

Sujet de contrôle – systèmes embarqués

J.-M Friedt, 18 janvier 2014

A society, most of whose members spend a great part of their time, not on the spot, not here and now and in the calculable future, but somewhere else, in the irrelevant other worlds of sport and soap opera, of mythology and metaphysical fantasy, will find it hard to resist the encroachments of those who would manipulate and control it.
A. Huxley, *Brave New World Revisited* (1958)

1 Opérations arithmétiques

Le logiciel embarqué dans un des systèmes électroniques commercialisés par la société SENSEOR s'exécute sur un processeur ARM Cortex-M3, une architecture dont les registres internes sont codés sur 32 bits.

Au lieu de faire une analyse rigoureuse des résolutions nécessaires pour un calcul de racine carrée pour obtenir un module de complexe nécessaire lors d'une évaluation d'un coefficient d'une série de Fourier, une solution triviale consiste à définir des variables de type `long long` afin d'éliminer tout risque de dépassement de capacité. Cette solution est certes fiable puisque le module s'obtient comme la racine de la somme de deux carrés de nombres, chacun défini comme `long`, mais n'est pas sans conséquence. En effet, en observant la taille du binaire résultant de la compilation, prêt à être transféré dans le microcontrôleur, nous constatons une augmentation de près de 5800 octets lors du passage de `long` à `long long` du calcul de racine :

98812 main.bin
104400 main.bin

Une analyse du code assembleur implémentant le calcul de \sqrt{A} ($A \neq 0$) comme la valeur vers laquelle la suite $x_{n+1} = \frac{1}{2}(x_n + x_n/A)$ converge, indique que la seule différence dans les fonctions que nous avons programmé se trouve dans la racine carrée :

Version avec `long long` :

```
int jm_sqrt(long long A)
{long long x=0, xp=0;
8011ca8: b590      push {r4, r7, lr}
8011caa: b087      sub sp, #28
8011cac: af00      add r7, sp, #0
8011cae: e9c7 0100 strd r0, r1, [r7]
8011cb2: f04f 0200 mov.w r2, #0
8011cb6: f04f 0300 mov.w r3, #0
8011cba: e9c7 2302 strd r2, r3, [r7, #8]
8011cbe: f04f 0200 mov.w r2, #0
8011cc2: f04f 0300 mov.w r3, #0
8011cc6: e9c7 2304 strd r2, r3, [r7, #16]
if (A!=0)
8011cca: e9d7 2300 ldrd r2, r3, [r7]
8011cce: ea52 0103 orrs.w r1, r2, r3
8011cd2: d03a      beq.n 8011d4a <jm_sqrt+0xa2>
    {xp=(A/2); // initialisation : A/2
8011cd4: e9d7 0100 ldrd r0, r1, [r7]
8011cd8: ofcc      lsr.s r4, r1, #31
8011cda: 4622      mov r2, r4
8011cdc: 2300      movs r3, #0
8011cde: 1812      adds r2, r2, r0
8011ce0: eb43 0301 adc.w r3, r3, r1
8011ce4: 105b      asrs r3, r3, #1
8011ce6: ea4f 0232 mov.w r2, r2, rrx
8011cea: e9c7 2304 strd r2, r3, [r7, #16]
do {
    x=xp;
8011cee: e9d7 2304 ldrd r2, r3, [r7, #16]
8011cf2: e9c7 2302 strd r2, r3, [r7, #8]
    xp=(x+A/x)/2; // x(n+1)=1/2*(x(n)+A/x(n))
8011cf6: e9d7 0100 ldrd r0, r1, [r7]
8011cfa: e9d7 2302 ldrd r2, r3, [r7, #8]
8011cfe: f000 fb2b bl 8012358 <__aeabi_ldivmod>
8011d02: 4602      mov r2, r0
8011d04: 460b      mov r3, r1
8011d06: 4610      mov r0, r2
8011d08: 4619      mov r1, r3
8011d0a: e9d7 2302 ldrd r2, r3, [r7, #8]
8011d0e: 1880      adds r0, r0, r2
8011d10: eb41 0103 adc.w r1, r1, r3
8011d14: ofcc      lsr.s r4, r1, #31
8011d16: 4622      mov r2, r4
8011d18: 2300      movs r3, #0
8011d1a: 1812      adds r2, r2, r0
8011dic: eb43 0301 adc.w r3, r3, r1
8011d20: 105b      asrs r3, r3, #1
8011d22: ea4f 0232 mov.w r2, r2, rrx
8011d26: e9c7 2304 strd r2, r3, [r7, #16]
    } while ((x-xp)>1);
8011d2a: e9d7 0102 ldrd r0, r1, [r7, #8]
8011d2e: e9d7 2304 ldrd r2, r3, [r7, #16]
8011d32: 1a82      subs r2, r0, r2
8011d34: eb61 0303 sbc.w r3, r1, r3
8011d38: f04f 0001 mov.w r0, #1
8011d3c: f04f 0100 mov.w r1, #0
8011d40: 4290      cmp r0, r2
8011d42: eb71 0c03 sbcs.w ip, r1, r3
8011d46: dbd2      blt.n 8011cee <jm_sqrt+0x46>
8011d48: e005      b.n 8011d56 <jm_sqrt+0xae>
}
else xp=0;
8011d4a: f04f 0200 mov.w r2, #0
8011d4e: f04f 0300 mov.w r3, #0
8011d52: e9c7 2304 strd r2, r3, [r7, #16]
return((int)xp);
8011d56: 693b      ldr r3, [r7, #16]
}
```

