

Carte d'application pour PIC 16F876 – Bootloader i2c et rs232

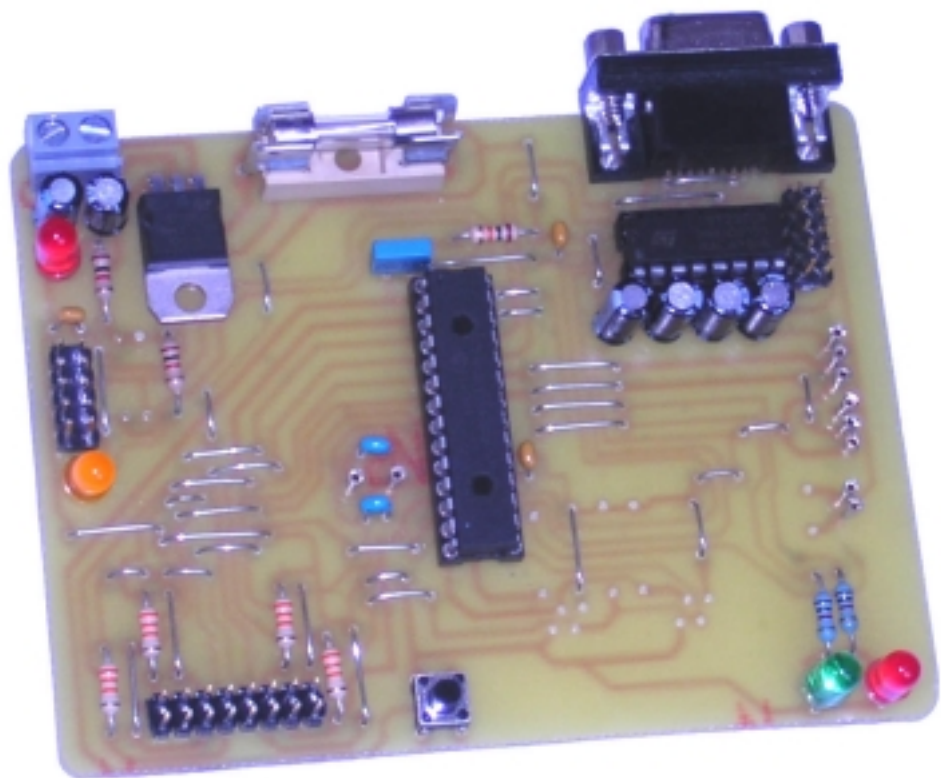


Table des matières 2

La carte 3

- 1. La carte 3
- 2. Les modifications à apporter 3
- 3. Le schéma 4

Bootloaders rs232 et i2c 5

- 1. Le source rs232 5
- 2. Le Downloader rs232 5
- 3. Utilisation 5
- 4. Améliorations 6
- 5. Le source i2c 6
- 6. Le Downloader i2c 7
- 7. Utilisation 7
- 8. Améliorations 8

Liaison RS232 9

- 1. Schéma et modification 9

Clavier 10

- 1. Fonctionnement 10
- 2. Brochage 10
- 3. Caractéristiques électriques 10
- 4. Protocole MF2 10
- 5. Transmission PIC vers Clavier 11
- 6. Programme 12

