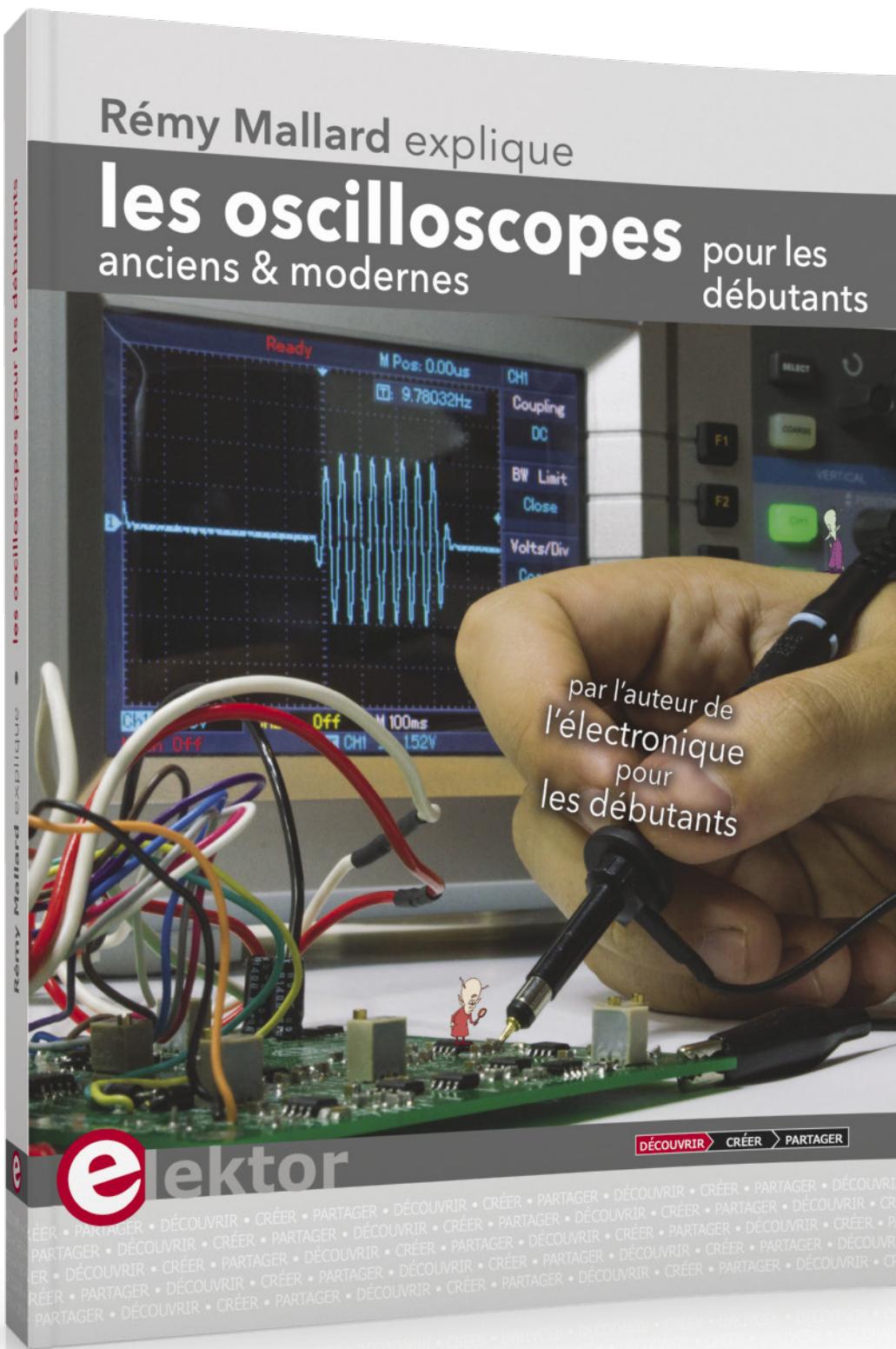


Livre « Oscilloscopes anciens et modernes » de Rémy Mallard



Pour rendre compte du contenu impressionnant du livre de Rémy Mallard sur les oscilloscopes paru récemment, il nous a semblé intéressant d'en reproduire un chapitre pour nos lecteurs. En accord avec l'auteur, nous avons retenu un extrait représentatif du ch. 6 (voir la table des matières complète ci-dessous pour situer l'extrait). Il traite de l'analyse de protocole et du décodage de données série à l'aide d'un oscilloscope. L'idée de la citation d'un long passage est bonne en elle-même, mais comporte en germe le risque de donner du contenu une impression faussée. En effet, comment un extrait aussi condensé pourrait-il rendre compte d'un livre aussi riche ? Les paragraphes de la section 6.7 condensés ici en quelques pages comptent 13 pages en réalité !

6.7 – Analyse de protocole et décodage de données série (I²C, SPI, UART, etc.)

(I²C, SPI, SMBUS, UART, CAN/LIN, I²S, USB 2.0, DMX, etc.)

