

Antoine MOÏSE, Ali NAJEM
Alexandre PUJOL, Lucas SOLTIC



Projet d'architecture - Bus CAN

Cahier de conception

Client : M. Rethore

Sommaire

Description matérielle

Protocoles utilisés

Méthodes d'analyses utilisées

Conception - Réalisation - Algorithmes

Jeu de tests envisagé

Annexes (Datasheet simplifiés)

Description matérielle

MCP25020 : Expander CAN (I/O Expander ECAN pre-programmé ; système figé)

Programmable jusqu'à 1Mb/s

Tension de fonctionnement : 2.7V à 5.5V

Intensité de fonctionnement : 10mA

Intensité de veille : 30 µA

Température de fonctionnement : -40°C à +85°C

Jeu d'instruction : Risc

PICs du micro-contrôleur : PIC18F6680 & PIC18F8680

Instruction sur 14bits

High current source 25 mA/25 mA

Three external interrupt pins

Tension de fonctionnement : 2.0V à 5.0V

Industrial and Extended temperature ranges

Low power consumption:

- < 1.6 mA typical @ 5V, 4 MHz

- 25 µA typical @ 3V, 32 kHz

- < 0.2 µA typical standby current

MCP2515 : Module The MCU interfaces to the device via the SPI interface.

Programmable jusqu'à 1Mb/s

High-speed SPITM Interface (10 MHz)

Tension de fonctionnement : 2.7V à 5.5V

Intensité de fonctionnement : 5mA

Intensité de veille : 30 µA

Température de fonctionnement : -40°C à +85°C

Jeu d'instruction : Risc

MICROCHIP - DV164005 - ICD2 KIT AVEC MODULE DE BASE + CABLE USB : Débogueur / Programmeur pour envoyer/déboguer le programme sur la carte

Cette carte est équipée de 2 PIC, avec module ECAN intégrés, 2 PIC18. C'est sur cette carte que l'on va avoir la liaison avec l'ordinateur par câble via le protocole RS232. De plus cette carte permet l'acquisition de données (Capteur de température), la commande de processus et l'alarme.

APGDT002 : (sonde analyseur) l'outil d'analyse de bus CAN est une sonde simple à utiliser et peu onéreuse qui permet de contrôler et de déboguer un BUS CAN.

Protocoles utilisés

RS-232

Pour établir une communication effective via RS-232, il est nécessaire de définir le protocole utilisé : notamment, le débit de la transmission, le codage utilisé, le découpage en trame, etc. La norme RS-232 laisse ces points libres, mais en pratique on utilise souvent des UART qui découpent le flux en trames d'un caractère ainsi constituées :

- 1 bit de départ ;
- 7 à 8 bit de données ;
- 1 bit de parité optionnel ;
- 1 ou plusieurs bit d'arrêt.

Le bit de départ a un niveau logique "0" tandis que le bit d'arrêt est de niveau logique "1". Le bit de donnée de poids faible est envoyé en premier suivi des autres.

Ici, nous relierons le PC au PIC18F6680 via le bus RS-232 à l'aide de l'USART. Pour cela nous nous servirons des fonctions indiquées dans la documentation du compilateur C18 MPLAB.

Documents

Documentation des fonctions USART (voir page 70) :

http://ww1.microchip.com/downloads/en/DeviceDoc/MPLAB_C18_Libraries_51297f.pdf

Tutoriel USART : http://wiki.ulcape.org/tutorials:pic:pic18_usart_transmit#open_usart

Calcul du débit en bauds

spbrg : la valeur écrite dans le registre générateur du débit en bauds qui détermine le débit en bauds auquel l'USART fonctionne (this is the value that is written to the baud rate generator register which determines the baud rate at which the USART operates).

FOSC : la fréquence de l'oscillateur, en hertz, ici 25 MHz.