Joystick 13

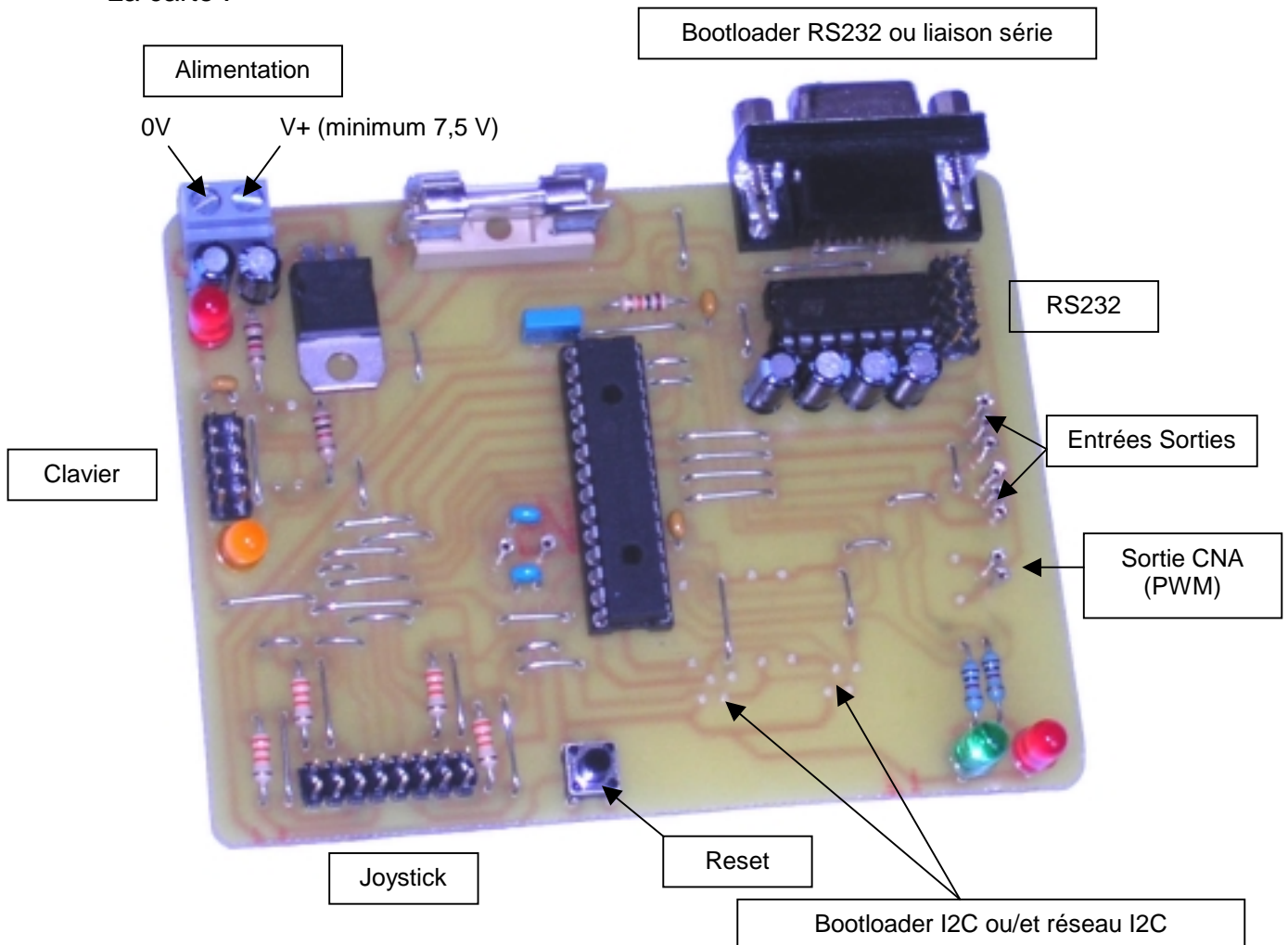
- 1. Schéma/Brochage 13
- 2. Programme 13

I2C 14

- 1. Schéma 14
- 2. Programme 14

La carte, les modifications et le schéma

La carte :