Dans ce chapitre, nous verrons comment « mesurer », « valider » et « décoder » des trames de données numériques véhiculées sur un bus de données série tel que RS232, I²C ou SPI. Les données peuvent être transmises sur une liaison de type asymétrique (RS232, I²C ou SPI par exemple) ou sur une liaison de type différentielle (RS485, USB ou Ethernet par exemple), et nous devrons identifier les ressources nécessaires pour ces deux types de liaison. [...] Si les signaux électriques objets de l'analyse sont trop déformés, ou si vous indiquez à l'oscilloscope qu'il va recevoir des données USB alors que vous lui fournissez un signal de type SPI, il y a fort à parier que le pauvre ne va pas s'y retrouver ! Dans ce chapitre, nous aborderons ces deux points. Une fois la validité électrique établie, nous verrons comment « décoder » des signaux numériques émis sur des bus de données tels que bus I²C ou SPI, de façon manuelle ou de façon automatique (avec les outils de décodage automatique inclus dans les oscilloscopes numériques Rigol DS1054Z et Picoscope 2204).

6.7.1 – Vérification de la validité électrique des signaux

Ce chapitre est consacré à la mesure de signaux numériques, qui permet de vérifier la validité de trames de données série (amplitude globale et forme des transitions). Il peut sembler curieux de parler de mesures sur un bus de données numériques, car le terme « mesure » pourrait faire penser à « tension » ou « fréquence ». Pourtant, une analyse de données en vue d'un décodage ne peut être fiable que si la voie de transmission (la liaison) est fiable ; les données doivent être assez « propres » pour être correctement interprétées.

Tous les signaux électriques que l'on peut observer sur un écran d'oscilloscope sont par nature analogiques. Un signal qu'on dit numérique ne fait pas exception à la règle, **c'est** un signal analogique. Si on l'appelle signal numérique, c'est parce qu'on se contente d'utiliser un nombre réduit de valeurs de tension, parmi toutes les valeurs que ce signal pourrait prendre. Avec des circuits logiques (numériques) TTL par exemple, il est commun de travailler avec les deux valeurs 0 V et +5 V ou 0 V et +3,3 V. Quand la tension de sortie d'un circuit logique passe de 0 V (état logique bas) à +5 V (état logique haut) cela ne se fait pas de façon instantanée, mais de façon progressive (même si la transition est très rapide), nous l'avons vu au chapitre décrivant la méthode de mesure du temps de montée ou de descente d'un signal numérique. [...] Comme pour les signaux analogiques, les signaux logiques sont représentés en fonction du temps. Mais au lieu de stocker des valeurs numériques qui correspondent à l'amplitude des points acquis, on conserve seulement des états logiques binaires (0 ou 1). Comparé à une acquisition analogique effectuée sur 8 bits, cela est beaucoup plus rapide et demande 8 fois moins d'espace de stockage. Dit autrement, un système d'acquisition 8 bits permet l'acquisition et l'affichage simultané de 8 signaux numériques indépendants. Une autre manière encore d'exprimer les choses est de dire que chaque voie numérique possède une résolution verticale (d'amplitude) de 1 bit. [...].

Sur un bus série, on trouve un ensemble d'informations binaires (bits) qui se suivent à la queue leu leu. Ces informations peuvent être simplifiées à l'extrême, par exemple un seul bit transmit pour un ordre marche ou arrêt d'un unique appareil. Bien évidemment, cette solution très simple est également très limitée, et dans la

pratique on rencontrera de très nombreux cas de figure avec transmission d'un ou plusieurs octets (par exemple 8 bits de données éventuellement encadrés de bits de START et STOP). [...] L'écran D de la figure 6.7.1.a met en évidence un problème de taille : le signal numérique montré sous sa forme analogique est

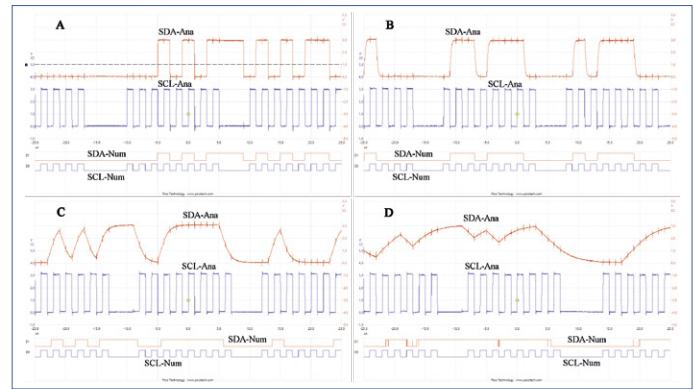


Figure 6.7.1.a – Dans chaque écran A à D, on peut voir un même signal SDA (ligne de données) sous ses formes analogique (SDA-Ana) et numérique (SDA-Num), et un même signal SCL (ligne d'horloge) sous ses formes analogique (SCL-Ana) et numérique (SCL-Num). L'écran A montre des signaux valides, l'écran B montre un signal de données légèrement déformé, et l'écran C montre qu'un signal moyennement dégradé peut encore être exploité. L'écran D quant à lui met en évidence un sérieux problème, le signal de données n'est plus exploitable du tout !

