



Cégep Limoilou

ÉLECTRONIQUE PROGRAMMABLE ET ROBOTIQUE

247-6 [1-2-3-4] 7-LI

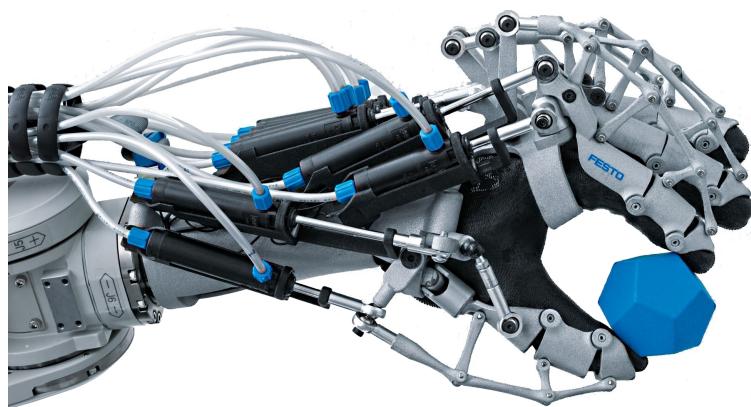
Projet de 5^e session

Étudiants :

Vincent Chouinard
Hicham Safoine
Gabriel Fortin-Bélanger
Louis-Nomand Ang-Houle

Professeurs :

Ali Tadli
Alain Champagne
Stéphane Deschênes
Étienne Tremblay



L'usine à gaz, et le gaz, c'est de l'air !

10 décembre 2014

Table des matières

1 Présentation du projet	5
1.1 Explication du projet	5
1.2 Schéma bloc d'ensemble du système	5
1.2.4 Schéma bloc de la station de pesée	6
1.2.1 Schéma bloc du bolide	7
1.2.2 Schéma bloc de la table FESTO	8
1.2.3 Schéma bloc du SOC8200	9
1.2.5 de la station no.1	10
1.3 Liste des logiciels	10
1.4 Liste des trames	11
2 Le matériel	12
2.1 Bloc 1	12
2.2 Bloc 2	12
2.3 Bloc 3	12
2.4 Bloc 4	12
2.5 Explication des types de liens	12
2.5.1 RS232	12
2.5.2 Xbee	12
2.6 Explication des trames	13
2.6.1 RS-232	13
2.6.2 CAN	14
2.6.3 XBEE	16
2.7 Liste des pièces	16
2.7.1 Liens web	17
2.7.2 Datasheets	18
3 Interface PC	20
3.1 Gestion de l'historique	22
3.1.1 Exemple d'historique typique	22
3.2 Structure du programme	22
3.2.1 Les Ghosts Labels	22
3.3 Explication des trames	22
3.4 Ordre de gestion des tâches	22
4 Logiciel du SOC8200	23
4.1 Description du programme	23
4.2 Schéma bloc du script shell	23
4.3 Gestion des processus et du temps de CPU	23
4.4 Format et récupération des logs	23
4.5 Liste des tests et logiciels	24
5 Logiciel de la station 1 et du bolide	25
5.1 La station no.1	25
5.2 La station no.2	25
5.4 Le bolide	26
5.3 Schéma des héritages de classes	27
5.5 Procédure de compilation sur IAR	28
5.6 Procédure de vérification	28

6 Logiciel du module PIC18F258	29
6.1 Description du fonctionnement du programme	29
6.2 Procédure de compilation sur MPLAB	29
6.3 Procédure de vérification	29
7 Schémas OrCAD	30
8 Fichiers Gerbers	34
9 Calculs	37
9.1 Calcul du pas de conversion de la pile	37
9.2 Calcul du baudrate	37
10 Conclusions	38
10.1 Ce que le projet m'a apporté	38
10.1.1 Vincent	38
10.1.2 Hicham	38
10.1.3 Gabriel	38
10.1.4 Louis-Norman	38
10.2 Difficultés et corrections	38
10.2.1 Vincent	38
10.2.2 Hicham	38
10.2.3 Gabriel	38
10.2.4 Louis-Norman	38
10.3 Ce que j'ai aimé ou pas	38
10.3.1 Vincent	38
10.3.2 Hicham	38
10.3.3 Gabriel	38
10.3.4 Louis-Norman	38
11 ANNEXE 1 : Code source du programme pour PC	39
12 ANNEXE 2 : Code source du Bolide et de la station 1	40
13 ANNEXE 4 : Code source de la table FESTO	41
14 ANNEXE 5 : Code source du programme PIC	42
15 ANNEXE 4 : Script Shell du SOC8200	43

Table des figures

1	Vue d'ensemble du projet	5
2	Schéma bloc du Bolide	7
3	Schéma bloc de la table FESTO	8
4	Schéma bloc du SOC8200	9
5	Programme de contrôle principal	20
6	Options CAN avancées	21
7	Historique des actions	22
8	Héritage de la classe de contrôle du véhicule	27
9	Héritage de la classe de contrôle de la station no.1	27
10	Qui hérite du SPI ?	27
11	Qui hérite de l'I2C ?	27
12	Choix de la cible sur IAR	28
13	Compiler avec MicroC PRO pour PIC	29
14	Schémas carte IO I2C, page 1	30
15	Schémas carte IO I2C, page 2	31
16	Schémas carte IO I2C, page 3	32
17	Schémas carte IO I2C, page 4	33
18	Couche TOP	34
19	Couche BOT	34
20	Silk Screen TOP	35
21	Solder mask TOP	35
22	Drill	36

Liste des tableaux

1	Index des identifiants matériel CAN	14
2	Index des trames CAN	14
3	Index des communications CAN	14
4	Informations sur le bus I2C du bolide	26

1 Présentation du projet

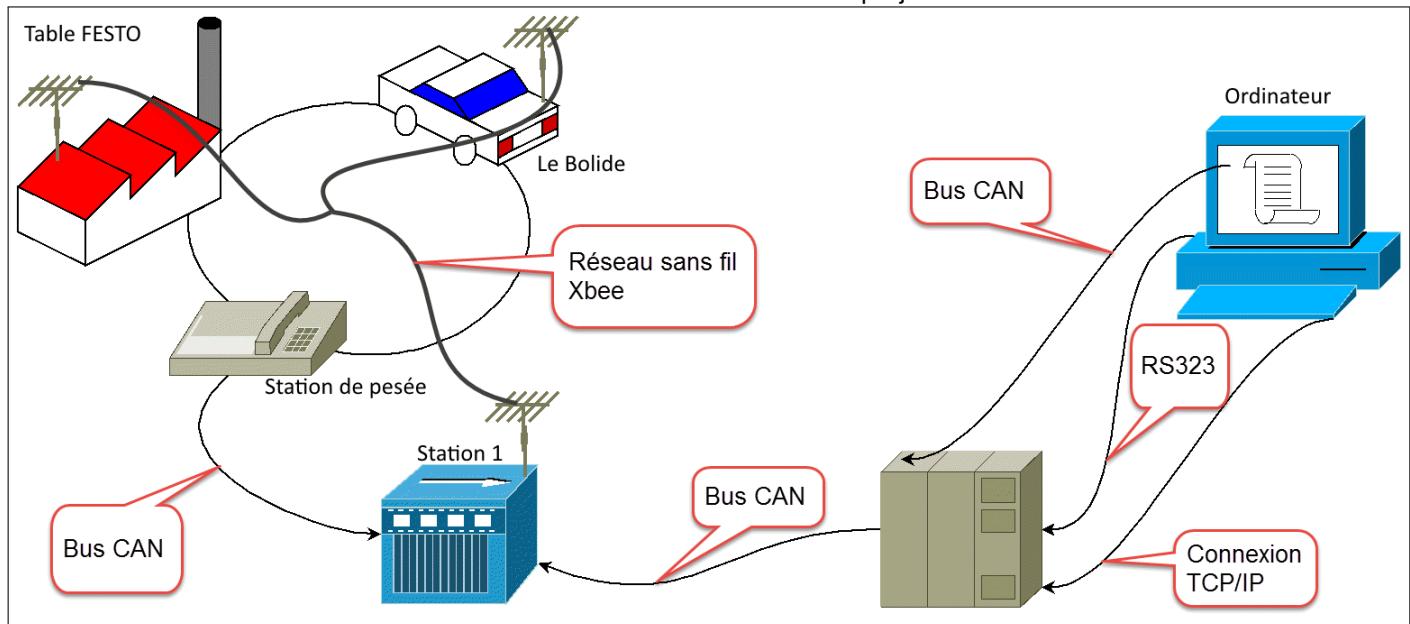
Le projet de la cinquième session consiste à réaliser (mettre un cours extrait des consignes)

- ⇒ Le Bolide
- ⇒ Carte Dallas DS89C450
- ⇒ Carte uPSD 3254A
- ⇒ SOC8200
- ⇒ Table FESTO
- ⇒ Carte PIC machin-chose-binouche
- ⇒ Carte d'extension I₂C
- ⇒ Carte d'extension SPI
- ⇒ Une pile de 10.8 volts
- ⇒ Quatre moteurs et autant de pneus