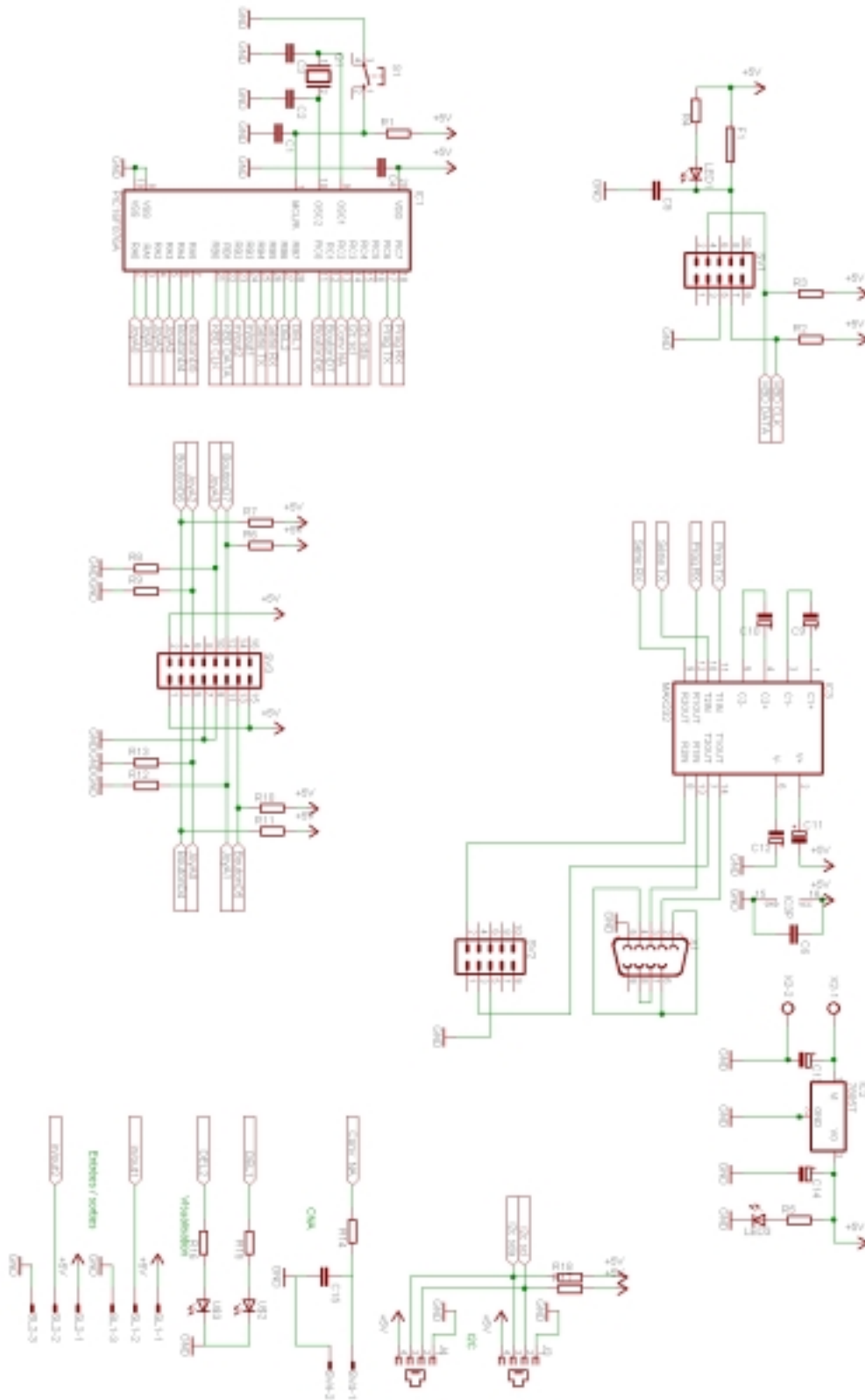


Les modifications à apporter :

- le brochage du joystick utilisé est faux. Il correspond à un ancien type.
- une résistance supplémentaire doit être rajoutée sur le max232 pour que le programme téléchargé dans le PIC démarre automatiquement (en mode Bootloader RS232) si le câble série n'est pas connecté au PC.
- rajouter la gestion de la vidéo et de l'audio.
- rajouter la gestion d'un écran LCD.
- prévoir le connecteur 5 broches de 1^{ère} programmation.

Ces modifications seront détaillées par la suite.

Le schéma :



Bootloader RS232 et I2C

1. Le source RS232

Il est écrit pour l'instant pour un PIC16F876 avec un quartz de 20MHz.