Version avec `long` :

```
int jm_sqrt(long A)
{long x=0, xp=0;
8011c90: b480      push {r7}
8011c92: b085      sub sp, #20
8011c94: af00      add r7, sp, #0
8011c96: 6078      str r0, [r7, #4]
8011c98: 2300      movs r3, #0
8011c9a: 60bb      str r3, [r7, #8]
8011c9c: 2300      movs r3, #0
8011c9e: 60fb      str r3, [r7, #12]
if (A!=0)
8011ca0: 687b      ldr r3, [r7, #4]
8011ca2: 2b00      cmp r3, #0
8011ca4: d016      beq.n 8011cd4 <jm_sqrt+0x44>
    {xp=(A/2); // initialisation : A/2
8011ca6: 687b      ldr r3, [r7, #4]
8011ca8: ofda      lsr.s r2, r3, #31
8011caa: 18d3      adds r3, r2, r3
8011cac: 105b      asrs r3, r3, #1
8011cae: 60fb      str r3, [r7, #12]
do {
    x=xp;
8011cb0: 68fb      ldr r3, [r7, #12]
8011cb2: 60bb      str r3, [r7, #8]
    xp=(x+A/x)/2; // x(n+1)=1/2*(x(n)+A/x(n))
8011cb4: 687a      ldr r2, [r7, #4]
8011cb6: 68bb      ldr r3, [r7, #8]
8011cb8: fb92 f2f3 sdiv r2, r2, r3
8011cbc: 68bb      ldr r3, [r7, #8]
8011cbe: 18d3      adds r3, r2, r3
8011cc0: ofda      lsr.s r2, r3, #31
8011cc2: 18d3      adds r3, r2, r3
8011cc4: 105b      asrs r3, r3, #1
8011cc6: 60fb      str r3, [r7, #12]
    } while ((x-xp)>1);
8011cc8: 68ba      ldr r2, [r7, #8]
8011cca: 68fb      ldr r3, [r7, #12]
8011ccc: 1ad3      subs r3, r2, r3
8011cce: 2b01      cmp r3, #1
8011cd0: dcee      bgt.n 8011cb0 <jm_sqrt+0x20>
8011cd2: e001      b.n 8011cd8 <jm_sqrt+0x48>
}
else xp=0;
8011cd4: 2300      movs r3, #0
8011cd6: 60fb      str r3, [r7, #12]
return((int)xp);
8011cd8: 68fb      ldr r3, [r7, #12]
}
```

On rappelle que pour identifier la composante à fréquence $n/N \in [0 : 1]$ d'une série temporelle x comprenant N points, le coefficient de Fourier n correspondant s'obtient par $\sum_{k=0}^{N-1} x_k \exp(2\pi \frac{n \cdot k}{N})$

1. Sur combien de bits est codé le produit de deux nombres entiers codés sur 32 bits chacun ?
2. Sur combien de bits est codé la somme de N nombres entiers codés sur 32 bits chacun ?
3. Identifier la cause de l'augmentation de la taille du code proposé ci-dessus et commenter ?
4. Rappeler l'expression du coefficient de Fourier de fréquence nulle et son sens physique. Comment évolue ce coefficient si la valeur moyenne du signal est non-nulle (nombre de bits nécessaires à encoder ce coefficient) ?
5. Lors du calcul des coefficients de Fourier sur N points de valeur moyenne M , en déduire une précaution à prendre sur la taille des données en entrée ? Quel palliatif proposer pour éviter ce problème ?