Les formules pour les débits en bauds sont les suivantes :

Mode asynchrone, vitesse élevée : $FOSC / (16 * (spbrg + 1))$

Mode asynchrone, vitesse réduite : $FOSC / (64 * (spbrg + 1))$

Mode synchrone : $FOSC / (4 * (spbrg + 1))$

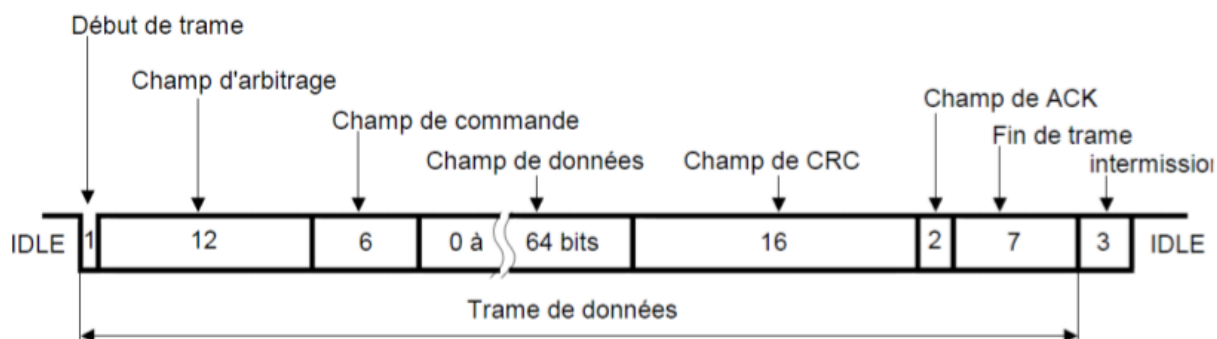
spbrg est le paramètre à donner à la fonction OpenUSART() pour mettre en place la liaison RS-232 (http://wiki.ulcape.org/tutorials:pic:pic18_usart_transmit#open_usart).

Pour le projet, on utilisera une transmission synchrone.

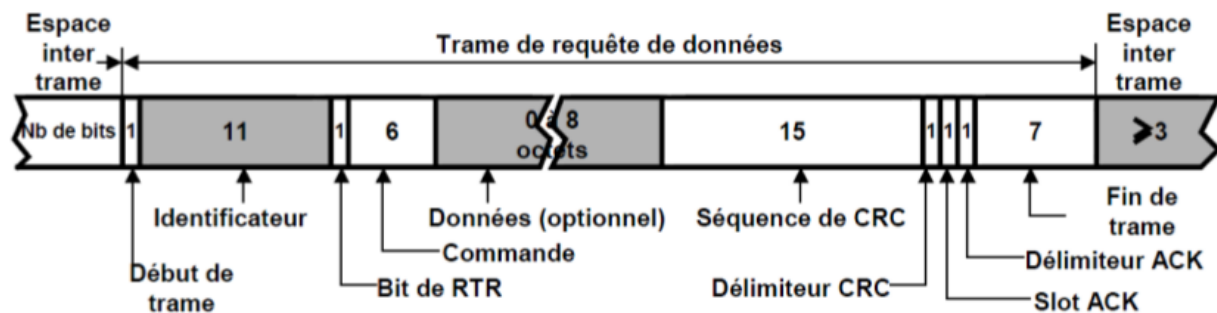
CAN

Le bus CAN est un réseau local. L'objet essentiel d'un réseau est de permettre le transfert de données d'au moins un point source vers au moins un point de destination. Le concept spécifique au bus CAN est la diffusion de l'information de plusieurs source sur un seul et même câble. Et chaque noeud décide si le message le concerne ou pas et si il doit répondre ou pas. On distinguera donc deux trames : une trame de données envoyée par une source et une trame de requête envoyée par le destinataire.

Trame de données :



Trame de sortie :



Légende :

Début de trame : Il indique la transmission d'une trame de données ou d'une trame de requête. Une station est autorisée à émettre lorsque le bus est libre

Champ d'arbitrage : 11 bits de ID-0 à ID-10

- Identificateur : Identifie le contrôleur sur le bus CAN
- RTR : (Remote Transmission Request) Définit la nature de la trame, il est dominant dans une trame de données et récessif dans une trame de requêtes.

