bit de poids min est le bit 2 de valeur 0,25 °C.

En résolution 9 bits, les bit 0, 1 et 2 ne sont pas définis. Le bit de poids min est le bit 3 de valeur 0,5 °C.

### **CODES de COMMANDES:**

Après avoir envoyé une commande ROM pour adresser un DS18B20 esclave, le maître doit envoyer un code de commande.

#### **Début de conversion: h'44'**

Cette commande lance la conversion de température. Le résultat est rangé dans les 2 octets LSB et MSB . Le temps de conversion dépend de la résolution choisie.

Le maître doit interroger le DS18B20 qui répond par un bit à "0" tant que la conversion n'est pas terminée.

Quand l'opération est terminée, l'esclave répond par un bit à "1".

#### **ECRITURE en RAM: h'4E'**

Seuls les octets 3, 4 et 5 de la zone RAM peuvent être écrits. Il s'agit des octets: Alarme seuil haut, Alarme seuil bas et Configuration.

Le maître doit commencer par envoyer en premier le LSB de l'octet 3. Tous les octets seront ensuite envoyés avec le LSB en tête.

Il doit impérativement envoyer les 3 octets, avant de faire un reset, pour que l'écriture soit effective.

#### **LECTURE de la RAM: h'BE'**

Les 9 octets de la RAM sont envoyés vers le maître. L'esclave commence par le bit 0 du premier octet et transmet ainsi les 9 octets de sa RAM.

Le maître peut interrompre à tout moment la lecture en faisant un Reset.

#### **COPIE RAM en EEPROM: h'48'**

Copie des octets 3, 4 et 5 de la zone RAM dans la zone EEPROM pour sauvegarde en cas de coupure d'alimentation.

### RECOPIE EEPROM en RAM: h'B8'

Cette commande récupère en EEPROM les octets Alarme seuil haut, Alarme seuil bas et Configuration pour les placer en RAM dans les octets 3, 4 et 5.

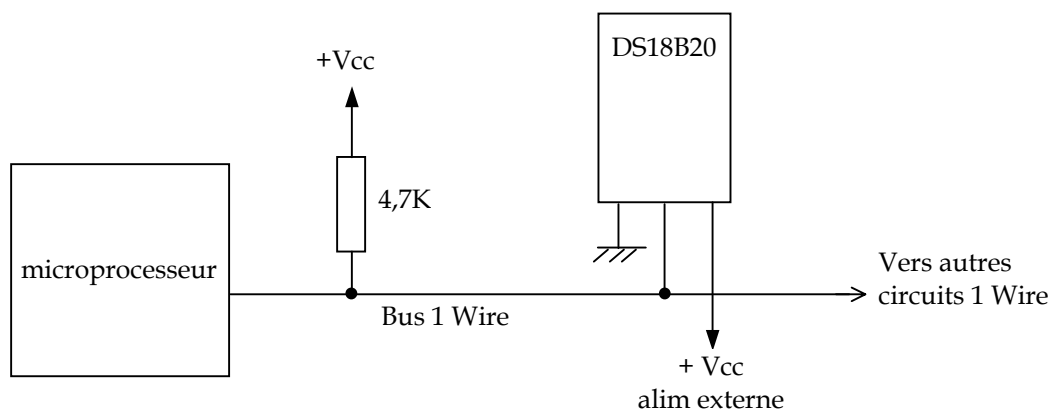
### ALIMENTATION PARASITE: h'B4'

L'esclave répond à cette commande par un bit à "1" s'il fonctionne avec une alimentation extérieure, c'est à dire sur 3 fils. Et par le bit à "0" s'il est en mode d'alimentation parasite, sur 2 fils.