2 Opérations logiques

Un compteur Geiger génère des impulsions pour chaque particule ionisante qui interagit avec le gaz présent entre deux électrodes polarisées par une tension suffisamment élevée pour induire une avalanche de particules secondaires détectées sous forme d'une impulsion de courant aux bornes du condensateur ainsi formé. En ce sens, un compteur Geiger s'apparente à un dispositif numérique puisque l'impulsion de courant dépend uniquement de la présence de la particule ionisante, indépendamment de sa nature ou de son énergie.

Afin de s'affranchir du bruit de fond électrique et ajouter une information de direction quant à la particule incidente, la méthode de **coïncidence** (http://www.luisabonolis.it/Bonolis/CV_Italian_files/BotheRossiCoincidence.pdf dont l'utilisation a donné lieu au prix Nobel 1954 à ses auteurs) consiste à analyser la sortie de deux compteurs Geiger spatialement séparés (typiquement d'une distance de l'ordre du décimètre), en considérant que seule une réponse "simultanée" des deux compteurs est représentative du passage d'une particule provenant d'une direction correspondant à l'alignement des deux détecteurs.

Au repos, le potentiel en sortie de chaque compteur Geiger est positif (+5 V), pour induire une impulsion négative (entre 0 et -5 V) brève (typiquement 50 μ s) lors du passage d'une particule.

1. Quel circuit logique permet de combiner les signaux des deux compteurs décrits dans le texte pour obtenir un signal unique représentatif de la coïncidence ?
2. Comment implémenter cette fonction logique avec des transistors uniquement ?
3. Proposer une stratégie pour tester le bon fonctionnement de ce circuit, et en particulier vérifier que les transistors sélectionnés pour la mise en œuvre expérimentale sont assez rapides.
4. Bien qu'un compteur Geiger nécessite une tension élevée (typiquement de l'ordre de 300 V) pour accélérer les ions formés par l'ionisation d'un gaz situé entre les deux électrodes d'un condensateur, le courant n'est généré que par la transition d'état selon le même principe que le fonctionnement des portes CMOS d'un processeur. Quelle est la consommation d'un tel circuit alimenté sous 3 V au travers d'un convertisseur DC-DC et comptant 1000 coups/minute, le condensateur étant formé de deux plaques planes parallèles de 1 cm² de surface est séparés de 1 cm. Pour rappel, la constante diélectrique du vide est $\epsilon_0 = 8,85 \cdot 10^{-12}$ F/m
5. Un tel circuit peut-il être alimenté par un port USB de PC ? Si l'impulsion générée par chaque particule dure 50 μ s, que se passe-t-il si le nombre de particules détectées atteint plus de 10⁶ coups/s. Quelle est la consommation électrique correspondante ?

3 Autonomie électrique d'un circuit : les supercondensateurs

Les supercondensateurs proposent une solution intéressante de stockage d'énergie car présentant une usure à la charge inférieure aux accumulateurs chimiques (batteries), tout en fournissant une capacité de décharge rapide pour alimenter brièvement un circuit gourmand en courant. Ils sont en particulier utilisés

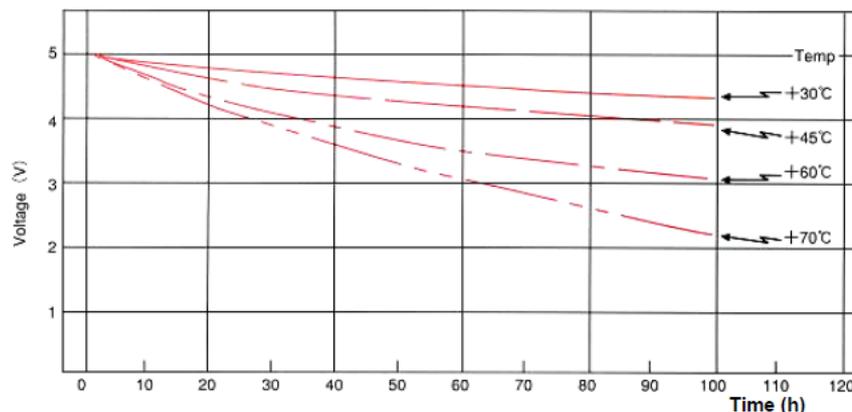
comme tampon d'énergie dans les applications visant à récupérer de l'énergie de diverses sources de l'environnement du capteur (*energy harvesting*) que peuvent être les thermopiles (différence de température), vibrations, cellules photovoltaïques, énergie électromagnétique radiofréquence ambiante ...