1.1 Explication du projet

1.2 Schéma bloc d'ensemble du système

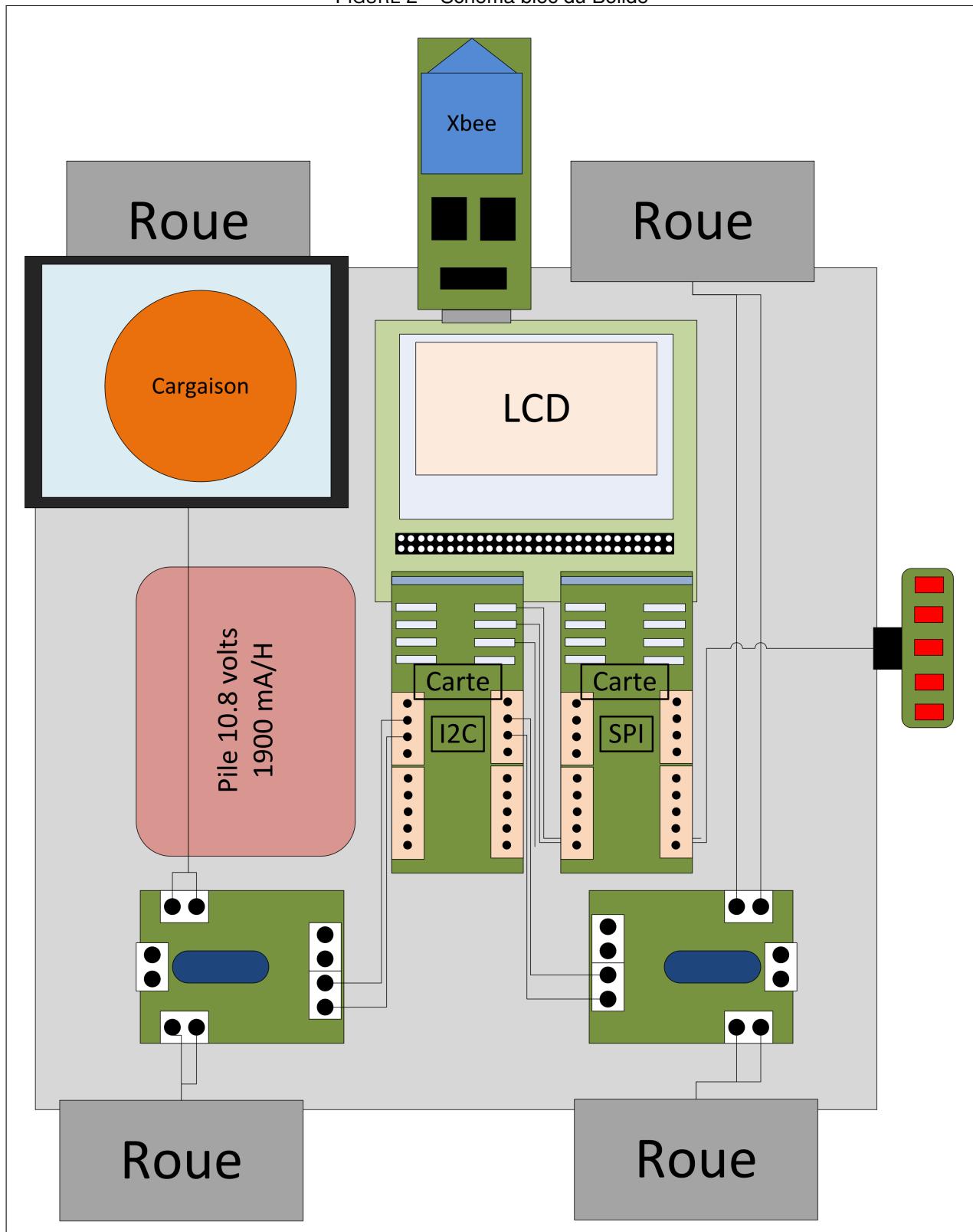
FIGURE 1 – Vue d'ensemble du projet



1.2.4 Schéma bloc de la station de pesée

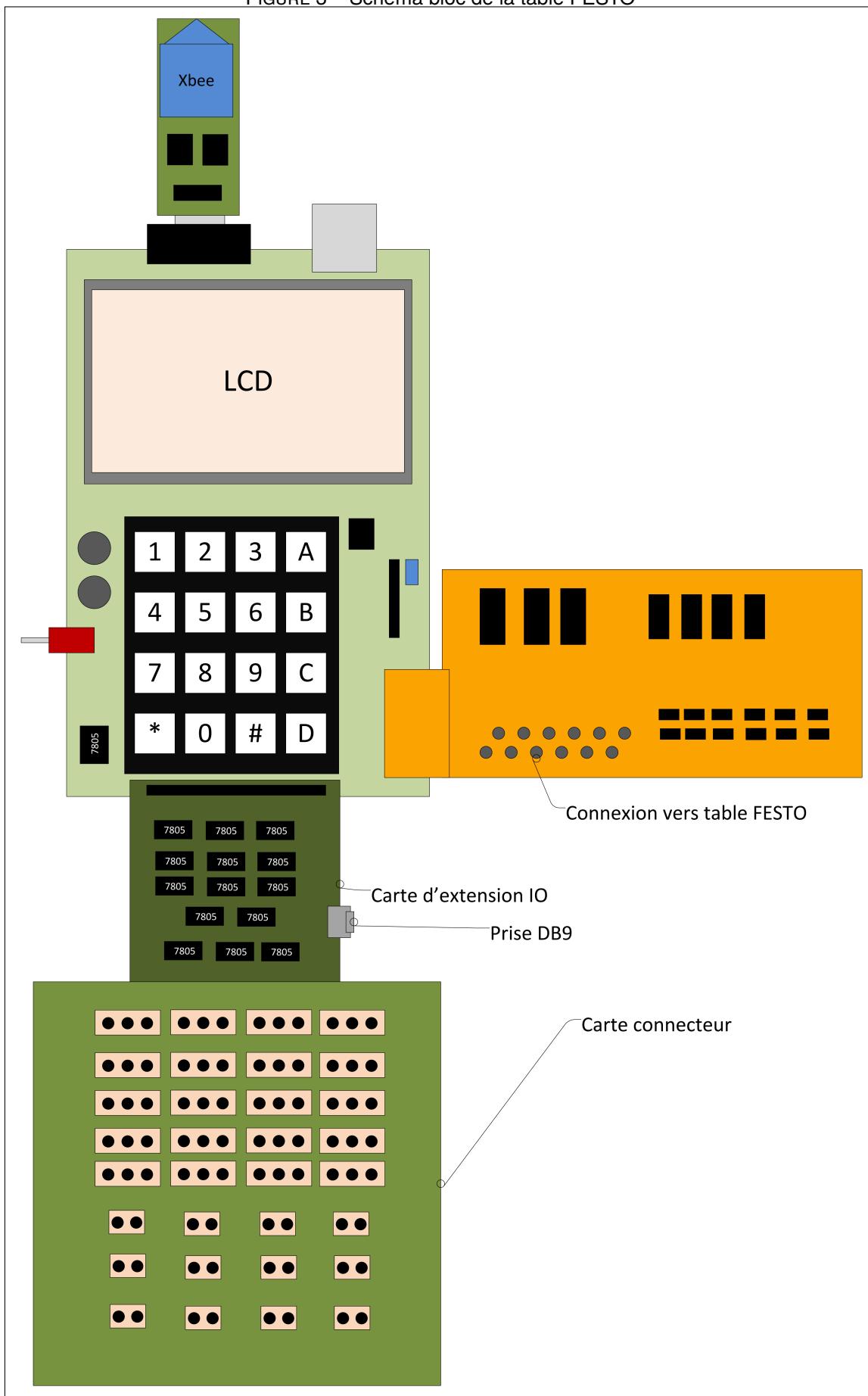
1.2.1 Schéma bloc du bolide

FIGURE 2 – Schéma bloc du Bolide



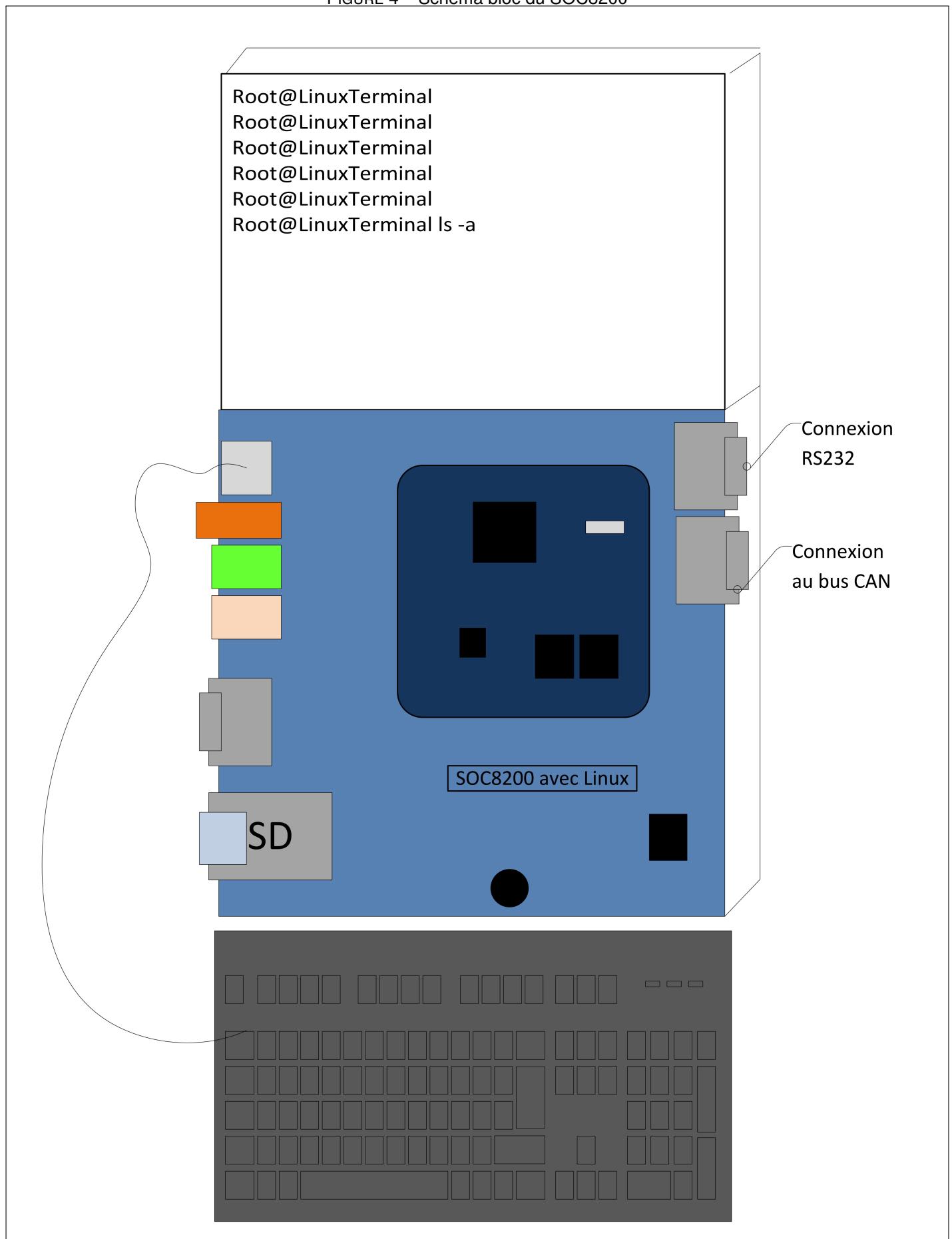
1.2.2 Schéma bloc de la table FESTO

FIGURE 3 – Schéma bloc de la table FESTO



1.2.3 Schéma bloc du SOC8200

FIGURE 4 – Schéma bloc du SOC8200



1.2.5 de la station no.1

1.3 Liste des logiciels

Terminaux

- UART Master 0.97
- Serializ3r 1.0.2
- TerraTerm
- Putty
- GTKterm 0.99.7-rc1
- Terminator
- CAPS
- tinyBootloader

Gestionnaires de projet

- MS Project 2010
- Git Hub

Compilateurs et IDE

- Visual Studio 2013

- Visual Studio 2010

- IAR 8.20

- MPLAB X version 6.00

Éditeurs de texte

- Notepad++
- BowPad
- medit 1.2.0

Schémas électriques

- OrCAD 16.2

Systèmes d'exploitation

- Windows 7 SP1

- Windows 8.1

- Windows XP SP3

- CentOS

- Arch Linux

Autres

- VMWare Workstation 10
- TeXmaker 4.3
- Dukto R6
- Dia
- Microsoft Visio 2013
- Festo configuration tool
- L^AT_EX(avec plug-in Doxygen)

1.4 Liste des trames

f

2 Le matériel

2.1 Bloc 1

D'un point de vue matériel, le bloc 1 est le plus simple, car il est composé d'un ordinateur Windows muni d'une carte PCI vers CAN et d'une prise RS232 et d'une prise Ethernet. Il n'y a rien à faire, mise à part brancher les bons câbles aux bons endroits. Son rôle est de contrôler et diriger toute l'opération et de veiller au bon fonctionnement de chaque composante à l'aide d'une application en Csharp. Le bloc 1 est le cerveau de l'usine.

2.2 Bloc 2

Le bloc 2 est composé du bolide et de la station no.1. Cette dernière, dont le cerveau est une carte uPSD, joue le rôle de centralisateur CAN. En effet, cette station reçoit des consignes¹ en provenance du PC, consignes qu'elle s'empresse d'expédier aux bons endroits via Xbee. De plus, cette station reçoit des informations de la station de pesée², de la table FESTO³ et du bolide⁴. Ces informations sont systématiquement retransmises au PC via le bus CAN.

2.3 Bloc 3

Le bloc 3 est composé d'une carte PIC (mettre modèle) et d'une balance (mettre modèle.) Comme son nom l'indique, la station de pesée pèse le bloc et envoie l'information (le poids) au PC et au SOC8200 via le bus CAN. La carte PIC fait office de convertisseur CAN vers RS232.

2.4 Bloc 4

Le bloc 4 est composé d'un système embarqué Linux basé sur le SOC8200. Son rôle principal est d'agir comme sniffeur d'information et d'afficher sur son écran toutes les données qui transitent sur le bus CAN. Toutefois, ce dernier est en mesure de détecter une défaillance du PC via un gestionnaire de HeartBeat et de prendre la relève en tant que cerveau de l'opération. Le SOC8200 agit comme vice-président du bus CAN.