tellement déformé que les valeurs numériques (bits) qu'il transporte ne peuvent plus être correctement reconnues. Regardez attentivement la trace SDA-Num de l'écran D, on y voit à certains endroits des états bas ou hauts très brefs, de durée bien inférieure à une demi-période du signal d'horloge SCL-Num. Cela montre sans ambiguïté que le signal numérique souffre d'hésitations et qu'il n'est pas valide. Le problème est moins évident avec l'écran C, dont l'affichage des données sous forme numérique (SDA-Num) laisse penser que tout va bien, alors qu'en fait on est à la limite tolérée, et que seule la vue sous forme analogique (SDA-Ana) permet de s'en rendre compte.

Remarque : les petits « pics » visibles sur les traces SDA-Ana et SCL-Ana des quatre écrans A à D de la figure 6.7.1.a sont dus à un câblage imparfait du circuit sous test. Au départ, l'auteur voulait les supprimer en améliorant le câblage, mais il a finalement décidé de les laisser, car ils constituent des « marqueurs » qui coïncident avec les transitions d'horloge. Finalement, ces petits pics se révèlent peu gênants et fort pratiques (pour un exercice, pas pour un produit fini) !

[...]

Dans les chapitres qui vont suivre, nous allons étudier plusieurs types de liaisons, certaines sont de type différentiel et d'autres non, certaines sont avec fil d'horloge associé et d'autres non. Voyons de quoi il s'agit.

Une liaison asymétrique (non différentielle) est constituée d'un seul fil électrique pour transmettre des données (dans un sens ou dans un autre) en plus de l'indispensable fil de masse qui sert de référence de potentiel (0 V). [...]

Une liaison symétrique (différentielle) ou une boucle de courant est constituée de deux fils électriques pour transmettre les données (fils Data+ et Data-). Un fil de masse peut y être associé, mais il n'est pas utilisé comme référence de potentiel. [...]

Une liaison sans fil d'horloge associé (dans le câble de liaison) signifie que les données (*Data*) sont transmises seules, sans fil supplémentaire qui pourrait véhiculer un signal de synchronisation : le récepteur qui reçoit les données (*Data*) doit se débrouiller pour reconnaître le début des messages et en extraire les bits d'information. Une telle liaison peut se faire en mode asymétrique (1 seul fil de données) ou en mode symétrique (2 fils de données). [...] **Une liaison avec fil d'horloge associé** possède, en plus du fil de liaison des données (*Data*), un fil supplémentaire pour transporter l'horloge (*Clock*) qui permettra au récepteur de se caler sur les données. Ce mode de liaison permet de s'affranchir des contraintes de vitesse imposées des deux côtés. [...]

6.7.2 – Décodage de données série (généralités)

Remarques préalables :

- L'analyse de données série impose l'usage d'un oscilloscope numérique, qui permet de capturer (mémoriser) une ou plusieurs trames complètes et de les visualiser après coup. Un oscilloscope analogique traditionnel (sans mémoire) ne le permet pas. [...]
- Un oscilloscope numérique possède une certaine quantité (profondeur) de mémoire. De cette quantité de mémoire dépendra la durée maximale de capture d'un ensemble de données numériques. [...]
- La liste des protocoles de données série pris en charge par les oscilloscopes numériques diffère selon leur modèle. Les exemples proposés ci-après correspondent à des modèles d'oscilloscope précis, et si votre appareil est différent vous devrez consulter son manuel utilisateur. [...]

Un analyseur logique est un système qui mémorise une suite de transitions ou de niveaux logiques (bits à 0 ou 1), puis permet ensuite de les afficher sur écran pour analyse fine. Le nombre d'états logiques pouvant être capturés dépend là encore de la quantité (profondeur) de mémoire mise à disposition.

Un analyseur de protocole est une forme évoluée d'analyseur logique. Il permet de reconnaître l'organisation des paquets d'informations binaires capturés (principalement par la détection des début et fin de paquet qui sont facilement reconnaissables) et de les décoder pour afficher en clair leur valeur sous forme numérique, généralement au format décimal ou hexadécimal. [...]

6.7.3 – Décodage de données série I²C

Les échanges de données sur un bus I²C s'effectuent sur « 2 fils » SDA (Serial **D**ata, données) et SCL (Serial **C**lock, horloge). Pour effectuer un décodage de façon manuelle ou automatique, nous avons besoin de visualiser des deux signaux DATA et CLOCK en même temps à l'écran. Heureusement, un oscilloscope à deux voies suffit. [...] Avant d'analyser une capture de données série I²C, rappelons la méthode de transfert d'un point de vue général :

- 1 - Le maître élabore une condition de début de transmission, également appelé bit de start, qui indique aux périphériques que des données vont être envoyées. Ces périphériques se mettent alors en mode d'écoute active.
- 2 - Une adresse codée sur 7 bits est envoyée juste après le bit de start. Chaque périphérique possède une adresse personnelle (différente de celle des autres périphériques) et à cet instant tous les périphériques écoutent puisqu'ils ne savent pas encore à qui s'adresse le message. Une fois les 7 bits d'adresse transmis, un seul périphérique tiendra compte des données qui suivront : c'est celui dont l'adresse personnelle correspond à

l'adresse envoyée par le maître.