Champ de commande : Indique le nombre de d'octets contenu dans le champ de données

Champ de données : Contient jusqu'à 8 octets de données à transmettre

Champ CRC : (Cyclic Redundancy Check) séquence CR

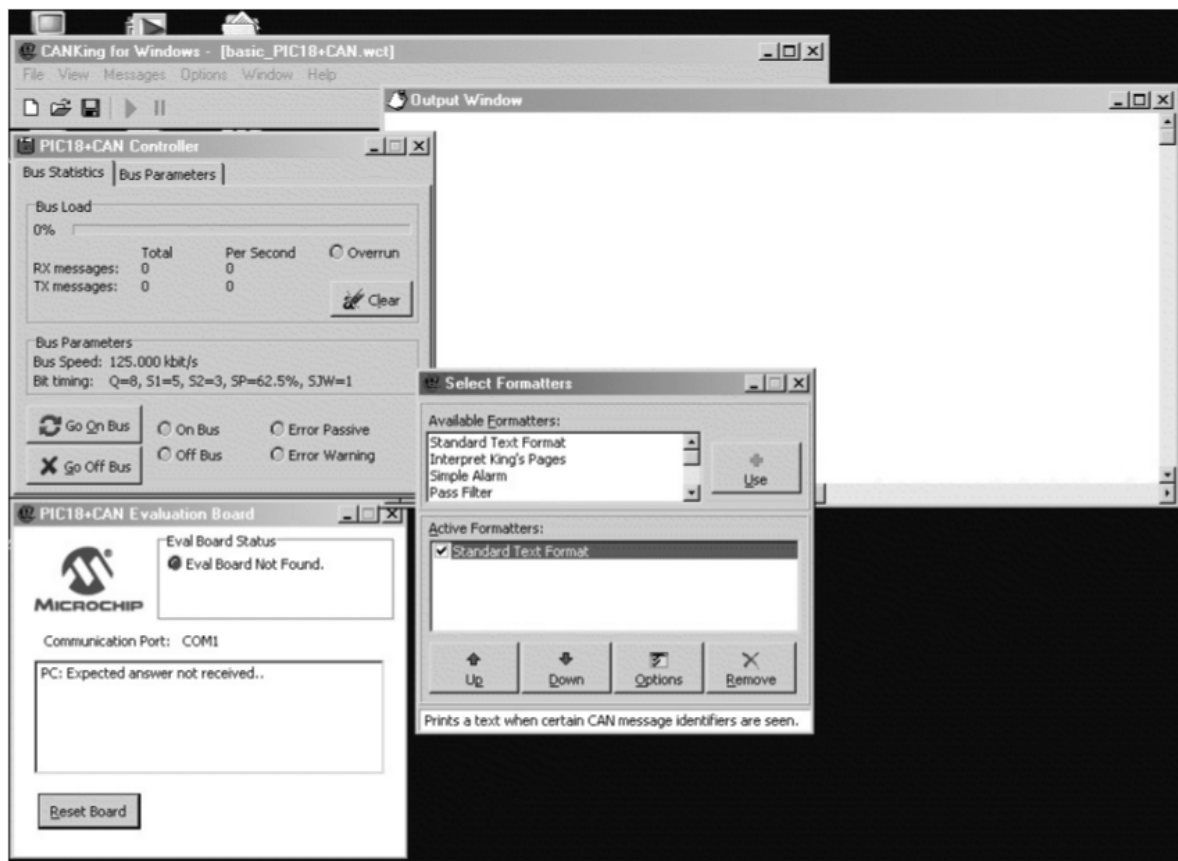
Champ d'acquittement : Composé de 2 bits (SLOT & DELIMITER), transmis à l'état récessif. Toutes les stations ayant validé la séquence CRC envoient un état dominant. Ainsi, le nœud émetteur sait que son message a été reçu par au moins une station.

Fin de trame : C'est une séquence de 7 bit « récessifs » qui ne subit pas la règle de « bit stuffing (bit sans information) ».

Méthodes d'analyses utilisées

Le processus qui va être mis en oeuvre pour analyser les fonctionnements du bus CAN c'est le "CANKing".