2.5 Explication des types de liens

2.5.1 RS232

Un lien RS232 9600 Bauds est établi entre l'ordinateur et le SOC8200. Ce lien sert à l'envoi et à la réception de HeartBeat, afin que le SOC8200 ou l'ordinateur soit informé de toute défaillance de l'autre.

2.5.2 Xbee

Lorsque les modules Xbee sont adéquatement configurés, ils font office de remplacement au câble RS232. En effet, nos Xbee discutent entre eux à l'aide du protocole de communication RS232 à 9600 bauds.

-
1. Sous forme de trames
 2. sous forme de trames CAN
 3. Via Xbee
 4. Idem

2.6 Explication des trames

2.6.1 RS-232

Le protocole RS-232 sert à envoyer et à recevoir des HeartBeat. Le PC et le SOC 8200 s'envoient tous deux un HeartBeat par seconde à 9600 bauds. Un HeartBeat, c'est simplement le mot "Allo". Le PC et le SOC2800 "écoutent" les HeartBeats, et si ces derniers ne sont pas entendus, chaque dispositif tient pour acquis que l'autre est hors service et prend la relève de la gestion du bus CAN.

2.6.2 CAN

Chaque composante matérielle, du Bolide au PC, dispose d'un identifiant CAN unique allant de 000 à 005. Chaque fonctionnalité dispose d'un code d'identification suivi de deux octets de données à transmettre.

TABLE 1 – Index des identifiants matériel CAN

Device	ID matériel
Ordinateur	000
SOC8200	001
Station 1	002
Station 2	003
Véhicule	004
Station de pesés	005

TABLE 2 – Index des trames CAN

Fonctionnalité	Identifiant	Données
Démarre le véhicule	0x00	0x00
Arrête le véhicule	0x00	0x01
Le véhicule est arrêté	0x01	0x00
Le véhicule est en marche	0x01	0x01
Le véhicule est hors circuit	0x01	0x02
Vitesse (0-100)	0x02	0x00 à 0xFF
Batterie	0x03	0x00 à 0xFF
Couleur du bloc	0x04	0x00 à 0x02
Poids du bloc	0x05	0x00 à 0xFF
Envoyer l'heure	0x06	à déterminer
No. de la station	0x07	0x00 à 0x02
Demande de l'historique	0xC0	0x00
Direction horaire et antihoraire	0x08	0x00 à 0x01

TABLE 3 – Index des communications CAN

Émetteur	Action	ID receveur	Donnée envoyée	Récepteur	Erreur
Ordinateur	Démarrer le véhicule	004	00 00	Véhicule	F1
Ordinateur	Arrêter le véhicule	004	00 01	Véhicule	F2
Véhicule	Dit : je suis arrêté	000	01 00	Ordinateur	F3
Véhicule	Dit : j'avance	000	01 01	Ordinateur	F4
Véhicule	Dit : je suis hors circuit	000	01 02	Ordinateur	F5
Véhicule	Dit sa vitesse	000	02 [00 à 64]	Ordinateur	F6
Véhicule	Dit le niveau de sa batterie	000	03 [00 à 64]	Ordinateur	F7
Station 1	Dit bloc = métal	000	04 00	Ordinateur	F8
Station 1	Dit bloc = orange	000	04 01	Ordinateur	F9
Station 1	Dit bloc = noir	000	04 02	Ordinateur	FA
Station 1	Dit le poids du bloc	000	05 [00 à 64]	Ordinateur	FB
Voiture	Dit qu'elle est à la station 1	000	07 00	Ordinateur	FC
Voiture	Dit qu'elle est à la station 2	000	07 01	Ordinateur	FD
Ordinateur	Envoie l'heure	003	06 à déterminer	Station 1	FE
Ordinateur	Demande le LOG	001	C0 00	SOC8200	E0
Ordinateur	Exige Horaire	004	08 00	Véhicule	E1
Ordinateur	Exige Antihoraire	004	08 01	Véhicule	E2

Exemples de trames CAN à transmettre au PC.

```
CAN.SendToPC("0100FF"); // Arrêté
CAN.SendToPC("0101FF"); // En marche
CAN.SendToPC("0102FF"); // Hors circuit
CAN.SendToPC("02xxFF"); // Vitesse de xx
CAN.SendToPC("03xxFF"); // Batterie chargée à xx %
CAN.SendToPC("0400FF"); // Bloc métallique
CAN.SendToPC("0401FF"); // Bloc noire
CAN.SendToPC("0402FF"); // Bloc orange
CAN.SendToPC("050064"); // Le bloc est lourd
CAN.SendToPC("0700FF"); // Rendu à la station 1
CAN.SendToPC("0701FF"); // Rendu à la station 2
CAN.SendToPC("0702FF"); // Rendu à la station 3
```

2.6.3 XBEE

Trois modules Xbee sont présents sur l'ensemble du projet, soit sur la station no.1 (la carte uPSD), sur la station no.2 (la table FESTO) et la station no.4, c'est-à-dire le bolide. La particularité des Xbee est que lorsqu'ils sont adéquatement configurés, tout ce qu'envoie un Xbee est reçu et lu par tous les autres Xbee à proximité, et c'est pourquoi nous avons défini un système de trames.

Note : mettre image d'un Xbee

2.7 Liste des pièces

- Carte Dallas
 - Carte uPSD
 - SOC 8200 (avec clavier et écran)
 - PIC18FmachinTruc
 - Carte d'extension SPI
 - Carte d'extension I2C
 - Carte CAN MCP2515
 - Xbee
 - Table FESTO
 - Carte connecteur IO 24 volts
 - Carte d'extension DAC ADC
 - Carte Xbee vers DB9
 - Câble Ethernet croisé
 - Câble Ethernet régulier
 - Câble DB9
 - Le Bolide
 - SaberTooth motor drive 2x5
 -
-

2.7.1 Liens web

Mettre ien vers GitHUB

DS89C450

<http://datasheets.maximintegrated.com/en/ds/DS89C430-DS89C450.pdf>

uPSD3254

http://www.datasheetcatalog.com/datasheets_pdf/U/P/S/D/UPSD3254.shtml

DAC65574

http://www.datasheetcatalog.com/datasheets_pdf/D/A/C/6/DAC6574.shtml

LM3914

http://www.datasheetcatalog.com/datasheets_pdf/L/M/3/9/LM3914.shtml

PCF8574

http://www.datasheetcatalog.net/datasheets_pdf/P/C/F/8/PCF8574.shtml

OPT101

http://www.datasheetcatalog.com/datasheets_pdf/O/P/T/1/OPT101.shtml

SeberTooth 2x5

2.7.2 Datasheets

Note : Toutes les datasheets sont en format non-compressé. Vous pouvez zoomer sur le document PDF⁵ afin de lire l'intégralité de leurs premières pages.

74HC14

PbA6 Semiconductor		Product Specification																									
Hex Inverting Schmitt trigger		74HCT14																									
FEATURES																											
<ul style="list-style-type: none"> • Output capable of standard • 10V output logic 																											
GENERAL DESCRIPTION																											
<p>The 74HCT14 is a high speed Schmitt CMOS device and are pin compatible with the power Schottky TTL, ECL/TTL, and CMOS logic families.</p> <p>The 74HCT14 provides six inverting buffers with Schmitt trigger action. They are capable of transforming logic changing input signals into sharply defined, square output signals.</p>																											
LOGIC SYMBOL DATA																											
SYMBOL PARAMETER CONDITIONS TYPICAL UNIT																											
<table border="1"> <tr> <td colspan="2">Max. Input</td><td colspan="2">Input voltage, V_{IN}, V_{DD} = 5 V</td><td colspan="2">0.8</td></tr> <tr> <td colspan="2">Max. Input</td><td colspan="2">Input voltage, V_{IN}, V_{DD} = 12 V</td><td colspan="2">1.8</td></tr> <tr> <td colspan="2">Max. Input</td><td colspan="2">Input voltage, V_{IN}, V_{DD} = 15 V</td><td colspan="2">2.2</td></tr> <tr> <td colspan="2">Max. Input</td><td colspan="2" rowspan="4">Input voltage, V_{IN}, V_{DD} = 18 V</td><td colspan="2">2.6</td></tr> </table>				Max. Input		Input voltage, V _{IN} , V _{DD} = 5 V		0.8		Max. Input		Input voltage, V _{IN} , V _{DD} = 12 V		1.8		Max. Input		Input voltage, V _{IN} , V _{DD} = 15 V		2.2		Max. Input		Input voltage, V _{IN} , V _{DD} = 18 V		2.6	
Max. Input		Input voltage, V _{IN} , V _{DD} = 5 V		0.8																							
Max. Input		Input voltage, V _{IN} , V _{DD} = 12 V		1.8																							
Max. Input		Input voltage, V _{IN} , V _{DD} = 15 V		2.2																							
Max. Input		Input voltage, V _{IN} , V _{DD} = 18 V		2.6																							
Notes <ul style="list-style-type: none"> 1. C_{in} is used to determine the dynamic power dissipation ($P_{d} = V_{DD}^2 \cdot C_{in}$) 2. $V_{DD} = 5 \text{ V}$ to 18 V, $V_{SS} = 0 \text{ V}$, $T = 25^\circ\text{C}$, $f = 1 \text{ MHz}$ 3. Input frequency is limited by $f_{IN} = 1/(2\pi R_{IN}C_{IN})$ 4. Input load expression is $I_{IN} = I_{IN}(V_{DD}, V_{SS}, V_{IN})$ 5. Input load expression is $I_{IN} = I_{IN}(V_{DD}, V_{SS}, V_{IN})$ 6. $V_{DD} = 12 \text{ V}$ to 18 V, $V_{SS} = 0 \text{ V}$, $T = 25^\circ\text{C}$, $f = 1 \text{ MHz}$ 7. For HCT conditions $V_{DD} = 12 \text{ V}$ to 18 V, $V_{SS} = 0 \text{ V}$, $T = 25^\circ\text{C}$, $f = 1 \text{ MHz}$ 8. For HCT conditions $V_{DD} = 12 \text{ V}$ to 18 V, $V_{SS} = 0 \text{ V}$, $T = 25^\circ\text{C}$, $f = 1 \text{ MHz}$ 																											
ORDERING INFORMATION																											
See "74HCT14-74HCT140 Logic Package Information".																											

LineSensors

Registers:

You can read the I/O line status by issuing an I/O read of the programmed address. A single byte representing the status of the sensors will be returned, ranging from 0 to 31, where 0 means "None" and 31 means "All On".