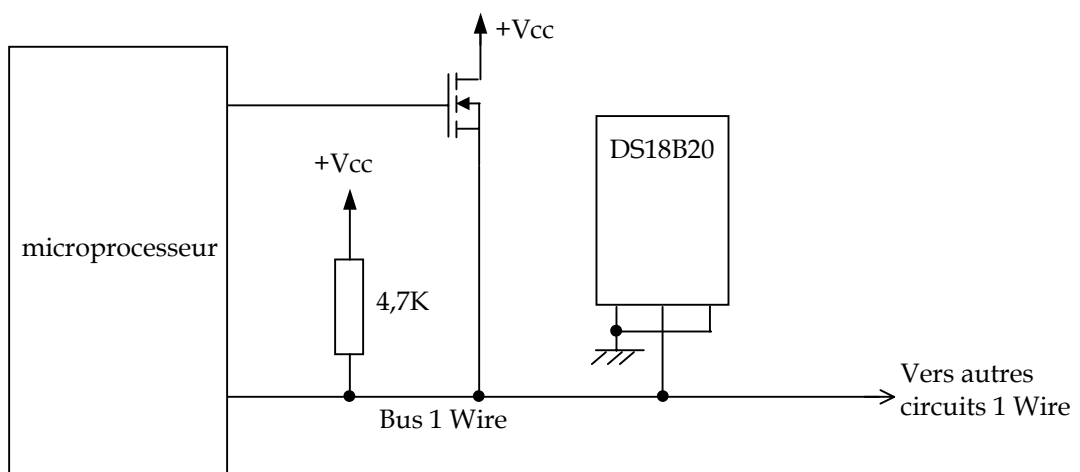
### ALIMENTATION du DS 18B20:

Il y a 2 méthodes pour alimenter ce circuit.

- Soit par une alimentation extérieure entre 3V et 5V :



- Soit par une alimentation parasite :



Quand le bus 1 wire est à l'état haut, une capacité interne se charge et sert de réservoir pour alimenter le circuit quand le bus est à l'état bas. Dans ce cas la broche Vcc du circuit est raccordée à la masse.

Seulement, quand le circuit est en mode conversion ou en mode écriture en EEPROM, le courant d'alimentation doit être d'au moins 1,5mA. Dans ce cas l'alimentation parasite ne suffit plus.

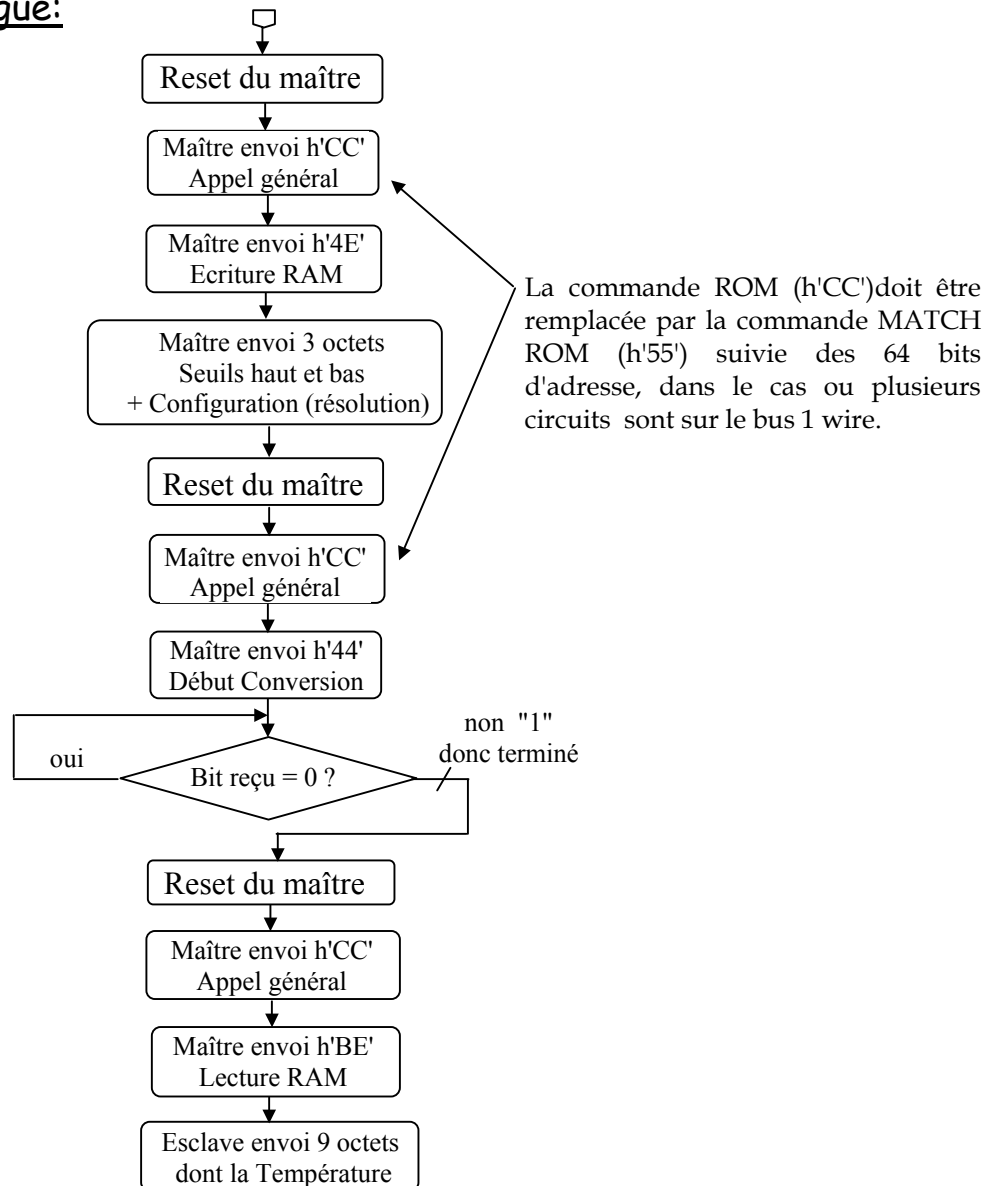
Il faut pour cela mettre le bus 1 Wire directement au +Vcc. C'est ce qui sera fait par la mise en conduction du transistor MOSFET.