En raison de la complexité potentielle de la gestion d'un microcontrôleur au niveau du bit - sans parler d'un protocole réseau tel que CAN - le problème de l'organisation de toutes les informations et les contrôles que l'utilisateur pourrait avoir besoin pour évaluer et gérer devient un défi majeur. "CANKing" simplifie les choses en regroupant tous les contrôles nécessaires pour un type particulier de tâche dans un modèle.



Les modèles individuels et leurs contrôles sont couverts en détail. Un bref aperçu de ces modèles :

- PIC® Register View Template

Le modèle PIC de registre permet de faible niveau de contrôle de "Node0". Il est généralement utilisé pour évaluer et tester le dispositif "Node0" au registre. Tous les registres requis pour le mode 0 du module ECAN sont disponibles dans ce modèle.

- Basic Template

Le modèle de base est un outil de haut niveau qui se concentre sur la circulation des bus CAN. Il est généralement "utilisé pour observer le dispositif PIC18F6X8X/8X8X sur le bus. Le

développement d'un nœud CAN basée sur "Node1" peut être évaluée à l'aide de ce modèle avec "Node0".

- PIC18+CAN Evaluation Board Template

Le modèle de carte d'évaluation fournit en temps réel le contrôle de l'ensemble du conseil comme un outil de démonstration. Il permet aux utilisateurs de contrôler les "LED" pour chacun des nœuds et de surveiller en temps réel l'état des "user push buttons" et les potentiomètres.



Conception - Réalisation - Algorithmes

Fonctions de gestions d'interruptions

Pour gérer les interruptions sur les PICS nous devons réaliser des fonctions permettant les tâches à exécuter face à une interruption donnée.

Par exemple si l'utilisateur appuie sur interrupteur, il faut que le pic sache qu'il doit envoyer une notification à l'utilisateur.

Voici une brève description d'un traitant d'exception.

Procédure traitant_interruption :

DEBUT : =

*/*On bloque toutes les autres interruptions*/*

bloquer_interruptions=1;

si (interruption==interruption_1)

alors

*/*On place les instructions à faire pour l' interruption_1*/*

finsi

.....

si (interruption==interruption_k)

alors

*/*On place les instructions à faire pour l' interruption_k*/*

finsi

*/*On met à Zéro l'interruption car elle à été traitée*/*

interruption=0;

*/*On débloquent toutes les interruptions*/*

bloquer_interruptions=0;

FIN : =

Du fait de la gestion des interruptions réalisées par les PIC18F, il y aura deux procédures traitant_interruption, l'une pour les interruptions dites prioritaires et une autre pour les non prioritaires(cf Annexe interruption)

La sonde de température(Le potentiomètre)

Pour recueillir la température de la sonde, on devra d'abord activer le canal analogique pour récupérer la valeur du potentiomètre puis activer les bons pin pour lire cette valeur.

Jeu de tests envisagé

RS-232

Test de la connexion

Création d'une commande dite de test.

Cette commande consiste à envoyer un signal particulier au PIC#1 et en retour le PIC devra renvoyer une valeur ou message particulier à déterminer.

Allumer une diode et éteindre celle-ci

Création d'un chenillard permettant d'allumer et d'éteindre certaine lumière alternativement.

Les interruptions

On teste les interruptions en utilisant le fonctionnement des interruptions sur les PIC 18F6680/8860(cf datasheet).

Donc si une interruption liée à l'appui du bouton s'est produite alors la diode (N°2) s'allume.

CAN

On visualisera les registres dédiée aux fonctions CAN grâce au Logiciel CANKing, qui permet aussi les données envoyées et traitées par le PIC.

Annexes (Datasheet simplifiés)

Gestion des Interruptions par le microcontrôleur

13 registres utilisés pour la gestion des interruptions

- RCON
- INTCON
- INTCON2
- INTCON3
- PIR1, PIR2, PIR3
- PIE1, PIE2, PIE3
- IPR1, IPR2, IPR3

Chaque interruption possède trois bits qui configure sa configuration:

1. Un bit pour indiquer qu'une interruption a eu lieu (Drapeau)
2. Enable bit qui autorise le programme à se brancher sur le traitant de l'exception
3. Un bit de priorité pour sélectionner la priorité de l'interruption

Sur ce microcontrôleur, le traitant est en mémoire à deux adresses :

- 000008h pour une interruption prioritaire
- 000018h sinon

La fonction de priorité est activable via la mise à un du 7e bits du registre RCON

Le registre INTCON permet de masquer les interruptions grâce à l'activation des bits :

- 7 pour les prioritaires
- 6 pour les autres

L'adresse de retour est sauvegardée sur la pile et le Compteur du Programme(CO) prend la valeur 0x000008h ou 0x000018h en fonction de la priorité.