	Address	Description	Address #	Address #	Address #	Decimal Meaning
0	0	None	0	0	0	0
1	0	0	0	0	0	1
0	1	0	0	0	0	2
0	2	0	0	0	0	3
0	3	0	0	0	0	4
1	0	1	0	0	0	5
0	1	1	0	0	0	6
1	1	1	0	0	0	7
0	2	0	1	0	0	8
1	0	0	1	0	0	9
0	1	0	1	0	0	10
1	1	0	1	0	0	11
0	0	1	1	0	0	12
1	0	1	1	0	0	13
0	1	1	1	0	0	14
1	1	1	1	0	0	15
0	0	0	0	1	0	16
1	0	0	0	1	0	17
0	1	0	0	1	0	18

PCF8573

	REMOTE 4-BIT I/O EXPANDER FOR PC-BUS PCF8574																																				
<p>Product Summary</p> <p>The PCF8574 is a remote 4-bit bidirectional I/O expander designed for the PC-Bus interface. It features a built-in 4-bit bidirectional parallel port (PPI) which includes high-current drivers and receivers. The PCF8574 is a high-speed device with a maximum propagation delay of 10 ns. It also has a low power consumption of 1.5 mA at 5 V. The device is housed in a compact 16-pin DIP package.</p> <p>Key Features</p> <ul style="list-style-type: none"> • Low Standby Current Consumption of 1.5 mA at Maximum • PC-Bus Compatible • Open-Drain Input/Output • Compatible With Most Microcontrollers • Latched Outputs With High-Speed Drive Capability for Directly Driving LEDs • Logic Input Current Exceeds 100-nA Per JEDEC Specification Class II 																																					
<p>Pinouts</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%;">DW OR IN PACKAGE</td> <td style="width: 33%;">PIN NUMBER</td> <td style="width: 33%;">PIN PACKAGE</td> </tr> <tr> <td></td> <td>A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14 A15 A16</td> <td></td> </tr> <tr> <td></td> <td>B1 B2 B3 B4 B5 B6 B7 B8 B9 B10 B11 B12 B13 B14 B15 B16</td> <td></td> </tr> <tr> <td></td> <td>1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16</td> <td></td> </tr> </table> <p>DCS OR IN PACKAGE</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%;">DCS OR IN PACKAGE</td> <td style="width: 33%;">PIN NUMBER</td> <td style="width: 33%;">PIN PACKAGE</td> </tr> <tr> <td></td> <td>SCL SDA NC NC</td> <td></td> </tr> <tr> <td></td> <td>1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16</td> <td></td> </tr> <tr> <td></td> <td>1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16</td> <td></td> </tr> </table> <p>DS9 OR IN PACKAGE</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%;">DS9 OR IN PACKAGE</td> <td style="width: 33%;">PIN NUMBER</td> <td style="width: 33%;">PIN PACKAGE</td> </tr> <tr> <td></td> <td>SCL SDA NC NC</td> <td></td> </tr> <tr> <td></td> <td>1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16</td> <td></td> </tr> <tr> <td></td> <td>1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16</td> <td></td> </tr> </table>		DW OR IN PACKAGE	PIN NUMBER	PIN PACKAGE		A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14 A15 A16			B1 B2 B3 B4 B5 B6 B7 B8 B9 B10 B11 B12 B13 B14 B15 B16			1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16		DCS OR IN PACKAGE	PIN NUMBER	PIN PACKAGE		SCL SDA NC			1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16			1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16		DS9 OR IN PACKAGE	PIN NUMBER	PIN PACKAGE		SCL SDA NC			1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16			1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16	
DW OR IN PACKAGE	PIN NUMBER	PIN PACKAGE																																			
	A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14 A15 A16																																				
	B1 B2 B3 B4 B5 B6 B7 B8 B9 B10 B11 B12 B13 B14 B15 B16																																				
	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16																																				
DCS OR IN PACKAGE	PIN NUMBER	PIN PACKAGE																																			
	SCL SDA NC																																				
	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16																																				
	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16																																				
DS9 OR IN PACKAGE	PIN NUMBER	PIN PACKAGE																																			
	SCL SDA NC																																				
	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16																																				
	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16																																				
<p>Description/Ordering Information</p> <p>The DS9 input/output (I/O) response for the four-bit bidirectional port is designed for the PC-Bus interface.</p> <p>The PCF8574 provides pin-compatible remote I/O expansion for most microcontroller families via the PC-Bus interface.</p> <p>The device features a built-in quad-buffered 16-bit port (PPI), including bidirectional outputs with high-current drive capability for direct connection to LEDs. The PPI is a standard bidirectional port with the use of a bidirectional control signal, <i>Port Enable</i>. In the mode, the user can connect source to V_{DD} or ground. The bidirectional port is controlled by the bidirectional control signal, <i>Port Enable</i>. The bidirectional port can be on when a logic 1 is applied to it, whether high or switched by the negative edge of <i>SCl</i>. The I/Os should be high during logic 1.</p>																																					
<p>Ordering Information</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th style="width: 10%;">Part No.</th> <th style="width: 20%;">Package</th> <th style="width: 20%;">Description</th> <th style="width: 10%;">Voltage</th> <th style="width: 10%;">Temperature Range</th> <th style="width: 10%;">Lead-Free</th> </tr> <tr> <td>PCF8574</td> <td>DIP-16</td> <td>PCF8574</td> <td>5V</td> <td>-40°C to +85°C</td> <td>Yes</td> </tr> <tr> <td>PCF8574-100</td> <td>DIP-16</td> <td>PCF8574 with 100-nA logic input current</td> <td>5V</td> <td>-40°C to +85°C</td> <td>Yes</td> </tr> <tr> <td>PCF8574-100-100</td> <td>DIP-16</td> <td>PCF8574 with 100-nA logic input current and 100-nA logic output current</td> <td>5V</td> <td>-40°C to +85°C</td> <td>Yes</td> </tr> <tr> <td>PCF8574-100-100-100</td> <td>DIP-16</td> <td>PCF8574 with 100-nA logic input current, 100-nA logic output current, and 100-nA logic enable current</td> <td>5V</td> <td>-40°C to +85°C</td> <td>Yes</td> </tr> </table> <p>WEEE Directive</p> <p>Household waste equipment. This symbol indicates that a product must not be disposed of as household waste. It must be handed over to a designated collection point for the recycling of electrical and electronic equipment.</p> <p>Caution</p> <p>Do not exceed an input voltage exceeding 30 Vdc, absolute maximum, and an output voltage of 30 Vdc, absolute maximum, for any pins connected to ground.</p> <p>Texas Instruments</p> <p>POST OFFICE BOX 5010 • DALLAS, TEXAS 75222-5010 • (800) 447-5596 • FAX (214) 349-5587</p> <p>Copyright © 1995 Texas Instruments Incorporated</p>		Part No.	Package	Description	Voltage	Temperature Range	Lead-Free	PCF8574	DIP-16	PCF8574	5V	-40°C to +85°C	Yes	PCF8574-100	DIP-16	PCF8574 with 100-nA logic input current	5V	-40°C to +85°C	Yes	PCF8574-100-100	DIP-16	PCF8574 with 100-nA logic input current and 100-nA logic output current	5V	-40°C to +85°C	Yes	PCF8574-100-100-100	DIP-16	PCF8574 with 100-nA logic input current, 100-nA logic output current, and 100-nA logic enable current	5V	-40°C to +85°C	Yes						
Part No.	Package	Description	Voltage	Temperature Range	Lead-Free																																
PCF8574	DIP-16	PCF8574	5V	-40°C to +85°C	Yes																																
PCF8574-100	DIP-16	PCF8574 with 100-nA logic input current	5V	-40°C to +85°C	Yes																																
PCF8574-100-100	DIP-16	PCF8574 with 100-nA logic input current and 100-nA logic output current	5V	-40°C to +85°C	Yes																																
PCF8574-100-100-100	DIP-16	PCF8574 with 100-nA logic input current, 100-nA logic output current, and 100-nA logic enable current	5V	-40°C to +85°C	Yes																																

DAC6574

The figure shows the front page of the DAT4574 Datasheet. It features the Texas Instruments logo at the top left, followed by the part number 'DAT4574' and its package type '16-pin PLCC'. The title 'QUAD, 10-BIT LOW-POWER, VOLTAGE OUTPUT, FC INTERFACE DIGITAL-TO-ANALOG CONVERTER' is centered above the features section. Below the title, there are two columns: 'FEATURES' and 'DESCRIPTION'. The 'FEATURES' column lists 14 key product characteristics, while the 'DESCRIPTION' column provides a detailed technical overview. At the bottom of the page, there is a large circuit diagram showing the internal architecture of the DAT4574, including four DAC cores, a central control logic, and various interface and power supply components.

LM3914

Texas
Instruments

LM3914

LM3914 DotBar Display Driver

Check for Update

FEATURES

- Drives LEDs, LCDs or vacuum fluorescent
- Bar or dot display mode externally selectable
- Expandable to display up to 999 steps
- Input voltage range from 2.7V to 12V
- Operates with single supply or TTL
- 16-bit resolution with 12-bit DAC
- Output current programmable from 2 mA to 30 mA

DESCRIPTION

The LM3914 is a monolithic integrated circuit designed to provide serial dot-matrix displays and drive 100 LEDs, providing a large number of applications. The device can be used as a digital-to-analog converter, a digital-to-light converter, or a digital-to-LCD driver and programmed, alternating the need for resistors. The feature is that one chip operator can control up to 999 digits.

The circuit contains its own self-timed reference and access 100-step registers. The loadlessness input interface allows the LM3914 to be connected directly to a microprocessor or microcontroller without the need for glue logic. The byte shift 100 individual refreshes, referred to the previous decade, driver and loadlessly provides a 100-step digital-to-analog converter. The output current is programmable from 2 mA to 30 mA. The device was designed to use the LM3904 (not the LM3904A), and expanded scale factors are available for each digit. The LM3914 is a high-resolution digital-to-analog converter with a resolution of 16 bits. Many LM3914's have been used to display up to 100,000 digits. Both ends of the voltage source can be connected to ground, so the LM3914 can be used as a digital-to-light converter.

The LM3914 is very easy to use as an LED matrix driver. A 12-bit digital timer receives only 1 memory and 16-bit digital data. The digital data is converted to analog current by a 12-bit DAC and fed into the 100-step brightness control. The printed circuit board diagram shows the various internal circuitry.

The LM3914 has a built-in 100-step digital-to-analog converter. This ensures that at no time will 100% of LEDs be lit, and any temporary display is stable. Various read cycles are available to read the data from the 100-step registers.

The digital-to-light display driver from the bar to the dot displays are individual. DC regulated currents, various refresh rates, and various output currents are available. The LM3914 is a high-resolution digital-to-light driver. If the LED of the name is not used, a current limiting function "dot" control can be performed. The LM3914 can also act as a digital-to-LCD driver, and a current limiting function "bar" control can be performed.

The LM3914 is ready to operate from DC = 2.7V to 12V, and the LM3914 is available in an 16-pin plastic (PDIP) package.

The following typical application illustrates adjusting of the reference to a desired value, and proper grounding for accurate operation, and connecting resistors.

No multiple switching or interaction between registers is required.
Input voltages between 2.7V and 12V without damage or risks.
LED driver stages are current regulated.
Output can interface with TTL or CMOS logic.
The LM3914 is a high-resolution digital-to-light converter and can be referenced to a wide range of voltages.

For more information on the LM3914, refer to the LM3914 Data Sheet.

APPLICATION INFORMATION
The LM3914 is a monolithic integrated circuit designed to provide serial dot-matrix displays and drive 100 LEDs, providing a large number of applications. The device can be used as a digital-to-analog converter, a digital-to-light converter, or a digital-to-LCD driver and programmed, alternating the need for resistors. The feature is that one chip operator can control up to 999 digits.

The circuit contains its own self-timed reference and access 100-step registers. The loadlessness input interface allows the LM3914 to be connected directly to a microprocessor or microcontroller without the need for glue logic. The byte shift 100 individual refreshes, referred to the previous decade, driver and loadlessly provides a 100-step digital-to-analog converter. The output current is programmable from 2 mA to 30 mA. The device was designed to use the LM3904 (not the LM3904A), and expanded scale factors are available for each digit. The LM3914 is a high-resolution digital-to-analog converter with a resolution of 16 bits. Many LM3914's have been used to display up to 100,000 digits. Both ends of the voltage source can be connected to ground, so the LM3914 can be used as a digital-to-light converter.

The LM3914 is very easy to use as an LED matrix driver. A 12-bit digital timer receives only 1 memory and 16-bit digital data. The digital data is converted to analog current by a 12-bit DAC and fed into the 100-step brightness control. The printed circuit board diagram shows the various internal circuitry.

The LM3914 has a built-in 100-step digital-to-analog converter. This ensures that at no time will 100% of LEDs be lit, and any temporary display is stable. Various read cycles are available to read the data from the 100-step registers.

The digital-to-light display driver from the bar to the dot displays are individual. DC regulated currents, various refresh rates, and various output currents are available. The LM3914 is a high-resolution digital-to-light driver. If the LED of the name is not used, a current limiting function "dot" control can be performed. The LM3914 can also act as a digital-to-LCD driver, and a current limiting function "bar" control can be performed.