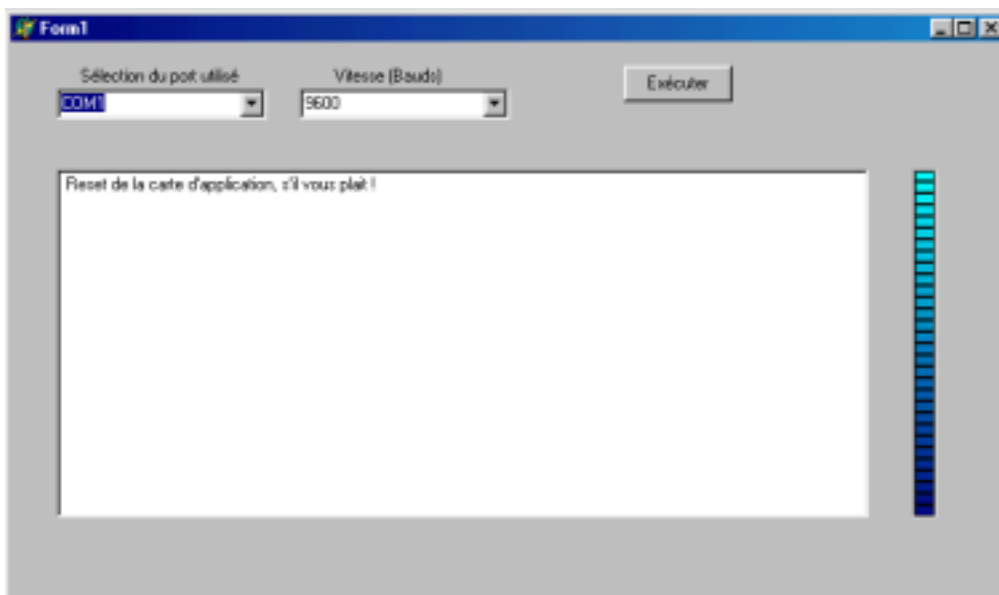
Compilé avec SourceBoost, il se trouve dans le répertoire :

: \ download rs232 \ pack 5 \ Boot v7 et checksum et raccourci moitie OK.

2. Le Downloader RS232

Il se trouve dans le répertoire :

: \ download rs232 \ pack 5 \ Down_232



3. Utilisation

- Programmer le PIC avec le bootloader « Main.hex » avec icprog.
- Le mettre sur la carte de développement.
- Brancher la liaison série entre le PC et la carte de développement sans la mettre sous tension.
- Démarrer Down_232.
- Configurer le port série comme souhaité et appuyer sur exécuter.

- Faire un reset du PIC.
- Un dialogue commence. Taper 1 pour charger un nouveau programme ou 2 pour lancer le programme qui avait ou qui vient d'être téléchargé.
- Après avoir tapé 1, si le PIC est OK, « Pic prêt à recevoir ... » s'affiche. Sélectionner alors le programme avec l'extension .hex à télécharger.
- Lorsque le téléchargement est terminé, le message « OK » est retourné. Si ce n'est pas le cas, recommencer l'opération.
- Faire un reset du PIC et taper 2 pour lancer le programme qui vient d'être téléchargé ou bien débrancher le câble PC rs232 et mettre le montage sous tension et le programme téléchargé démarre automatiquement.

4. Améliorations

Il y a une multiplication par 128 que je n'ai pas changée par << 7.

Il y en a sûrement d'autres.

5. Le source i2c

Il est écrit pour l'instant pour le même PIC16F876 avec un quartz de 20MHz.

Compilé avec SourceBoost, il se trouve dans le répertoire :

: \ download i2c \ pack 4 (ou pack 2 pour moi)

Deux versions sont disponibles :

: \ download i2c \ pack 2 \ Boot i2c avec visu sur liaison série raccourci et checksum OK