Les registres de configuration d'interruption : INTCON , INTCON2 et INTCON3

Ces différents registres gèrent toute la configuration des interruptions ainsi que le comportement de celui-ci lorsqu'il doit en traiter une.

Les registres d'Interruption de périphérique : PIR1, PIR2 et PIR3 ainsi que leurs registres de configuration PIE1, PIE2 et PIE3.

Stand-Alone CAN Controller With SPI Interface

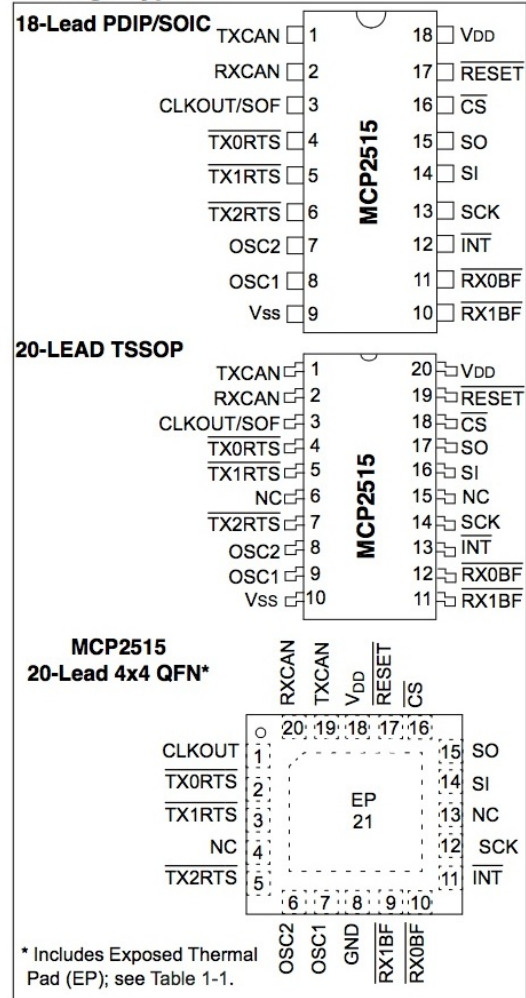
Features

- Implements CAN V2.0B at 1 Mb/s:
 - 0 – 8 byte length in the data field
 - Standard and extended data and remote frames
- Receive buffers, masks and filters:
 - Two receive buffers with prioritized message storage
 - Six 29-bit filters
 - Two 29-bit masks
- Data byte filtering on the first two data bytes (applies to standard data frames)
- Three transmit buffers with prioritization and abort features
- High-speed SPI Interface (10 MHz):
 - SPI modes 0,0 and 1,1
- One-shot mode ensures message transmission is attempted only one time
- Clock out pin with programmable prescaler:
 - Can be used as a clock source for other device(s)
- Start-of-Frame (SOF) signal is available for monitoring the SOF signal:
 - Can be used for time-slot-based protocols and/or bus diagnostics to detect early bus degradation
- Interrupt output pin with selectable enables
- Buffer Full output pins configurable as:
 - Interrupt output for each receive buffer
 - General purpose output
- Request-to-Send (RTS) input pins individually configurable as:
 - Control pins to request transmission for each transmit buffer
 - General purpose inputs
- Low-power CMOS technology:
 - Operates from 2.7V – 5.5V
 - 5 mA active current (typical)
 - 1 μ A standby current (typical) (Sleep mode)
- Temperature ranges supported:
 - Industrial (I): -40°C to +85°C
 - Extended (E): -40°C to +125°C

Description

Microchip Technology's MCP2515 is a stand-alone Controller Area Network (CAN) controller that implements the CAN specification, version 2.0B. It is capable of transmitting and receiving both standard and extended data and remote frames. The MCP2515 has two acceptance masks and six acceptance filters that are used to filter out unwanted messages, thereby reducing the host MCUs overhead. The MCP2515 interfaces with microcontrollers (MCUs) via an industry standard Serial Peripheral Interface (SPI).