The LM3914 is ready to operate from DC = 2.7V to 12V, and the LM3914 is available in an 16-pin plastic (PDIP) package.

The following typical application illustrates adjusting of the reference to a desired value, and proper grounding for accurate operation, and connecting resistors.

No multiple switching or interaction between registers is required.
Input voltages between 2.7V and 12V without damage or risks.
LED driver stages are current regulated.
Output can interface with TTL or CMOS logic.
The LM3914 is a high-resolution digital-to-light converter and can be referenced to a wide range of voltages.

For more information on the LM3914, refer to the LM3914 Data Sheet.

APPLICATION INFORMATION
The LM3914 is a monolithic integrated circuit designed to provide serial dot-matrix displays and drive 100 LEDs, providing a large number of applications. The device can be used as a digital-to-analog converter, a digital-to-light converter, or a digital-to-LCD driver and programmed, alternating the need for resistors. The feature is that one chip operator can control up to 999 digits.

The circuit contains its own self-timed reference and access 100-step registers. The loadlessness input interface allows the LM3914 to be connected directly to a microprocessor or microcontroller without the need for glue logic. The byte shift 100 individual refreshes, referred to the previous decade, driver and loadlessly provides a 100-step digital-to-analog converter. The output current is programmable from 2 mA to 30 mA. The device was designed to use the LM3904 (not the LM3904A), and expanded scale factors are available for each digit. The LM3914 is a high-resolution digital-to-analog converter with a resolution of 16 bits. Many LM3914's have been used to display up to 100,000 digits. Both ends of the voltage source can be connected to ground, so the LM3914 can be used as a digital-to-light converter.

The LM3914 is very easy to use as an LED matrix driver. A 12-bit digital timer receives only 1 memory and 16-bit digital data. The digital data is converted to analog current by a 12-bit DAC and fed into the 100-step brightness control. The printed circuit board diagram shows the various internal circuitry.

The LM3914 has a built-in 100-step digital-to-analog converter. This ensures that at no time will 100% of LEDs be lit, and any temporary display is stable. Various read cycles are available to read the data from the 100-step registers.

The digital-to-light display driver from the bar to the dot displays are individual. DC regulated currents, various refresh rates, and various output currents are available. The LM3914 is a high-resolution digital-to-light driver. If the LED of the name is not used, a current limiting function "dot" control can be performed. The LM3914 can also act as a digital-to-LCD driver, and a current limiting function "bar" control can be performed.

The LM3914 is ready to operate from DC = 2.7V to 12V, and the LM3914 is available in an 16-pin plastic (PDIP) package.

The following typical application illustrates adjusting of the reference to a desired value, and proper grounding for accurate operation, and connecting resistors.

uPSD3254A

<p>ST</p> <p>uPSD3254A, uPSD3254BV uPSD3253B, uPSD3253BV</p> <p>Flash Programmable System Devices with 8032 Microcontroller Core and 256 Kbit SRAM</p> <hr/> <p>FEATURES SUMMARY</p> <ul style="list-style-type: none"> ■ 8032 MICROCONTROLLER <ul style="list-style-type: none"> - 4MHz to 12MHz, 16bit CPU - 4KB ROM, 128x16bit SRAM - 128x16bit SRAM ■ DATA FLASH MEMORY WITH MEMORY MANAGEMENT <ul style="list-style-type: none"> - User memory from 8KByte program memory to 128KByte data memory - 128KByte SRAM - Block Erase, Sector Erase, Page Erase, Block Programming and EEPROM - Single voltage power operation - High speed data transfer ■ CLOCK, RESET AND POWER <ul style="list-style-type: none"> - SRAM by battery backup option - Power-on self test - Power-down timer - Low power mode - Programmable Watchdog Timer - Power-on Reset, General Purpose - 4 timers - 10 interrupt source modules, 4 global ■ COMMUNICATION PORTS <ul style="list-style-type: none"> - 1 serial port with 8 bit resolution, 8 bit data width, 16 bit address bus - 1 I₂C Master (with software control) - 1 SPI Master (with software control) - 1 SPI Slave - Shift register with up to 16 bit pins - USART with 8 bit resolution - TxD, RxD, CTS, DSR, DTR, DCD - 16 bit timer with software control ■ PROGRAMMING <ul style="list-style-type: none"> - In-Circuit Serial Programming - Target system erase time about 15 seconds 	<p>Figure 1. Packages</p>  <p>TQFP100 TQFP100L</p>
--	---

DS89C450

OPT101

BB Burr-Brown Products
from Texas Instruments

OPT101

MONOLITHIC PHOTODIODE AND SINGLE-SUPPLY TRANSMIPODANCE AMPLIFIER

FEATURES

- MONOLITHIC INTEGRATED CIRCUIT
- INPUT BIAS CURRENT: $<1 \mu A$ at $-5V$
- PHOTODIODE: $10^{-10} - 10^{-15}$ A/W
- LINEARITY: $>99\%$
- HIGH RESPONSIVITY: 500nm (80%)
- BANDWIDTH: 10Hz to 10MHz
- LOW NOISE: 10^{-15} A (typical)
- AVAILABLE IN SOT-89 DIP AND SOT-23 PLASTIC PACKAGE

APPLICATIONS

- MEDICAL INSTRUMENTATION
- LABORATORY EQUIPMENT
- INDUSTRIAL PROCESS CONTROLS
- PHOTOGRAPHIC ANALYZERS
- REMOTE SENSORS
- SMOKE DETECTORS
- CURRENCY CHANGERS

DESCRIPTION

The OPT101 is a monolithic photodiode with shielded transimpedance amplifier. The photodiode has a wide spectral response from 300 nm to 1100 nm. The transimpedance amplifier has a low noise input stage and a high-gain, low-noise, unity-gain feedback stage.

The integrated combination of photodiode and transimpedance amplifier provides a solution to the problem normally encountered in discrete designs such as the need for a separate biasing source and a separate feedback source. The OPT101 is a 0.9 mm package providing a compact, low cost, high performance solution to your optical sensing needs.

The OPT101 features from $2.7V$ to $+5V$ supplies and greater current is only $1\mu A$. It is available in three packages: SOT-89, SOT-23, and a leadless chip carrier. Temperature range is -40°C to $+100^\circ\text{C}$.

CIRCUIT DIAGRAM

SPECTRAL SENSITIVITY

Wavelength (nm)	Responsivity (A/W)
400	0.05
500	0.15
600	0.10
700	0.08
800	0.05
900	0.03
1000	0.02
1100	0.02

TEMPERATURE COEFFICIENT

Temperature ($^\circ\text{C}$)	Current (μA)
-40	0.90
0	0.95
40	1.00
80	1.05
100	1.10

ACKNOWLEDGMENT

Please note that an important feature of many electronic products is their reliability. Burr-Brown is committed to the highest standards of quality and reliability. We are in critical applications of electronic components and we are fully aware of the importance of reliability.

TRADE SHOWS

OPT101 will be shown at the following trade shows:

- **EE Times**, Booth #1000, October 1-3, 1984, San Jose Convention Center, San Jose, CA
- **SEMICON**, Booth #1000, October 1-3, 1984, San Jose Convention Center, San Jose, CA
- **IRF**, Booth #1000, October 1-3, 1984, San Jose Convention Center, San Jose, CA
- **IBS**, Booth #1000, October 1-3, 1984, San Jose Convention Center, San Jose, CA

DISCLAIMER

Burr-Brown Products from Texas Instruments is a trademark of Texas Instruments Incorporated. All other trademarks and registered trademarks are the property of their respective owners.

TEXAS INSTRUMENTS INCORPORATED

SaberTooth motor drive

5. Présenter les datasheets de la sorte économise du papier, donc des arbres, mais requiert de consulter le document .PDF afin de lire adéquatement les datasheets.

74HC14

Pinouts Semiconductor		Product specification							
Hex inverting Schmitt trigger		74HC/HCT14							
FEATURES									
• Output capability: standard									
• I _{OL} : category: SII									
GENERAL DESCRIPTION									
The 74HC/HCT14 are high-speed CMOS devices and are pin compatible with the power-hungry TTL LSTTL. These devices are designed to be drop-in replacements for the corresponding TTL parts.									
The 74HC/HCT14 provide six inverting buffers with Schmitt-trigger action. They are capable of functioning like a standard inverter or as a buffer. The outputs can drive standard logic, Schmitt-trigger logic, and other logic families.									
QUICK REFERENCE DATA									
AND $T_{J\max} = 75^\circ\text{C}$, $T_{J\min} = -15^\circ\text{C}$									
Symbol	Parameter		Condition		Typical				
I_{OL} (I_{OH})	Output current (drive load)		$V_{DD} = 5\text{V}$, $V_{SS} = 0\text{V}$		12				
t_{PD}	Input-to-output propagation time		$V_{DD} = 5\text{V}$, $V_{SS} = 0\text{V}$		17				
t_{IOL}	Input-to-output hold time		$V_{DD} = 5\text{V}$, $V_{SS} = 0\text{V}$		10				
t_{IOH}	Input-to-output setup time		$V_{DD} = 5\text{V}$, $V_{SS} = 0\text{V}$		12				
t_{IOH}	Input-to-output hold time		$V_{DD} = 5\text{V}$, $V_{SS} = 0\text{V}$		17				
Notes									
1. V_{DD} is the supply voltage. The device will operate over a wider power dissipation range (P_D) in HSI mode.									
2. $V_{DD} \geq V_{SS} + 0.3\text{V}$.									
3. Input leakage is measured at $V_{DD} = 5\text{V}$, $V_{SS} = 0\text{V}$.									
4. I_{OL} and I_{OH} are measured at $V_{DD} = 5\text{V}$, $V_{SS} = 0\text{V}$.									
5. t_{PD} is the time required for the output to change from $V_{DD} - 0.4\text{V}$ to $V_{SS} + 0.4\text{V}$.									
6. t_{IOL} is the time required for the output to change from $V_{DD} - 0.4\text{V}$ to $V_{SS} + 0.4\text{V}$.									
7. t_{IOH} is the time required for the output to change from $V_{DD} - 0.4\text{V}$ to $V_{SS} + 0.4\text{V}$.									
8. The HC condition is $V_{DD} \geq 4.5\text{V}$, $V_{SS} \leq 0.3\text{V}$. The HCT condition is $V_{DD} \geq 5\text{V}$, $V_{SS} \leq 0.3\text{V}$.									
ORDERING INFORMATION									
See 74HC/HCT14N/Datasheet for package information.									

LineSensors

Registers:

You read the ADC fine sensor by issuing an ADC read of the programmed address. A single byte representing the status of the sensors is returned, ranging from 0 to 31, where 0 means 16 sensors are white and 31 means 16 sensors see black.

Sensor 1	Sensor 2	Sensor 3	Sensor 4	Sensor 5	Decimal Meaning
0	0	0	0	0	0
1	0	0	0	0	1
0	1	0	0	0	2
1	1	0	0	0	3
0	0	1	0	0	4
1	0	1	0	0	5
0	1	1	0	0	6
1	1	1	0	0	7
0	0	0	1	0	8
1	0	0	1	0	9
0	1	0	1	0	10
1	1	0	1	0	11
0	0	1	1	0	12
1	0	1	1	0	13
0	1	1	1	0	14
1	1	1	1	0	15
0	0	0	0	1	16
1	0	0	0	1	17
0	1	0	0	1	18

PCF8573

DAC6574

LM3914

Texas
Instruments

LM3914
DotBar Display Driver

Rev. C, 12/99

FEATURES

- Drives LED, LCD or vacuum fluorescent displays up to 120 segments by 16 columns
- 16-bit digital control for dot matrix displays
- Operates in a display of 160 segments
- Internal voltage reference from 2.2V to 5V
- Operates at supply voltages from 2.7V to 5V
- Input current requirements less than 100 nA
- Output current programmable from 2 nA to 2 mA

DESCRIPTION

The LM3914 is a high performance dot matrix driver that can drive up to 160 segments in a binary mode. A single 16-bit parallel bus displays from a memory location in a fast page. Control is via a 16-bit bidirectional serial port. The LM3914 is designed to be used in a wide variety of applications such as electronic calculators, scientific calculators, and portable electronic devices.