- 3 - Un bit « lecture/écriture » est transmis par le maître pour indiquer si l'opération souhaitée est une opération de lecture (bit à 1) ou d'écriture (bit à 0). [...]
- 4 - Selon le type de périphérique I²C sollicité, on peut ensuite avoir transmission de données dans un seul sens (commande) ou dans les deux sens (question/réponse), sur le fil SDA. Les données sont regroupées par paquets de 8 bits, un paquet équivalant à un octet (*byte*). Le nombre d'octets transmis peut être plus ou moins élevé et dépend là encore du type de périphérique I²C. [...]
- 5 - Après transmission (unilatérale ou bidirectionnelle) de tous les octets requis, une condition de fin, également appelée bit de stop, est élaborée. Ce bit de stop indique la fin du message. [...]

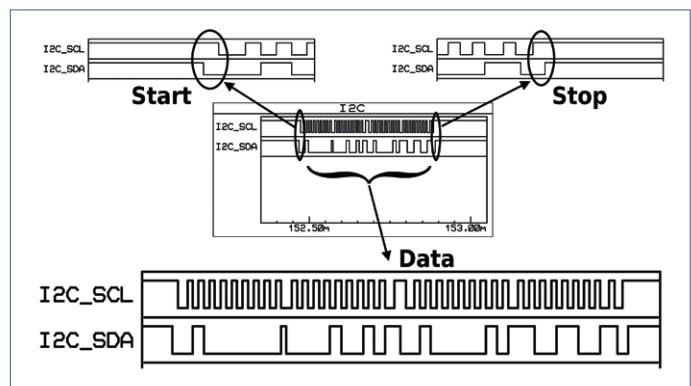


Figure 6.7.3.a – Au repos, les lignes de données SDA et d'horloge SCL sont toutes deux à l'état logique haut. Le bit de start correspond au changement d'état haut-bas de la ligne de données SDA immédiatement suivi du changement d'état haut-bas de la ligne d'horloge SCL. Le bit de stop correspond au changement d'état bas-haut de la ligne d'horloge SCL immédiatement suivi du changement d'état bas-haut de la ligne de données SDA. Les données utiles (Data) sont transmises entre les bits de start et de stop.

L'exemple de la figure 6.7.3.a montre des signaux d'horloge (SCL) régulièrement espacés, les messages I²C ayant été élaborés grâce à un module hard (MSSP) inclus dans un microcontrôleur PIC. Cette grande régularité se retrouve rarement quand les messages I²C sont élaborés de façon purement logicielle avec des *délais*, mais l'absence de non-régularité n'est généralement pas problématique si elle affecte en même temps les deux lignes SDA (*Data*) et SCL (*Clock*).

6.7.3.1 – Décodage manuel de données série I²C

Nous allons utiliser la trame de données visible en figure 6.7.3.a pour nous faire la main, voyons maintenant la figure 6.7.3.1.a qui la décortique en détail.

Pour commencer, nous allons essayer de retrouver les différents paquets de bits (octets) contenus dans la trame visible en figure 6.7.3.1.a, qui correspond à la lecture de l'état des lignes d'entrée/sortie d'un expandeur de port MCP23008. De façon purement visuelle et intuitive, nous constatons qu'il y a 4 octets dans cette trame (4 « gros paquets » de bits/coups d'horloge sur la ligne SCL). Maintenant, nous allons tracer un trait vertical en pointillé en regard de chaque front montant d'horloge sur la ligne SCL. Si en regard de ce trait vertical on découvre un état bas sur la ligne SDA, cela signifie que la valeur de la donnée correspondante est à 0. Si au contraire en face de ce trait vertical on a un état haut