Package Types



**MICROCHIP****PIC18F6585/8585/6680/8680**

**64/68/80-Pin High Performance, 64-Kbyte Enhanced FLASH
Microcontrollers with ECAN Module**

High Performance RISC CPU:

- Source code compatible with the PIC16 and PIC17 instruction sets
- Linear program memory addressing to 2 Mbytes
- Linear data memory addressing to 4096 bytes
- 1 Kbyte of data EEPROM
- Up to 10 MIPS operation:
 - DC - 40 MHz osc./clock input
 - 4 MHz - 10 MHz osc./clock input with PLL active
- 16-bit wide instructions, 8-bit wide data path
- Priority levels for interrupts
- 31-level, software accessible hardware stack
- 8 x 8 Single Cycle Hardware Multiplier

**External Memory Interface
(PIC18F8X8X Devices Only):**

- Address capability of up to 2 Mbytes
- 16-bit interface

Peripheral Features:

- High current sink/source 25 mA/25 mA
- Four external interrupt pins
- Timer0 module: 8-bit/16-bit timer/counter
- Timer1 module: 16-bit timer/counter
- Timer2 module: 8-bit timer/counter
- Timer3 module: 16-bit timer/counter
- Secondary oscillator clock option - Timer1/Timer3
- One Capture/Compare/PWM (CCP) module:
 - Capture is 16-bit, max. resolution 6.25 ns ($T_{CY}/16$)
 - Compare is 16-bit, max. resolution 100 ns (T_{CY})
 - PWM output: PWM resolution is 1- to 10-bit
- Enhanced Capture/Compare/PWM (ECCP) module:
 - Same Capture/Compare features as CCP
 - One, two, or four PWM outputs
 - Selectable polarity
 - Programmable dead-time
 - Auto-shutdown on external event
 - Auto-restart
- Master Synchronous Serial Port (MSSP) module with two modes of operation:
 - 3-wire SPI™ (supports all 4 SPI modes)
 - I²C™ Master and Slave mode
- Enhanced Addressable USART module:
 - Supports RS-232, RS-485 and LIN 1.2
 - Programmable wake-up on START bit
 - Auto baud detect
- Parallel Slave Port (PSP) module

Analog Features:

- Up to 16-channel, 10-bit Analog-to-Digital Converter module (A/D) with:
 - Fast Sampling Rate
 - Programmable Acquisition Time
 - Conversion available during SLEEP
- Programmable 16-level Low Voltage Detection (LVD) module:
 - Supports interrupt on Low Voltage Detection
- Programmable Brown-out Reset (BOR)
- Dual analog comparators:
 - Programmable input/output configuration

ECAN Module Features:

- Message bit rates up to 1 Mbps
- Conforms to CAN 2.0B ACTIVE Specification
- Fully backward compatible with PIC18XXX8 CAN modules
- Three modes of operation:
 - Legacy, Enhanced Legacy, FIFO
- Three dedicated transmit buffers with prioritization
- Two dedicated receive buffers
- Six programmable receive/transmit buffers
- Three full 29-bit acceptance masks
- 16 full 29-bit acceptance filters with dynamic association
- DeviceNet™ data byte filter support
- Automatic remote frame handling
- Advanced Error Management features

Special Microcontroller Features:

- 100,000 erase/write cycle Enhanced FLASH program memory typical
- 1,000,000 erase/write cycle Data EEPROM memory typical
- 1-second programming time
- FLASH/Data EEPROM Retention: > 40 years
- Self-reprogrammable under software control
- Power-on Reset (POR), Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own On-Chip RC Oscillator
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options including:
 - Software enabled 4X Phase Lock Loop (of primary oscillator)
 - Secondary Oscillator (32 kHz) clock input
- In-Circuit Serial Programming™ (ICSP™) via two pins
- MPLAB® In-Circuit Debug (ICD) via two pins