Nous nous intéressons à quelques caractéristiques des supercondensateurs et leur conséquence dans l'autonomie d'un circuit ainsi alimenté, et en particulier leur courant de fuite. En effet, l'auto-décharge d'un supercondensateur est un handicap majeur puisque les électrons chèrement acquis de l'environnement se dissipent sans alimenter la charge utile qu'est le circuit que nous désirons alimenter. En consultant par exemple <http://industrial.panasonic.com/www-data/pdf/ABC0000/ABC0000TE5.pdf> (p.27) nous constatons que la tension aux bornes d'un supercondensateur évolue en l'absence de toute charge de 5 V (tension de charge) à 3 V (tension minimum d'utilisation) en 100 h.

Influence of ambient temperature on self-discharging characteristics

Charging condition: 5V, 24hours

Part number: EECS0HD104V (5.5V 0.1F)



(Note) In the self-discharge characteristics, the terminal voltage drop is affected by ambient temperature. This means a self-discharge current becomes great as ambient temperature rises. In case where it is used with a micro applied current of nano-ampere order, ambient temperature allows a difference to occur in back-up time.

FIGURE 1: Auto-décharge par courant de fuite d'un supercondensateur (extrait de <http://industrial.panasonic.com/www-data/pdf/ABC0000/ABC0000TE5.pdf>)

1. quel est le courant de fuite de ce condensateur lorsqu'il fonctionne autour de 60°C (Fig. 1) ?
2. les supercondensateurs ont le mauvais goût de ne fonctionner qu'à basse tension de charge. Quel est l'effet sur le courant de fuite de placer plusieurs condensateurs en série ?
3. si nous désirons augmenter la capacité équivalente du circuit de charge, nous placerons plusieurs supercondensateurs en parallèle : quel est l'effet sur le courant de fuite observé ? sur l'autonomie ?
4. un supercondensateur de 1 F chargé sous 5 V alimente un circuit consommant 15 μA en moyenne et nécessitant une tension de fonctionnement de 3 V au travers d'un régulateur linéaire nécessitant une tension d'entrée de 3,3 V pour fonctionner. Quelle est l'autonomie du circuit ?
5. un supercondensateur de 1 F chargé sous 5 V alimente un circuit consommant 15 μA en moyenne et nécessitant une tension de fonctionnement de 3 V au travers d'un convertisseur DC-DC nécessitant une tension d'entrée de 3,3 V pour fonctionner. Quelle est l'autonomie du circuit en négligeant cette fois le courant de fuite ?

TSVP

4 Questions de cours

1. Exprimer 0xdd en décimal.
2. Exprimer 254 en hexadécimal.
3. Quel est le pas de quantification (en tension) d'un convertisseur analogique-numérique de 8 bits de résolution de tension de référence 5,12 V ?
4. comment copier un fichier `a` dans l'emplacement `b` sous unix ?
5. comment définir une variable non-signée codée sur 8 bits en C ?
6. comment créer un tableau `tab` de 12 éléments en C ?
7. comment accéder au dernier élément de ce tableau ?
8. quelle est la fonction principale appelée lors de l'exécution d'un programme C ?
9. qu'affiche le programme

```
int main() {char c[10]={1,2,3,4,72,'e','l','l','o',0};printf("%s\n",&c[4]);}
?
```

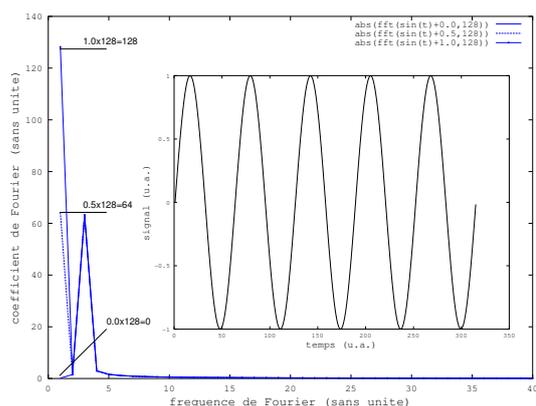
10. proposer un chronogramme de la transmission de l'octet 0x55 en format 8N1 (8 bits de données, pas de bit de parité, 1 bit de d'arrêt) par liaison RS232. En quoi cet octet est-il spécial ?