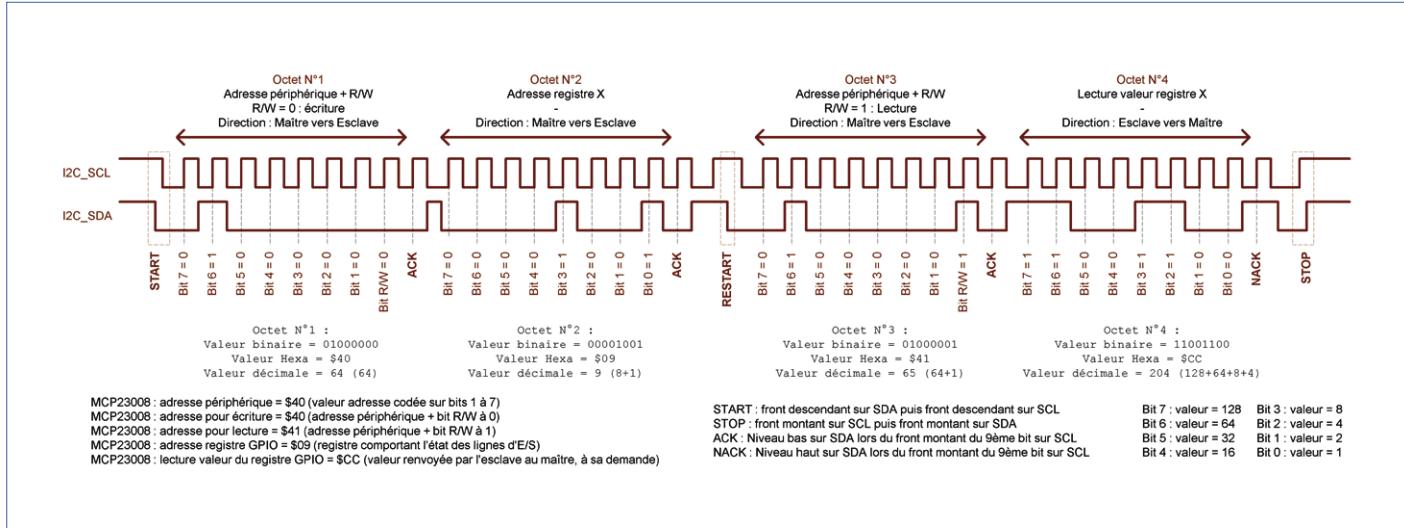


Figure 6.7.3.1.a – Même sans analyseur de protocole, on peut facilement lire la valeur des octets transmis dans une trame de données. Cela demande simplement un peu de temps...

sur la ligne SDA, cela signifie que la valeur de la donnée correspondante est à 1. En analysant de la sorte les 8 bits de chaque paquet, on peut en déduire la valeur de chaque octet (qu'on pourra exprimer en binaire, hexadécimal ou décimal). Le 9e coup d'horloge correspond à l'acquittement (ACK) ou au non-acquittement (NACK) de l'octet complet. C'est pourquoi on trouve 9 coups d'horloge pour chaque octet transmis. Comme vous pouvez le constater, la lecture des octets n'est pas difficile et demande simplement un peu de patience.

6.7.3.2 – Décodage automatique de données série I²C avec Picoscope 2204

L'exercice précédent a montré qu'il n'était pas difficile de décoder des octets bit par bit pour en déduire manuellement leur valeur. Maintenant, nous allons profiter d'une option de décodage de données opérée par l'analyseur de protocole inclus dans le logiciel Picoscope de PicoTech. La figure 6.7.3.2.a montre un exemple de décodage après capture. Les données ont été capturées dans un premier temps, puis dans un second temps le mécanisme de décodage série a été mis en œuvre.

Une fois l'acquisition faite des données à décoder, voici la procédure à suivre :

- Menu Outils > Décodage en série. La fenêtre « Décodage en série » apparaît ;
- Dans l'écran « Décodage en série », presser le bouton « Créer ». Dans la liste des protocoles pris en charge, sélectionner « I²C ». La fenêtre « I²C » apparaît ;
- Dans la zone « Canaux » de la fenêtre « I²C », spécifier l'entrée de l'oscilloscope qui reçoit le signal de données (liste déroulante Data) et spécifier l'entrée de l'oscilloscope qui reçoit le signal d'horloge (liste déroulante Clock). Dans notre exemple, il s'agit des entrées logiques D1 et D0, respectivement.
- Presser le bouton OK des deux fenêtres ouvertes, pour les fermer. Le décodage série démarre et l'on voit le résultat s'afficher à l'écran.

Les octets décodés sont visualisés sous trois formes :

- graphique (le décodage peut se faire depuis les entrées analogiques ou logiques)

- valeurs numériques dans des « paquets » avec libellé « Address » ou « Data »
- table horodatée avec indications détaillées pour chaque octet [...]

La procédure décrite ici est succincte, car l'objet n'était pas de faire un tutoriel. Le manuel utilisateur de PicoTech est suffisant, et il évoluera certainement au fil des versions.

6.7.3.3 – Décodage automatique de données série I²C avec Rigol DS1054Z

[...] Le décodage des données série I²C est assez intuitif, une fois qu'on a compris que les options de décodage sont accessibles par le bouton MATH situé en face avant de l'appareil. Lors de la première utilisation de l'option de décodage, l'auteur est resté perplexe devant les données décodées, qui ne semblaient pas conformes à celles attendues (comparées en parallèle avec le décodage série du Picoscope). Tout fut rentré dans l'ordre une fois que l'auteur eut configuré le mode d'affichage sur HEXA et non plus sur ASCII