Cette opération devra intervenir dans les 10  $\mu$ s qui suivent l'envoi de la commande "Conversion Température" ou "Copie RAM en EEPROM".

Ensuite le bus devra rester au +Vcc pendant toute la durée de la conversion de température (750 ms) ou de l'écriture en EEPROM (10ms).

Aucune autre activité sur le bus 1 Wire ne peut avoir lieu pendant ce temps là, à la différence du mode d'alimentation extérieure, où ce temps peut être mis à profit pour envoyer d'autres données sur le bus.

Exemple de dialogue:



## PORT 1 SORTIE DS 2405

Ces circuits possèdent un code unique sur 64 bits comme tous les circuits 1 Wire. Le code famille est h"05", suivi de 6 octets propre au circuit et d'un octet de CRC.

La détection de présence de ce circuit se fait en envoyant le pulse de Reset, qui est un état bas pendant au moins 480  $\mu$ s. Quand un circuit DS 2405 est présent sur le bus il le signale en maintenant le bus à l'état bas pendant 60 à 240  $\mu$ s comme tout circuit répondant au protocole 1 Wire.

Tout dialogue doit débuter par ce pulse de détection de présence ou Reset envoyé par le maître.

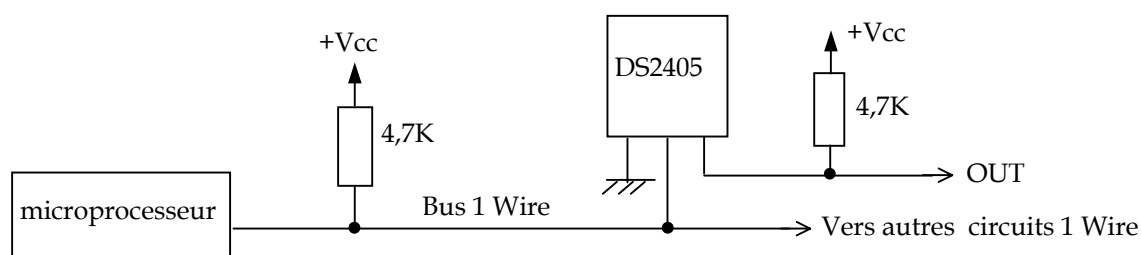
Ensuite le maître doit envoyer une commande ROM.