Solutions

J.-M Friedt, 18 janvier 2014

1 Opérations arithmétiques

1. Le produit de deux nombres est codé sur la somme du nombre de bits sur lequel chaque nombre est codé. Le produit de deux nombres sur 32 bits est donc codé sur 64 bits.
2. La somme de N nombres codés sur M bits chacun est codée sur $M + \log_2(N)$ bits.
3. Une architecture ARM possède une division comme instruction supportée par le processeur sur des arguments de 32 bits (fonction `sdiv` dans le code assembleur proposé). Lors du passage à 64 bits, une telle fonction matérielle n'existe plus et le programme doit faire appel à la bibliothèque mathématique (fonction `__aeabi_ldivmod` dans le code assembleur proposé, et toute la collection de dépendances qui vont avec). L'augmentation de la taille du code est due à l'inclusion de ces fonctions additionnelles.

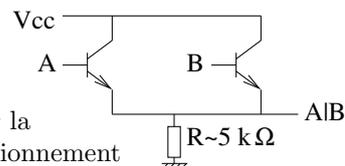


4. Le coefficient de Fourier de fréquence nulle représente la valeur moyenne du signal : $\sum x = \langle x \rangle$. En l'absence de facteur de normalisation, cette somme croît comme N le nombre d'éléments de la somme. Si la valeur moyenne est une nombre encodé sur M bits, alors la somme est encodée sur $M + \log_2(N)$ bits. On se convaincra en testant sous GNU/Octave de la véracité de cette affirmation (varier la valeur moyenne du signal et varier le nombre de coefficients de la transformée de Fourier – second argument de la fonction `fft` de Matlab ou GNU/Octave)
5. une première solution pour éviter ce problème consiste à normaliser les données d'entrée x pour garantir que $M + \log_2(N)$ tient sur les 32 bits disponibles lors d'un calcul sur des `long`, c'est à dire que la valeur moyenne du signal est inférieure à 2^M où M vérifie $M = 32 - \log_2(N)$ pour une transformée de Fourier sur N éléments. Un palliatif plus efficace, puisque nous sommes rarement intéressés dans la valeur moyenne du signal mais plutôt dans ses composants spectrales de fréquence non-nulle, est de retrancher avant tout calcul la valeur moyenne du signal. Ce problème est plus grave dans le cas de la transformée de Fourier rapide (FFT) que dans le cas de la transformée de Fourier discrète (DFT) car alors que dans le second cas chaque coefficient se calcule indépendamment de ses voisins (et l'erreur ne porte que sur un coefficient que nous pouvons décider d'ignorer), la FFT obtient sa vitesse de la mutualisation des calculs pour tous les coefficients et un dépassement sur un calcul de coefficient a toutes les chances de se propager sur les autres coefficients et de rendre tout le calcul faux.

2 Opérations logiques

1. Au repos, le tube du compteur Geiger est à l'état haut, et on ne veut déclencher un signal à l'état bas que si les deux compteurs passent simultanément à l'état bas : une telle fonction est fournie par une port OU.

2. Deux transistors en parallèle fournissent une telle fonctionnalité :



3. En court-circuitant les deux entrées A et B et en les branchant sur la sortie d'un compteur Geiger, on pourra se convaincre du bon fonctionnement du circuit.

4. La capacité est $C = \epsilon_0 \frac{A}{d}$ avec $A = 10^{-4} \text{ m}^2$ et $d = 10^{-2} \text{ m}$ donc $C \simeq 0,1 \text{ pF}$. Puis chaque transition d'état induit une consommation d'énergie de $1/2 \cdot CU^2$ donc 45 nJ, ou pour 1000 transitions une énergie de 45 μJ . Ces 45 μJ sont consommés en 60 s donc 0,75 μW = 750 nW. Que l'on considère cette puissance avant ou après le convertisseur DC-DC ne change rien.