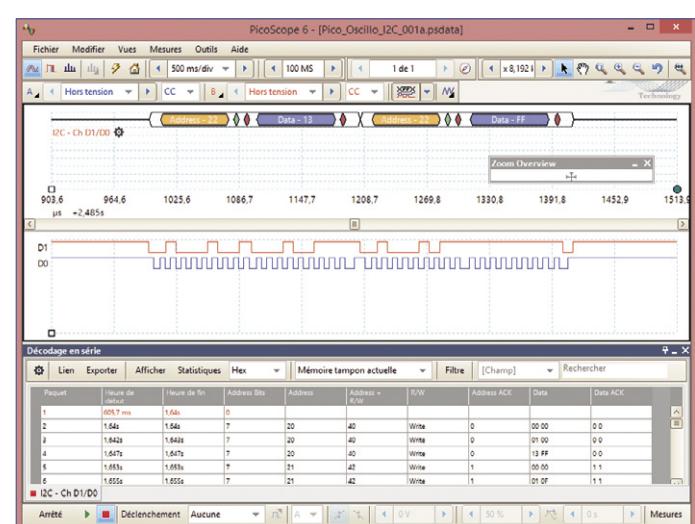


Figure 6.7.3.2.a – Le décodage série d'un flux I²C est très aisés avec l'option de décodage incluse dans le logiciel Picoscope (ici en version V6).

(format proposé par défaut). La figure 6.7.3.3.a montre le résultat d'un décodage opéré sur un flux I²C cadencé à 100 kHz. Les données ont été acquises sur une durée de plusieurs secondes, puis décodées après coup.

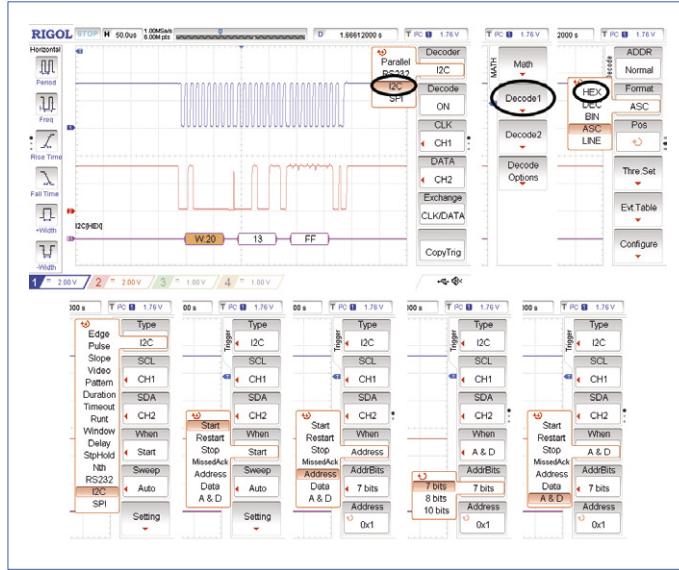


Figure 6.7.3.3.a – L'affichage des données décodées est très clair, et l'on peut choisir le format d'affichage (hexadécimal, décimal, binaire, ASCII). À de rares exceptions, l'auteur préfère le format hexadécimal qui permet une comparaison/vérification plus rapide par rapport au tracé. Les options de déclenchement (Trigger) permettent en outre de définir les conditions de capture de la trame I²C (filtrage sur adresse par exemple).

6.7.4 – Décodage de données série de données SPI

Les échanges de données sur un bus SPI peuvent s'effectuer sur «3 fils» ou «4 fils» :

- Liaison SPI «3 fils» => SDIO ou SDA, CLOCK et CS ou SS ou CE
- Liaison SPI «4 fils» => MOSI, MISO, CLOCK et CS ou SS ou CE.

Le fil d'horloge est toujours sous le contrôle du maître. Pour rappel :

- SDIO = **S**erial **D**ata **I**n **O**ut, (maître vers esclave et esclave vers maître)
- SDA = **S**erial **D**ata (maître vers esclave et esclave vers maître)
- MOSI = **M**aster **O**ut **S**lave **I**n (maître vers esclave seulement)
- MISO = **M**aster **I**n **S**lave **O**ut (esclave vers maître seulement)

Dans les exercices qui suivent, nous utiliserons des données transmises en mode «4 fils», le mode de travail en «3 fils» étant similaire au bus I²C avec son fil de liaison DATA bidirectionnel. Le mode «4 fils» offre l'avantage d'une liaison *full duplex* (bidirectionnelle), le maître et l'esclave peuvent émettre des données et en recevoir en même temps.

Remarque : l'idéal ici est de posséder un oscilloscope à quatre voies, qui permet de montrer simultanément les données MISO, MOSI, CLOCK et CS. Si votre oscilloscope ne possède que deux voies, alors vous ne pourrez observer que les couples MOSI+CLOCK ou MISO+CLOCK. Ne comptez pas trop vous en sortir en visualisant le couple MISO+MOSI, en pratique c'est difficile, voire impossible [...] **Astuce** : quand plusieurs périphériques sont câblés sur un même bus SPI, tous reçoivent en même temps les données véhiculées

par le bus, alors qu'un seul d'entre eux est réellement concerné (ou éventuellement plusieurs, s'ils sont chainés). Si vous voulez capturer des données spécifiques à un seul périphérique, vous pouvez utiliser son fil de sélection de boîtier (CS, SS ou CE) pour déclencher la trace de l'oscilloscope, ou tout du moins utiliser ses changements d'état affichés à l'écran pour repérer «qui est où» (à quel périphérique les données sont destinées). [...]