The LM3914 has a built-in 10-bit DAC and accurate charge pump driver. The fractional-current input accepts speeds down to $1\text{ }\mu\text{s}$, yet never exceeds 100 nA of total current above ground. The DAC can also be programmed to accept a 10-bit digital word. The DAC output current can be programmed to have 10-bit resolution, even with a wide temperature range.

Temperature compensation is provided for the DAC and segment driver. Segment driver feedback is also used on the display driver. The system can drive LCDs of many colors or monochromic liquid crystal displays. The LM3914 can also be used to drive dot-matrix displays, such as dot-matrix liquid crystal displays, or dot-matrix light-emitting diode displays.

The LM3914 has a built-in 10-bit DAC and accurate charge pump driver. The DAC output current can be programmed to have 10-bit resolution, even with a wide temperature range. The DAC output current can be programmed to have 10-bit resolution, even with a wide temperature range.

When in the dot mode, there is a small amount of overhead to switch between segments. This overhead is due to the time required to switch the DAC output current to the next segment. The overhead is dependent on the number of segments in the display and the DAC resolution.

Most of the display driver time is due to the time it takes to update the DACs. The update time varies as the number of segments in the display increases. The update time can be increased or decreased by changing the DAC resolution. The update time is also dependent on the number of segments in the display.

The LM3914 has a built-in 10-bit DAC and accurate charge pump driver. The DAC output current can be programmed to have 10-bit resolution, even with a wide temperature range.

The LM3914 has a built-in 10-bit DAC and accurate charge pump driver. The DAC output current can be programmed to have 10-bit resolution, even with a wide temperature range.

LMP3914 DotBar Display Driver

(Rev. C, 12/99)