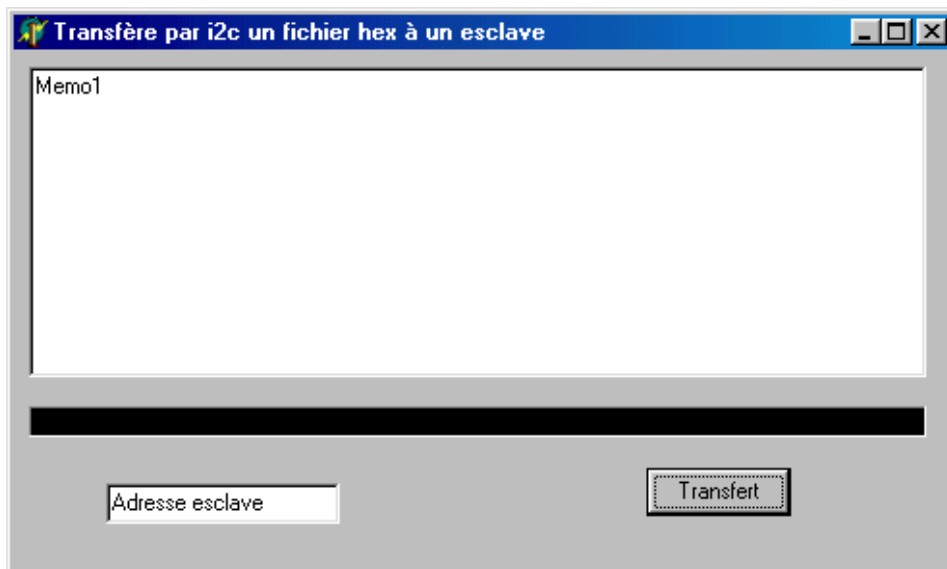
Celle-ci (comme vous le savez tous, j'aime bien les liaisons série) permet de visualiser la transmission qui se fait par l'i2c directement sur la liaison série.

: \ download i2c \ pack 4 \ Boot i2c avec visu sur leds raccourci et checksum OK

Comme son nom l'indique, on se base sur les diodes pour savoir dans quelle phase on est.

6. Le Downloader i2c

C'est le programme dans le répertoire Downloader Delphi i2c nommé Paral_i2c pour l'instant mais j'ai déjà discuté avec Thomas qui va nous écrire un plus joli programme que le mien.



Il est rustique. On tape l'adresse dans Adresse esclave (en décimal) et on appui sur transfert. Il n'est pas très difficile à utiliser, il n'y a qu'un bouton.

7. Utilisation

- Programmer le PIC avec le bootloader « Main.hex » avec icprog.
- Le mettre sur la carte de développement.
- Facultatif : Brancher la liaison série entre le PC et la carte de développement.
- Démarrer Paral_i2c.
- Faire un reset du PIC.
- La diode verte qui clignote indique que le PIC. S'il ne se passe rien (pas d'envoi de l'adresse du bootloader (E0 ici)) pendant ces 2 secondes, le programme qui avait ou qui vient d'être téléchargé démarre. Sinon, si on a cliqué sur « transfert » puis sélectionné un programme .hex, le téléchargement d'un nouveau programme commence.
- Durant la phase téléchargement, les diodes de visualisation clignotent au rythme du téléchargement.

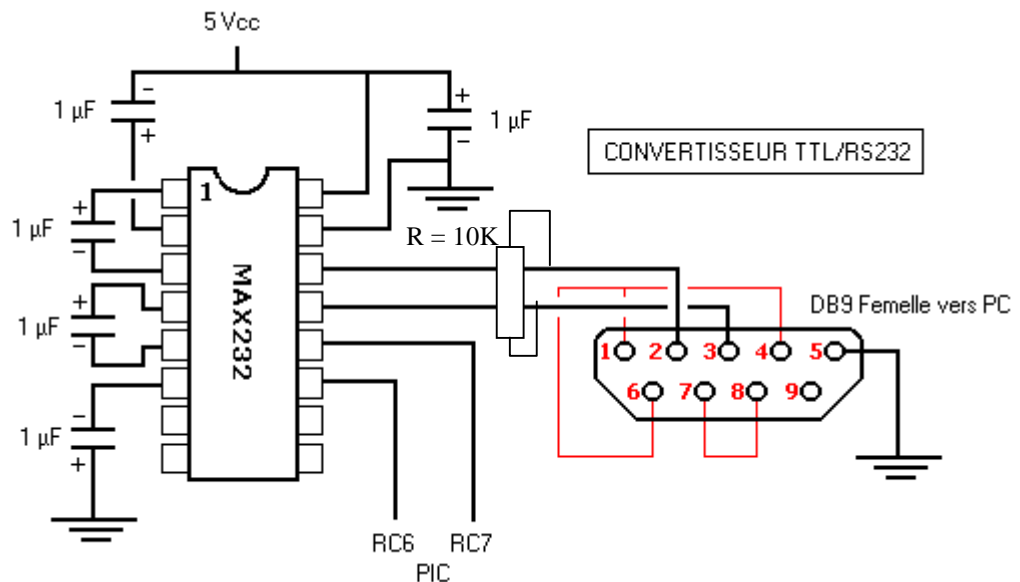
- Lorsque le téléchargement est terminé, s'il s'est bien passé, seule la diode verte est allumée, sinon s'il y a eu un problème de checksum alors c'est la rouge qui est allumée.
- Faire un reset du PIC et attendre 5 secondes, puis boucler la ou c'est nécessaire.