Quelques faits nouveaux doivent être portés à votre connaissance :

- en SPI, il n'y a pas de bit d'acquittement (ACK) ni de non-acquittement (NACK). On aura donc 8 impulsions d'horloge par paquet (par octet) et non 9 comme en I²C. L'absence d'acquittement implique que le maître peut envoyer des données sans se rendre compte qu'il parle dans le vide (qu'aucun esclave ne lui répond).
- les impulsions du signal d'horloge peuvent être matérialisées par des états hauts avec un état de repos (Idle) bas (CPOL, **Clock POLarity** = 0), ou être matérialisées par des états bas avec un état de repos haut (CPOL = 1).
- l'instant où doit s'opérer l'analyse de la ligne SDA peut se faire au début de l'impulsion d'horloge (CPHA, **Clock PHase** = 0) ou à la fin de l'impulsion d'horloge (CPHA = 1). Pour résumer, il existe 4 Modes possibles pour une liaison SPI :
 - Mode 0 : CPOL = 0 et CPHA = 0
 - Mode 1 : CPOL = 0 et CPHA = 1
 - Mode 2 : CPOL = 1 et CPHA = 0
 - Mode 3 : CPOL = 1 et CPHA = 1

[...]

La figure 6.7.4.a montre les quatre modes utilisables pour la transmission d'un octet de valeur \$47 (valeur binaire 01000111).

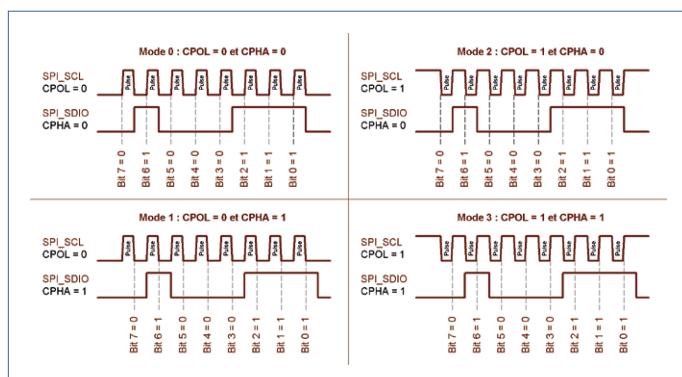


Figure 6.7.4.a – Octet de valeur \$47 (valeur binaire 01000111) transmis selon les 4 Modes possibles d'une liaison SPI. Le mode adopté conditionne la polarité (CPOL) et la phase (CPHA) du signal d'horloge SCL. Quand CPHA = 0, l'état de la ligne de donnée est analysé au début de l'impulsion d'horloge (*leading*). Quand CPHA = 1, l'état de la ligne de donnée est analysé à la fin de l'impulsion d'horloge (*trailing*).

Dans l'exemple donné avec transmission de l'octet de valeur \$47 (figure 6.7.4.a) c'est le bit de poids fort (MSB, Bit 7) qui est transmis en premier. Si le bit de poids faible (LSB, Bit 0) devait être transmis en premier, cela ajouterait 4 nouveaux cas de figure. Qui a dit que la configuration d'un bus SPI n'était pas amusante ?

200098-01

Rémy Mallard explique
les oscilloscopes anciens et modernes pour les débutants
376 pages (en couleur) | 17 x 23,5 cm | ISBN 978-2-86661-208-5

Table des matières

1 – Introduction

- 1.1 – Qu'est-ce qu'un oscilloscope ?
- 1.2 – Fonctionnement de base d'un oscillo analogique
- 1.3 – Caractéristiques de base d'un oscillo
Sensibilité et échelle verticale
Base de temps, balayage et échelle horizontale
Bande passante
Caractéristiques spécifiques aux oscillos numériques

2 – Sondes d'oscilloscope

- 2.1 – Qu'est-ce qu'une sonde ?
- 2.2 – Sécurité de l'utilisateur
- 2.3 – Sonde de tension passive
impédance et capacité
limite en fréquence | en tension
Influence du fil de masse de la sonde
quand utiliser une sonde passive ?
- 2.4 – Sonde de tension active
- 2.5 – Sonde de tension différentielle
limite en tension | en fréquence
isolation galvanique complète
- 2.6 – Autres types de sondes
haute tension | courant
champ proche | contact
- 2.7 – Types de connecteur
- 2.8 – Choix d'une sonde

3 – Choix d'un oscilloscope

- 3.1 – Pas trop vite, papillon...
- 3.2 – Analogique ou numérique ?
- 3.3 – Oscilloscopes de laboratoire, portatifs...
labo | portatifs ou « de terrain » (avec écran)
sur PC | autres types
DSO, MSO, DPO, MDO, DSO (bis)...
- 3.4 – Évaluation des besoins de l'utilisateur
lieu d'utilisation | nombre de signaux
tension et fréquence max. | nature des signaux |
domaine fréquentiel
- 3.5 – Évaluation des performances d'un
oscilloscope
performances générales | résolution et type
d'écran
ergonomie | autres « petits détails »
- 3.6 – Mise à jour du matériel ou du logiciel
(oscilloscope numérique)
mises à jour officielles et non officielles
maintenance et vérification métrologique
(recalibrage)
- 3.7 – Oscilloscope d'occasion : une bonne affaire ?