Order Number LM3914Z

Order Number LM3914N

Order Number LM3914P

Order Number LM3914A

Order Number LM3914B

Order Number LM3914C

Order Number LM3914D

Order Number LM3914E

Order Number LM3914F

Order Number LM3914G

Order Number LM3914H

Order Number LM3914I

Order Number LM3914J

Order Number LM3914K

Order Number LM3914L

Order Number LM3914M

Order Number LM3914N

Order Number LM3914O

Order Number LM3914P

Order Number LM3914Q

Order Number LM3914R

Order Number LM3914S

Order Number LM3914T

Order Number LM3914U

Order Number LM3914V

Order Number LM3914W

Order Number LM3914X

Order Number LM3914Y

Order Number LM3914Z

Order Number LM3914AA

Order Number LM3914AB

Order Number LM3914AC

Order Number LM3914AD

Order Number LM3914AE

Order Number LM3914AF

Order Number LM3914AG

Order Number LM3914AH

Order Number LM3914AI

Order Number LM3914AJ

Order Number LM3914AK

Order Number LM3914AL

Order Number LM3914AM

Order Number LM3914AN

Order Number LM3914AO

Order Number LM3914AP

Order Number LM3914AQ

Order Number LM3914AR

Order Number LM3914AS

Order Number LM3914AT

Order Number LM3914AU

Order Number LM3914AV

Order Number LM3914AW

Order Number LM3914AX

Order Number LM3914AY

Order Number LM3914AZ

Order Number LM3914AAZ

Order Number LM3914ABZ

Order Number LM3914ACZ

Order Number LM3914ADZ

Order Number LM3914AEZ

Order Number LM3914AFZ

Order Number LM3914AGZ

Order Number LM3914AHZ

Order Number LM3914AZZ

Order Number LM3914AAZZ

Order Number LM3914ABZZ

Order Number LM3914ACZZ

Order Number LM3914ADZZ

Order Number LM3914AEZZ

Order Number LM3914AFZZ

Order Number LM3914AGZZ

Order Number LM3914AHZZ

Order Number LM3914AZZZ

Order Number LM3914AAZZZ

Order Number LM3914ABZZZ

Order Number LM3914ACZZZ

Order Number LM3914ADZZZ

Order Number LM3914AEZZZ

Order Number LM3914AFZZZ

Order Number LM3914AGZZZ

Order Number LM3914AHZZZ

Order Number LM3914AZZZZ

Order Number LM3914AAZZZZ

Order Number LM3914ABZZZZ

Order Number LM3914ACZZZZ

Order Number LM3914ADZZZZ

Order Number LM3914AEZZZZ

Order Number LM3914AFZZZZ

Order Number LM3914AGZZZZ

Order Number LM3914AHZZZZ

Order Number LM3914AZZZZZ

Order Number LM3914AAZZZZZ

Order Number LM3914ABZZZZZ

Order Number LM3914ACZZZZZ

Order Number LM3914ADZZZZZ

Order Number LM3914AEZZZZZ

Order Number LM3914AFZZZZZ

Order Number LM3914AGZZZZZ

Order Number LM3914AHZZZZZ

Order Number LM3914AZZZZZZ

Order Number LM3914AAZZZZZ

Order Number LM3914ABZZZZZ

Order Number LM3914ACZZZZZ

Order Number LM3914ADZZZZZ

Order Number LM3914AEZZZZZ

Order Number LM3914AFZZZZZ

Order Number LM3914AGZZZZZ

Order Number LM3914AHZZZZZ

Order Number LM3914AZZZZZZ

Order Number LM3914AAZZZZZ

Order Number LM3914ABZZZZZ

Order Number LM3914ACZZZZZ

Order Number LM3914ADZZZZZ

Order Number LM3914AEZZZZZ

Order Number LM3914AFZZZZZ

Order Number LM3914AGZZZZZ

Order Number LM3914AHZZZZZ

Order Number LM3914AZZZZZZ

Order Number LM3914AAZZZZZ

Order Number LM3914ABZZZZZ

Order Number LM3914ACZZZZZ

Order Number LM3914ADZZZZZ

Order Number LM3914AEZZZZZ

Order Number LM3914AFZZZZZ

Order Number LM3914AGZZZZZ

Order Number LM3914AHZZZZZ

Order Number LM3914AZZZZZZ

Order Number LM3914AAZZZZZ

Order Number LM3914ABZZZZZ

Order Number LM3914ACZZZZZ

Order Number LM3914ADZZZZZ

Order Number LM3914AEZZZZZ

Order Number LM3914AFZZZZZ

Order Number LM3914AGZZZZZ

Order Number LM3914AHZZZZZ

Order Number LM3914AZZZZZZ

Order Number LM3914AAZZZZZ

Order Number LM3914ABZZZZZ

Order Number LM3914ACZZZZZ

Order Number LM3914ADZZZZZ

Order Number LM3914AEZZZZZ

Order Number LM3914AFZZZZZ

Order Number LM3914AGZZZZZ

Order Number LM3914AHZZZZZ

Order Number LM3914AZZZZZZ

Order Number LM3914AAZZZZZ

Order Number LM3914ABZZZZZ

Order Number LM3914ACZZZZZ

Order Number LM3914ADZZZZZ

Order Number LM3914AEZZZZZ

Order Number LM3914AFZZZZZ

Order Number LM3914AGZZZZZ

Order Number LM3914AHZZZZZ

Order Number LM3914AZZZZZZ

Order Number LM3914AAZZZZZ

Order Number LM3914ABZZZZZ

Order Number LM3914ACZZZZZ

Order Number LM3914ADZZZZZ

Order Number LM3914AEZZZZZ

Order Number LM3914AFZZZZZ

Order Number LM3914AGZZZZZ

Order Number LM3914AHZZZZZ

Order Number LM3914AZZZZZZ

Order Number LM3914AAZZZZZ

Order Number LM3914ABZZZZZ

Order Number LM3914ACZZZZZ

Order Number LM3914ADZZZZZ

Order Number LM3914AEZZZZZ

Order Number LM3914AFZZZZZ

Order Number LM3914AGZZZZZ

Order Number LM3914AHZZZZZ

Order Number LM3914AZZZZZZ

Order Number LM3914AAZZZZZ

Order Number LM3914ABZZZZZ

Order Number LM3914ACZZZZZ

Order Number LM3914ADZZZZZ

Order Number LM3914AEZZZZZ

Order Number LM3914AFZZZZZ

Order Number LM3914AGZZZZZ

Order Number LM3914AHZZZZZ

Order Number LM3914AZZZZZZ

Order Number LM3914AAZZZZZ

Order Number LM3914ABZZZZZ

Order Number LM3914ACZZZZZ

Order Number LM3914ADZZZZZ

Order Number LM3914AEZZZZZ

Order Number LM3914AFZZZZZ

Order Number LM3914AGZZZZZ

Order Number LM3914AHZZZZZ

Order Number LM3914AZZZZZZ

Order Number LM3914AAZZZZZ

Order Number LM3914ABZZZZZ

Order Number LM3914ACZZZZZ

Order Number LM3914ADZZZZZ

Order Number LM3914AEZZZZZ

Order Number LM3914AFZZZZZ

Order Number LM3914AGZZZZZ

Order Number LM3914AHZZZZZ

Order Number LM3914AZZZZZZ

Order Number LM3914AAZZZZZ

Order Number LM3914ABZZZZZ

Order Number LM3914ACZZZZZ

Order Number LM3914ADZZZZZ

Order Number LM3914AEZZZZZ

Order Number LM3914AFZZZZZ

Order Number LM3914AGZZZZZ

Order Number LM3914AHZZZZZ

Order Number LM3914AZZZZZZ

Order Number LM3914AAZZZZZ

Order Number LM3914ABZZZZZ

Order Number LM3914ACZZZZZ

Order Number LM3914ADZZZZZ

Order Number LM3914AEZZZZZ

Order Number LM3914AFZZZZZ

Order Number LM3914AGZZZZZ

Order Number LM3914AHZZZZZ

Order Number LM3914AZZZZZZ

Order Number LM3914AAZZZZZ

Order Number LM3914ABZZZZZ

Order Number LM3914ACZZZZZ

Order Number LM3914ADZZZZZ

Order Number LM3914AEZZZZZ

Order Number LM3914AFZZZZZ

Order Number LM3914AGZZZZZ

Order Number LM3914AHZZZZZ

Order Number LM3914AZZZZZZ

Order Number LM3914AAZZZZZ

Order Number LM3914ABZZZZZ

Order Number LM3914ACZZZZZ

Order Number LM3914ADZZZZZ

Order Number LM3914AEZZZZZ

Order Number LM3914AFZZZZZ

Order Number LM3914AGZZZZZ

Order Number LM3914AHZZZZZ

Order Number LM3914AZZZZZZ

Order Number LM3914AAZZZZZ

Order Number LM3914ABZZZZZ

Order Number LM3914ACZZZZZ

Order Number LM3914ADZZZZZ

Order Number LM3914AEZZZZZ

Order Number LM3914AFZZZZZ

Order Number LM3914AGZZZZZ

Order Number LM3914AHZZZZZ

Order Number LM3914AZZZZZZ

Order Number LM3914AAZZZZZ

Order Number LM3914ABZZZZZ

Order Number LM3914ACZZZZZ

Order Number LM3914ADZZZZZ

Order Number LM3914AEZZZZZ

Order Number LM3914AFZZZZZ

<div data-bbox="375 3152 536 316

uPSD3254A

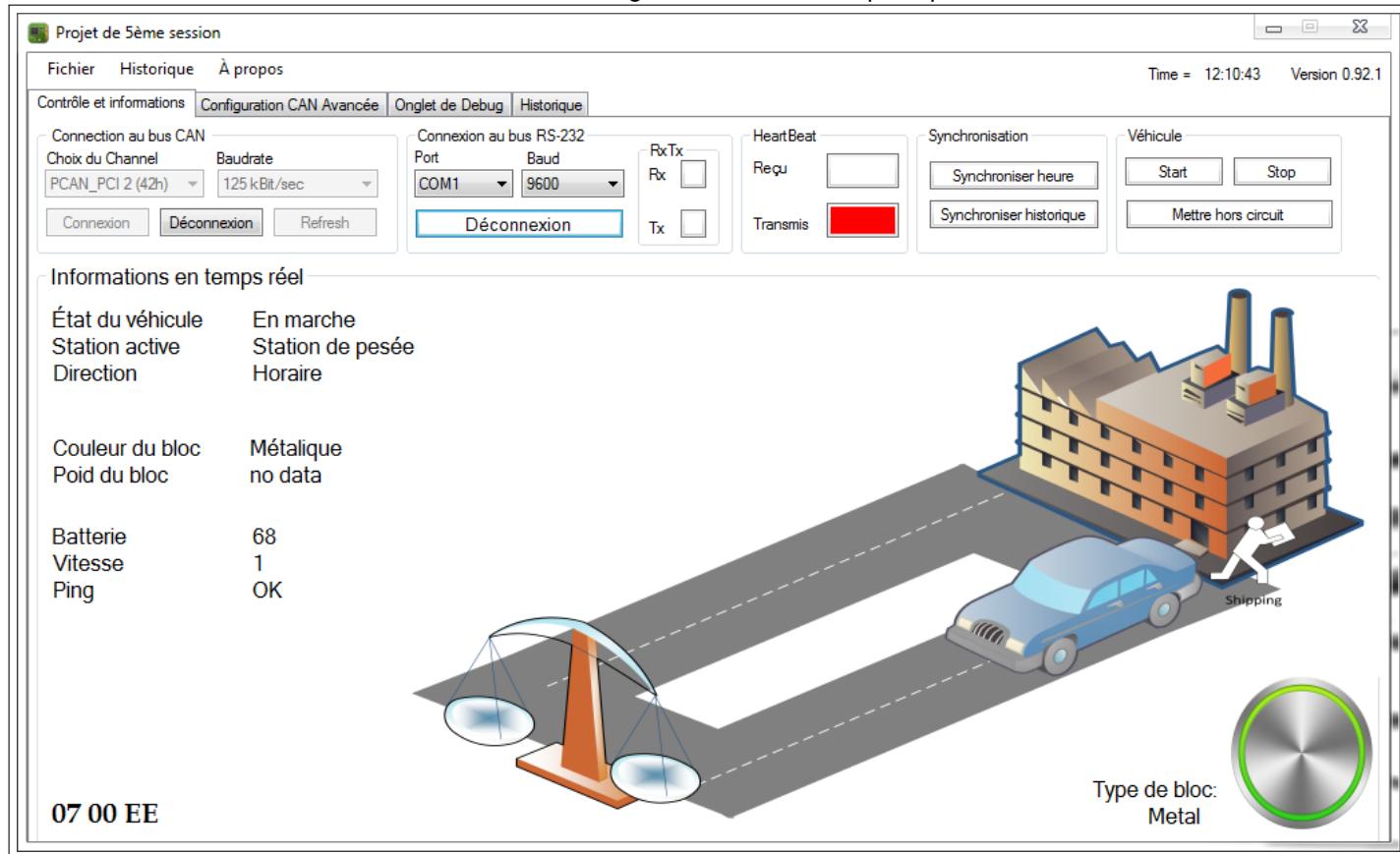
DS89C450

OPT101

SaberTooth motor drive

3 Interface PC

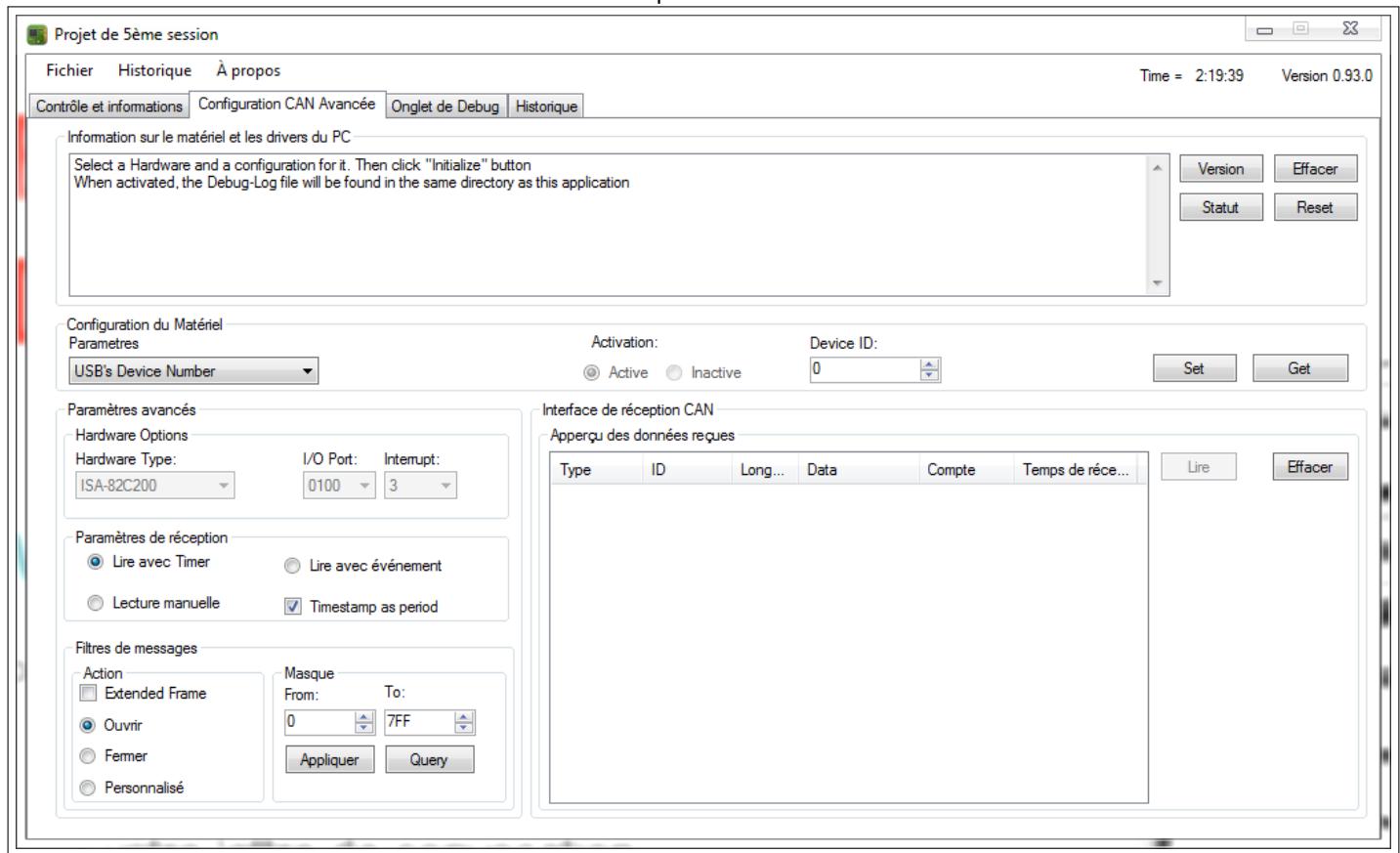
FIGURE 5 – Programme de contrôle principal



Notre programme, écrit en C# à l'aide de Visual Studio, peut se connecter au bus CAN via une carte SPI⁶ et au bus RS232 via un câble DB9 ou USB⁷. La connexion RS232 sert à l'envoi et à la réception du HeartBeat afin d'informer le SOC8200 si l'ordinateur en venait à connaître une défaillance. De plus, des témoins lumineux s'allument en présence de données transmises et reçues. Le programme peut lire l'heure interne du PC et, par un simple clic sur le bouton « Synchroniser », inscrire son heure de référence sur la station no.1 via le bus CAN.

6. Spécifier le fabricant
7. S'il y a présence d'un FTDI

FIGURE 6 – Options CAN avancées



Il est possible d'utiliser des fonctionnalités CAN avancées telles que les masques et filtres de données. De plus, cette fenêtre permet de visualiser les données CAN reçues à l'état brut et non traitées, ce qui peut s'avérer utile pour du débogage.

3.1 Gestion de l'historique

FIGURE 7 – Historique des actions

```

Projet de 5ème session
Fichier Historique À propos
Contrôle et informations Configuration CAN Avancée Onglet de Debug Historique
Time = 12:11:51 Version 0.92.1

Le véhicule est à la table FESTO
Heartbeat Envoyé
Heartbeat Reçu

2014-12-10 12:11:45
Error sending data on CAN bus (try catch error)

2014-12-10 12:11:46
Le bloc est orange
Heartbeat Envoyé
Heartbeat Reçu
Heartbeat Envoyé
Heartbeat Reçu

2014-12-10 12:11:47
Le véhicule est à la station de pesée
Heartbeat Envoyé
Heartbeat Reçu

2014-12-10 12:11:49
Le véhicule est à la table FESTO
Heartbeat Envoyé
Heartbeat Reçu

2014-12-10 12:11:50
Transfert de l'historique du SOC vers le PC
Heartbeat Envoyé
Heartbeat Reçu
Heartbeat Reçu

2014-12-10 12:11:51
Le véhicule est en marche
Heartbeat Envoyé

```

Toute action effectuée via le programme ainsi que toute donnée ayant transité sur le bus CAN, RS232 et TCP/IP est cataloguée en bonne et due forme dans un historique qu'il est possible de consulter et sauvegarder à tout moment.

3.1.1 Exemple d'historique typique

Insérer copié-collé de l'historique ici

3.2 Structure du programme

3.2.1 Les Ghosts Labels

Un ghost label est un label de texte présent sur l'interface, mais définit comme invisible. Il est donc impossible pour l'usager de le voir et d'y accéder. Leurs principales utilités est de faire office de variable globale afin de passer des paramètres entre fonctions et de déclencher des événements systèmes lorsqu'ils sont lus ou modifiés.

3.3 Explication des trames

3.4 Ordre de gestion des tâches

4 Logiciel du SOC8200

4.1 Description du programme

D'un commun accord de l'équipe, le programme du SOC2800 est écrit en script Shell. La principale raison de ce choix est Sourcery Codebench lui-même. La gestion des projets avec Sourcery est un cauchemar. Quant à la nécessité de sauvegarder pour compiler et d'utiliser une machine virtuelle, elles ne viennent qu'aggraver la situation. De plus, son gestionnaire de licence⁸ frustré quiconque souhaite l'utiliser. L'utilisation du script shell est à la fois plus simple et permet de faire plus en moins de temps.

4.2 Schéma bloc du script shell

De tous les scripts, *Projet.sh* est le maître, et contient l'équivalent du main. Ce fichier initialise le port série, le bus CAN, la lecture et l'envoi des heartbeat en asynchrone, ainsi que la lecture du clavier USB. Le script *Projet.sh* peut aussi prendre la relève du bus CAN si le PC est dans l'incapacité d'assurer ses fonctions. Quant aux autres scripts, soit *connexion.sh*, *RxCan.sh*, *tcp.sh* et *Envoi.sh*, ce ne sont que des fonctionnalités que *Projet.sh* appelle en asynchrone.

4.3 Gestion des processus et du temps de CPU

Le seul processus synchrone est script principal *Projet.sh*. Tous les scripts appelés par *Projet.sh* ainsi que leurs processus enfants sont exécutés en asynchrone. Lorsqu'un script doit passer un paramètre à un autre script, un fichier est créé (à l'aide de la commande echo ou cat) afin de contenir le paramètre en question. Les processus impliqués se contentent de lire des fichiers.