8. Améliorations et debuggage

Ca viendra sûrement en phase de test réel.

Liaison RS232

1. Schéma et modification



La résistance supplémentaire doit être rajoutée sur le max232 pour que le programme téléchargé dans le PIC démarre automatiquement (en mode Bootloader RS232) si le câble série n'est pas connecté au PC.

Dans le Bootloader, la première commande effectuée est envoyer un « D » à destination de la liaison série. S'il est capté par le PIC, (par l'intermédiaire de la résistance de 10K) c'est que le PC n'est pas branché. Sinon si le PC est branché, la résistance n'interfère pas et le bootloader sera lancé.

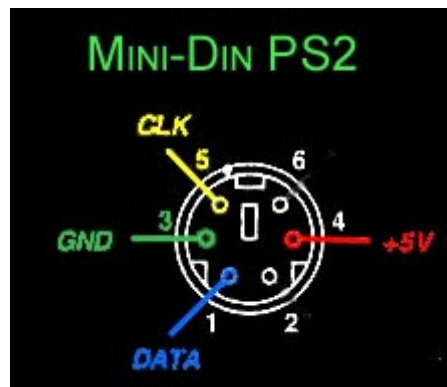
Clavier

1. Fonctionnement

Claviers compatibles PS2.

2. Brochage

Connecteur male clavier PC PS2:



+5V: Alimentation du clavier **GND:** Masse clavier
DATA: Transmission bidirectionnelle des données (KBD_DATA)
CLK: Horloge de synchronisation (KBD_CLK)

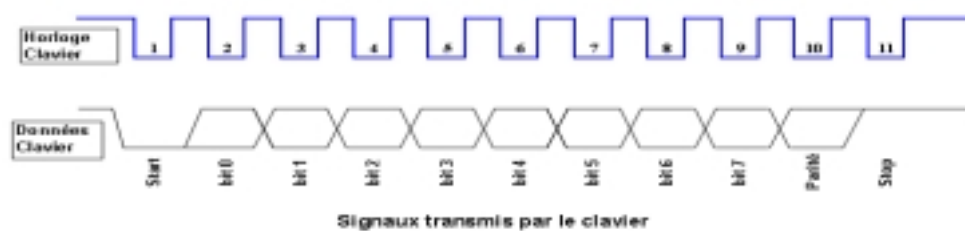
3. Caractéristiques électriques

Les résistances de tirage doivent être présentes.

4. Protocole MF2

Protocole **synchrone bidirectionnel**

Trame:



L'horloge CLK:

Elle est toujours générée par le clavier, elle peut varier entre **10 et 20kHz**

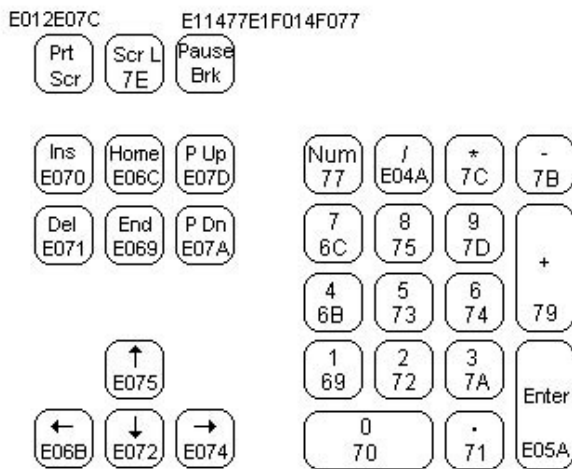
CLK est généré uniquement pendant l'envoi des données et si DATA est à 0, ce qui permettra la détection de présence clavier.