- un port USB peut typiquement fournir 500 mA sous 5 V ou 2,5 W : un port USB peut donc aisément alimenter ce circuit. Si le nombre de particules est tel qu'il n'y a plus de transition (1 million de particules/minute correspond à 17000 coups/s proche de l'inverse de la durée de l'impulsion de 50 μ s) et donc la consommation électrique du système devient nulle, le condensateur reste continuellement déchargé!

3 Autonomie électrique d'un circuit : les supercondensateurs

- $q = CU$ et $i = \frac{dq}{dt} = C \frac{dU}{dt}$ donc si U varie de 2 V en 100 h, le courant de fuite est $i = 0,1 \times \frac{2}{100 \times 3600} = 0,6 \mu\text{A}$.
- le courant est constant dans une chaîne de composants en série donc le courant de fuite reste le même quelquesoit le nombre de condensateurs placés en série. La tension est augmentée par la mise en série des condensateurs mais la capacité équivalente est divisée par le nombre d'éléments.
- les courants se somment sur des branches parallèles se rejoignant à un nœud d'un circuit électrique donc les courants de fuite se somment. Néanmoins, la capacité est multipliée par N en plaçant N condensateurs en parallèle et le courant de fuite est lui aussi multiplié par N donc l'autonomie a néanmoins été améliorée de ce même facteur.
- un régulateur linéaire fonctionne à courant constant donc les 15 μA consommés par le circuit se somment aux 6 μA du courant de fuite (on est passé d'un condensateur de 0,1 à 1 F) et $I = 21 \mu\text{A}$ induisent une chute de tension de $\Delta U = 5 - 3,3 = 1,7 \text{ V}$ en $t = \frac{C}{I} \cdot \Delta U = 81000 \text{ s} = 22,5 \text{ h}$
- un condensateur concentre une énergie $E = \frac{1}{2}CU^2$ et la puissance est la dérivée dans le temps de l'énergie. Le convertisseur DC-DC travaille à puissance constante P donc $\Delta E = \frac{1}{2} \cdot (U_{init}^2 - U_{fin}^2) = P\Delta t$ ou $\Delta t = \frac{1}{2P} \cdot (U_{init}^2 - U_{fin}^2)$ avec $P = 3 \times 15 \cdot 10^{-6} = 45 \mu\text{W}$ et $\Delta t = 43,5 \text{ h}$.
Une seconde approche consiste à considérer le courant moyen consommé entre les deux conditions, ce qui revient à approximer l'exponentielle décroissante de la tension localement par une fonction affine (développement au premier ordre). Dans ce cas la puissance P nécessite initialement un courant $I_{ini} = P/U_{ini} = 45 \cdot 10^{-6}/5 = 9 \mu\text{A}$ pour passer en fin de vie du condensateur à $I_{fin} = P/U_{fin} = 45 \cdot 10^{-6}/3,3 = 13,6 \mu\text{A}$. Le courant moyen consommé est donc de 11,3 μA , et nous revenons au calcul précédent de calcul du temps de décharge du condensateur sur ce courant moyen $t = \frac{C}{I} \cdot \Delta U = 41,7 \text{ h}$. L'intérêt de cette seconde approche est de permettre de tenir compte du courant de fuite (qui n'était pas demandé dans la question) en prenant comme courant dans la dernière phase du calcul la somme du courant moyen et du courant de fuite. Dans ce cas, $t = \frac{C}{(I)+I_{fuite}} \cdot \Delta U = 27,3 \text{ h}$ en prenant un courant de fuite $I_{fuite} = 6 \mu\text{A}$.

4 Questions de cours

- 0xddd=221
- 254=0xfe
- $5,12/2^8 = 0,02 \text{ V} = 20 \text{ mV}$
- cp a b
- unsigned char
- type tab[12];
- tab[11];
- main ()
- Hello : l'affichage commence au 4ème élément, c'est à dire le caractère de code ASCII 72 (ou 'H'), et affiche la chaîne de caractères jusqu'à atteindre la valeur 0 indiquant sa fin (format %s de printf()).
- la transmission se fait par le bit de poids le plus faible (LSB) en premier, donc la transmission commence par le passage du niveau haut à bas pour le START bit, puis alterne entre haut et bas pour les 8 bits qui suivent avant de repasser au niveau haut pour le STOP bit. Cette transmission est spéciale car elle alterne à la fréquence de transmission de chaque bit l'état du bus de communication et permet donc de simplement identifier le débit de communication s'il est inconnu.