- Si c'est la commande "Read ROM" = h'33', le circuit répondra ces 64 bits de code et attendra un pulse de reset. Tant que le maître n'aura pas envoyé ce pulse, le DS2405 enverra, si on le lit, un bit à "1" et retournera en attente du pulse de reset.

- Si c'est la commande "Match ROM" = h'55', suivie des 64 bits propres au circuit, le DS2405 qui s'est reconnu fait changer d'état sa sortie (Toggle) et attend un Reset. Si on lit le circuit avant de faire un reset, le DS2405 répond un bit qui est l'état du port. Il reviens ensuite en attente de Reset ou de lecture de son bit d'état.

- Si c'est la commande "Skip ROM" d'appel général = h'CC', le DS2405 répondra avec son bit d'état toujours à "1" tant qu'il n'aura pas reçu le pulse de Reset.

- Si c'est la commande "Search ROM" = h'F0', le maître va éliminer bit par bit les autres esclaves. Le DS2405 qui restera, aura donc reconnu ces 64 bits propres et renverra si on le lit un bit qui sera l'état de son port, comme pour la commande "Match ROM". La commande "Active Search ROM" = h'EC' fonctionne de la même manière, mais seuls les DS2405 qui ont leur Port à l'état bas peuvent être sélectionnés par la recherche bit à bit. Le bit lu sera donc "0".



## PORT 8 ENTREES/SORTIES DS 2408

Ces circuits possèdent un code unique sur 64 bits comme tous les circuits 1 Wire. Le code famille est h"29", suivi de 6 octets propre au circuit et d'un octet de CRC.

La détection de présence de ce circuit se fait en envoyant le pulse de Reset, qui est un état bas pendant au moins 480  $\mu$ s. Quand un circuit DS 2405 est présent sur le bus il le signale en maintenant le bus à l'état bas pendant 60 à 240  $\mu$ s comme tout circuit répondant au protocole 1 Wire.

Tout dialogue doit débuter par ce pulse de détection de présence ou Reset envoyé par le maître.

Le maître doit ensuite envoyer une commande ROM, qui pourra être suivie d'une commande de fonction.

### Commandes de fonction

#### Lecture registres : h'F0'

Cette commande doit être suivie de l'adresse du registre interne à atteindre. L'octet LSB de l'adresse est envoyé en premier suivi de l'octet MSB de cette adresse. Le maître lit alors l'octet pointé par l'adresse, et à chaque lecture il lit l'octet suivant jusqu'à l'adresse h'008F'. Le maître peut envoyer à tout moment un Reset 1 wire pour sortir de cette commande.

Adresse	Type	REGISTRE
h '0000' à h '0087'	x	Non défini
h '0088'	Read	Etat logique du PIO
h '0089'	Read	Registre d'état de Latch du PIO
h '008A'	Read	Registre d'état d'activité du PIO
h '008B'	Read/Write	Masque de sélection pour recherche conditionnelle
h '008C'	Read/Write	Sélection de polarité pour recherche conditionnelle.
h '008D'	Read/Write	Registre CONTROLE/STATUS
h '008E' à h '008F'	Read	Non définis. Toujours à h'FF'