5. Transmission PIC vers Clavier

- Le PIC met DATA à la masse
- Le clavier envoie le signal d'horloge sur CLK.
- Le PIC détecte la présence du clavier et remet DATA en Entrée puis attend de recevoir un code clavier
- Si le PIC reçoit un code clavier il le mémorise et l'associe à son caractère.

Exemple du code de la touche « M » :

Code touche « M » 01001100b ou 4Ch



6. Programme.

Il est écrit pour l'instant pour un PIC16F876 avec un quartz de 20MHz.

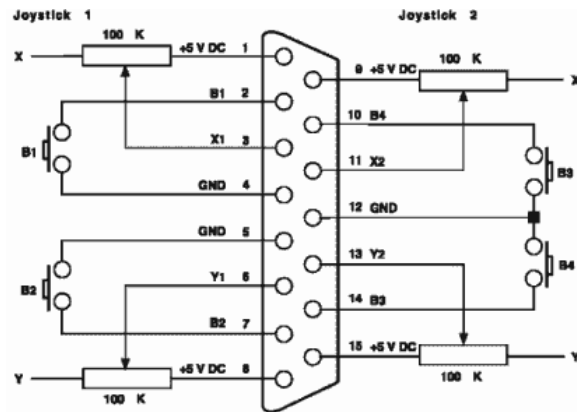
Compilé avec SourceBoost, il se trouve dans le répertoire :

:Application Bootloader\ Clavier\ Clavier et ecriture sur liaison serie

Joystick

1. Schéma/Brochage

Voici le schéma interne de la manette choisie:



et bien, ce n'est pas le bon.

La carte d'application a été modifiée maison (un genre de modification qui ne plairait pas à François) pour que les 3 ponts diviseurs (2 résistances variables, 1 bouton tir) arrivent sur les ports RA0, RA1 et RA2.

2. Programme

Il est écrit pour l'instant pour un PIC16F876 avec un quartz de 20MHz.

Compilé avec SourceBoost, il se trouve dans le répertoire :

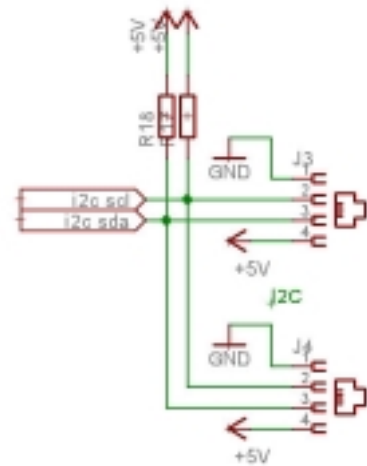
:Application Bootloader\joystick\joystick.

I2C

1. Schéma

Classique, ça marche toujours aussi bien qu'avant, sauf que maintenant, ça marche en plus avec la liaison série. Pour ceux (par exemple Thomas) qui se posent la question du pourquoi du comment, ils n'ont qu'à lire l'excellente documentation du PIC16F876 chapitre 10, et ne pas faire comme moi, pauvre abruti, et se dire que le PIC16F872 et le 876 sont les mêmes à part la mémoire.

Il n'en est rien, le 876 intègre une liaison série sur les broches RC6 et RC7 qui est découplée de l'i2c (en première vérification).



Donc on pourra intégrer des liaisons série partout et je pourrais surtout réécrire le programme de la caméra. Beurk....

2. Programme

Toujours le même, il a juste été « dégraissé » pour le bootloader i2c et s'appelle pic_i2cs. Il se trouve dans :\\Application Bootloader\\ Bootloader i2c\\ et une ou l'autre des versions. Il a été modifié pour fonctionner en polling.