4 – Configuration de l'oscilloscope

- 4.1 – Installation et position d'utilisation
- 4.2 – Prise en main et visualisation d'une trace

4.3 – Calibrage d'une sonde | LF | HF

- 4.4 – Modes de couplage
DC (continu) | AC (alternatif) | autres
- 4.5 – Modes de déclenchement
NORM, NORMAL | automatique (AT, AUTO, AUTO
SETUP)
mono (MONO-COUP, ONE-SHOT, SINGLE)
source et pente de déclenchement | Autres types
HOLD-OFF (inhibition) | Mode Roll
- 4.6 – Réglages essentiels
Sensibilité | base de temps
- 4.7 – Visualisation simultanée de plusieurs
signaux
traces séparées | traces combinées | calculées
évoluées
- 4.8 – Utilisation des curseurs (marqueurs) de
mesure
- 4.9 – Fonctions « spéciales »
impédance | tension de décalage | voies isolées

5 – Mesures élémentaires

- 5.1 – Mesure d'une tension
continue | alternative | décalage (offset) | faible
- 5.2 – Mesures temporelles
intervalle | période | fréquence | rapport
cyclique
déphasage | montée | descente

6 – Mesures et fonctions évoluées

- 6.1 – Avertissements
- 6.2 – Mesure de la tension du secteur
sonde différentielle | transfo d'isolement |
optocoupleur
- 6.3 – Mesures dans une alimentation secteur
linéaire | découpage | sans transformateur
- 6.4 – Mesure de bruit
- 6.5 – Mesure en XY et figures de Lissajous
déphasage | caractéristiques d'un dipôle
- 6.6 – Valeurs calculées et traces calculées
Valeurs calculées | Courbes calculées
- 6.7 – Analyse de protocole et décodage de
données série
(I2C, SPI, UART, etc.)
validité électrique des signaux
données série (généralités) | I2C | SPI |
UART/RS232 | bus différentiel
- 6.8 – Analyse spectrale
- 6.9 – Mesures et fonctions spécifiques
puissance | impédance d'un dipôle
bande passante (réponse amplitude-fréquence)
jitter (gigue) | diagramme de l'œil
totaliseur/compteur d'événements
générateur de fonction intégré
Interface de communication
écran tactile | commande vocale

enregistrement des données
vérification d'une télécommande à infrarouge
signaux inconnus...

7 – Exercices (et solutions)

8 – Les générateurs de signaux

- 8.1 – Caractéristiques principales
technologie employée | type de signal
fréquence | amplitude du signal
types de sortie et protections
caractéristiques propres aux générateurs
numériques
modulations (AM, FM, sweep)
choix d'un générateur de signaux
- 8.2 – Utilité (et utilisation) d'un générateur de
signal
mesure de bande passante | distorsion
stimulus avec signaux non permanents
comportement d'un étage logique ou analogique
vérification des protections d'un circuit
simulation de capteurs analogiques

Annexe 1 – Sécurité électrique

Annexe 2 – Numérisation d'un signal

- Pourquoi numériser un signal analogique ?
Conversion analogique numérique (numérisation)
Fréquence d'échantillonnage (en hertz)
Quantification (en bits) et résolution
Conversion numérique analogique

Annexe 3 – Glossaire

Annexe 4 – Transformer...

- un téléviseur en oscilloscope
- un oscilloscope en téléviseur...
- un oscilloscope en phasemètre audio
- un oscilloscope en analyseur de spectre

Annexe 5 – Photographie oscilloscope analogique

Annexe 6 – Fabriquer soi-même une sonde ?

- passive | active | différentielle

Annexe 7 – Petits trucs cachés

- bande passante | Résolution verticale
- résolution supérieure à 8 bits avec CAN 8 bits ?
- taux de rafraîchissement (update rate)
- tension d'entrée maximale
- autres...
- limitation de bande passante automatique
- limitation de bande passante par logiciel

Annexe 8 – Liens utiles

Index

Liens et littérature

- [1] Table des matières complète (PDF) : www.elektor.fr/amfile/file/download/file/2103/product/9442/
- [2] Le livre dans la boutique en ligne d'Elektor : www.elektor.fr/les-oscilloscopes-anciens-modernes-pour-les-debutants
- [3] Rémy Mallard explique l'électronique pour les débutants qui sèchent les cours mais ne craignent pas de se brûler les doigts : www.elektor.fr/l-electronique-pour-les-debutants
- [4] Rémy Mallard explique les microcontrôleurs PIC pour les débutants qui veulent programmer sans patauger : www.elektor.fr/les-microcontrôleurs-pic-pour-les-debutants