### Lecture PIO : h'F5'

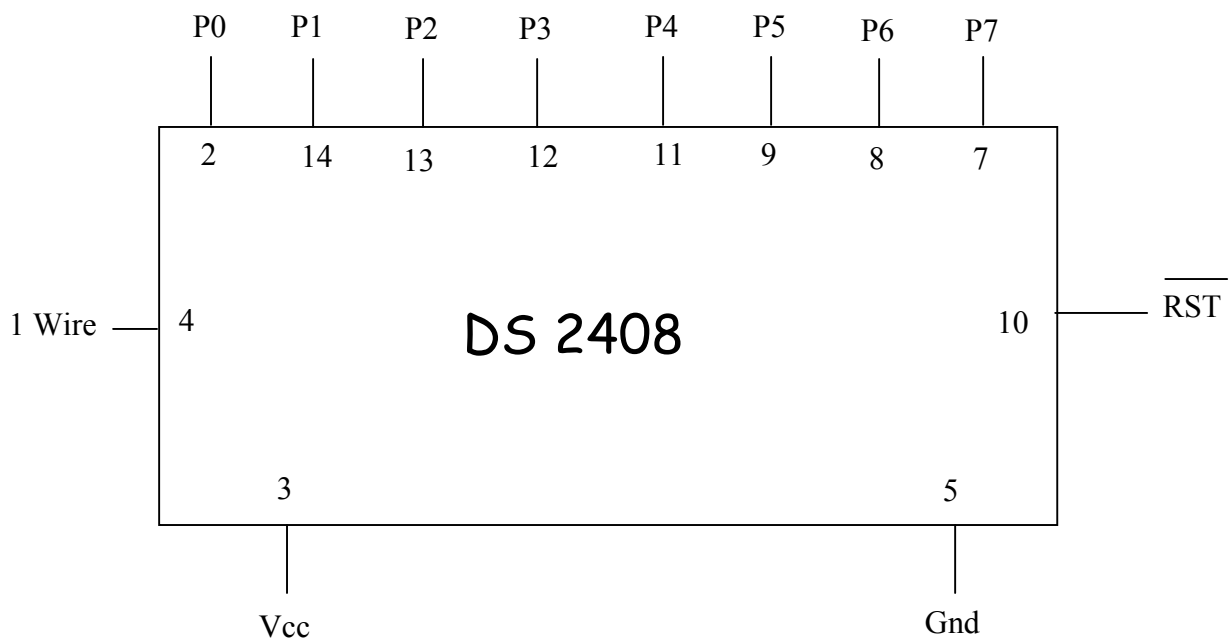
A la différence de la commande de lecture des registres suivie de l'adresse h'0088', cette commande renvoie pendant 32 fois l'état du PIO suivi d'un octet de CRC, qui va permettre au maître de vérifier que la transmission s'est effectuée sans erreur. Cette commande peut être terminée à tout moment par l'envoi d'un Reset 1 Wire.

### Ecriture PIO : h'5A'

C'est la seule possibilité que l'on a, pour écrire dans le Latch de sortie du PIO. Le maître doit envoyer la DATA après cette commande puis il doit envoyer la DATA complémentée. Si la transmission est bonne, le DS2408 répond par un bit à "0". Dans le cas contraire ce bit est à "1", et un Reset 1 Wire est nécessaire pour sortir de cette commande.

Si la transmission a été bonne, le DS2408 écrit la DATA reçue sur le PIO. Il envoie ensuite l'octet h'AA' pour confirmer qu'il n'y a pas eu d'erreur et tout de suite après, il lit le PIO et renvoie la valeur, pour permettre au maître de vérifier ce qu'il a écrit. Cette commande peut être terminée à tout moment par un Reset 1 Wire.

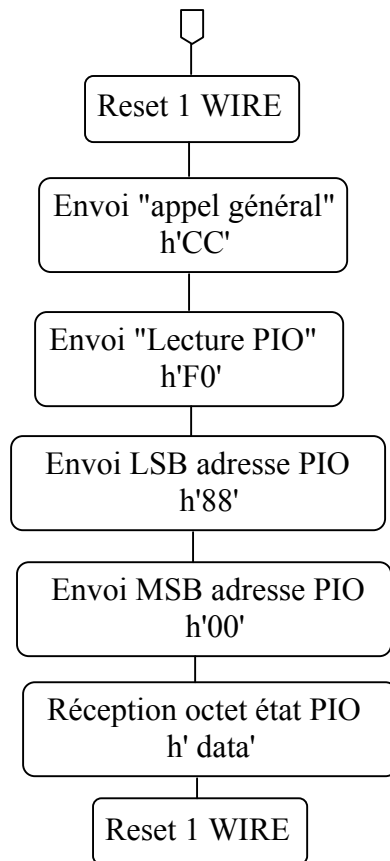
### Brochage :



La broche 10 ( Reset ) doit être tirée au + Vcc en fonctionnement normal.



**LECTURE du PIO :**



**ECRITURE dans le PIO :**