4.4 Format et récupération des logs

Toutes les trames CAN reçues sont enregistrées dans le fichier «histocan» dont voici un court aperçu :

```
can0      3  [7] 06 00 0E 0F 06 00 00
can0      3  [7] 00 00 0E 0F 08 00 00
can0      3  [7] 00 00 0E 0F 09 00 00
can0      3  [7] 00 00 0E 0F 0A 00 00
can0      3  [7] 00 01 0E 0F 0B 00 00
can0      3  [7] 2B 1E 0E 0F 0E 00 00
can0      3  [7] 00 00 0E 0F 1C 00 00
can0      3  [7] 00 01 0E 0F 1C 00 00
```

De plus, un autre fichier (histocandate) contient l'heure et la date des trames reçues.

```
*****
can0 3 [7] 00 00 0E 0F 09 00 00
Wed Dec 10 14:15:07 UTC 2014
```

```
*****
can0 3 [7] 00 00 0E 0F 0A 00 00
Wed Dec 10 14:15:08 UTC 2014
```

```
*****
can0 3 [7] 00 01 0E 0F 0B 00 00
Wed Dec 10 14:15:09 UTC 2014
```

8. L'un des membres de l'équipe fait dire que les licences sont une horreur inacceptable et inexcusable sur un système Linux

4.5 Liste des tests et logiciels

5 Logiciel de la station 1 et du bolide

Le programme de la station 1 et du bolide est écrit en C++ à l'aide d'IAR WorkBench 8.20 et la compilation conditionnelle offre de compiler pour chacune des deux stations mentionnées. De plus, la compilation conditionnelle permet au bolide d'utiliser soit une carte d'extension I2C, soit une carte d'extension SPI pour contrôler ses moteurs et ses divers capteurs.

5.1 La station no.1

La station no.1 est composée de uPSD et s'appelle Bloc no.2 dans le cahier de consignes. Cette station reçoit les directives du PC par le BUS CAN et les expédie sur le bus CAN (et vice-versa) aux endroits appropriés. C'est aussi à cette station qu'incombe la tâche de communiquer avec le bolide et la table FESTO via des Xbee.

5.2 La station no.2

La station no.2 (qui s'appelle Bloc no.3 dans le cahier de consignes) est composée de la table FESTO, de la carte uPSD, de la carte d'extensions IO que nous avons réalisées et d'un Xbee.

5.4 Le bolide

Composante	Adresse I2C	Description
MAX1236	0x68	Convertisseur analogique-numérique
DS1307	0xD0	Circuit d'horloge RTC
PCF8574	0x40	I/O Expander pour bus I2C
DAC6574	0x98	Convertisseur numérique-analogique
OPT101	0x50	Suiveur de ligne

TABLE 4 – Informations sur le bus I2C du bolide

5.3 Schéma des héritages de classes

Quelles classes utilisent quelles autres classes dans notre code ?

FIGURE 8 – Héritage de la classe de contrôle du véhicule

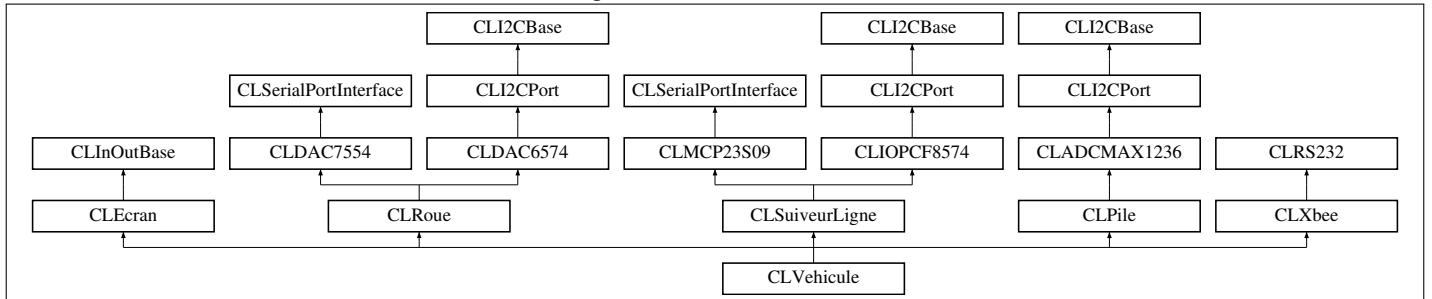


FIGURE 9 – Héritage de la classe de contrôle de la station no.1

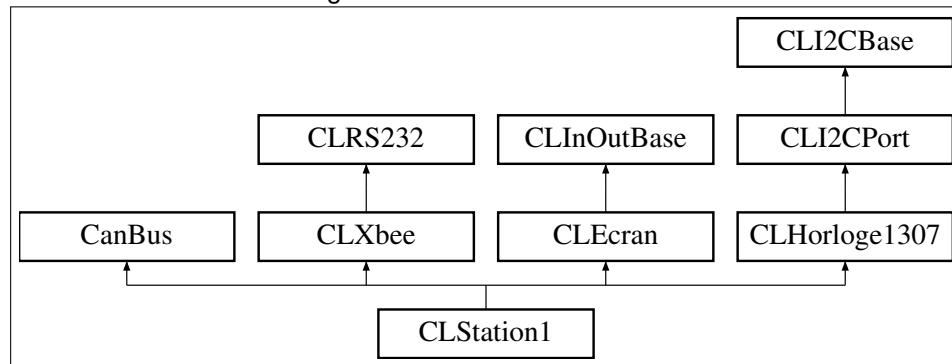


FIGURE 10 – Qui hérite du SPI ?

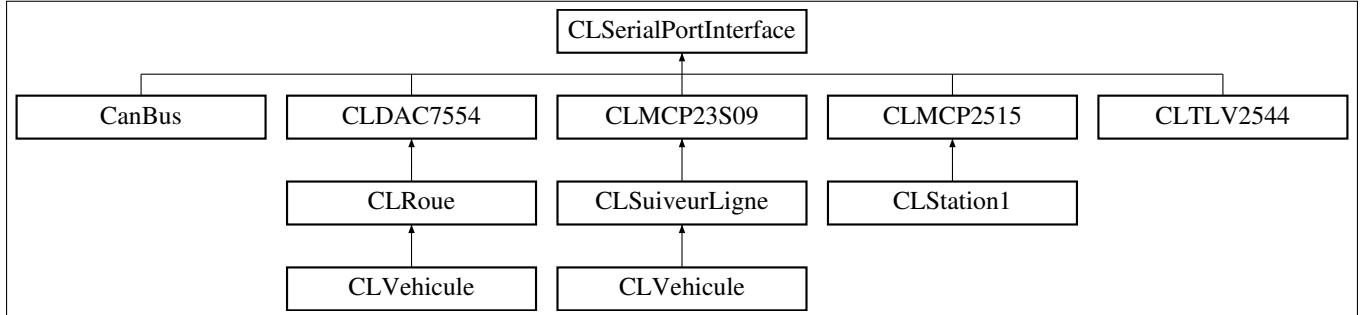
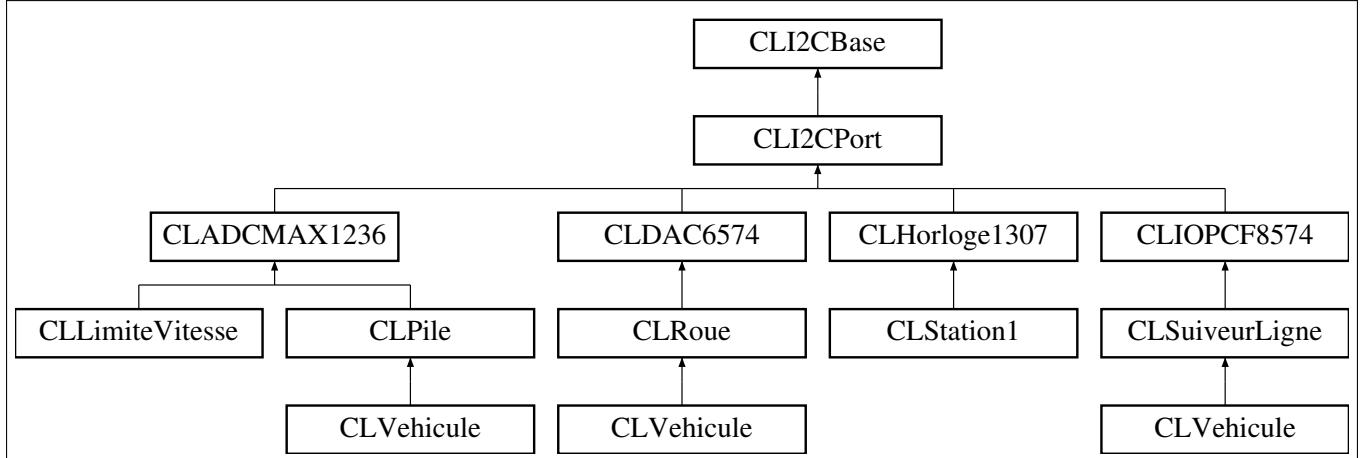


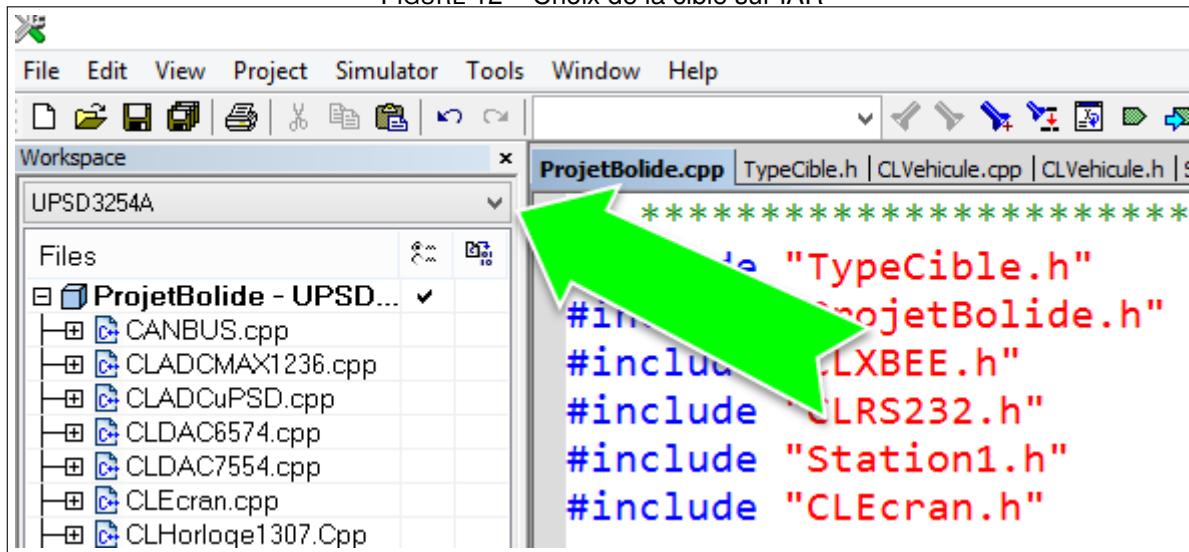
FIGURE 11 – Qui hérite de l'I2C ?



5.5 Procédure de compilation sur IAR

Sur IAR, vous pouvez utiliser le menu déroulant, illustré à la figure suivante, afin de compiler le code pour la carte Dallas ou pour la carte uPSD. Pour lancer une compilation, rien de plus simple que d'appuyer sur F6.

FIGURE 12 – Choix de la cible sur IAR



De plus, des paramètres de compilation optionnelle vous permettent, via la décommentation, de compiler le code pour la carte Dallas ou uPSD, pour la carte d'extension I2C ou SPI et pour un capteur de ligne à 3 ou à 5 photorécepteurs.

Appercu des directives de compilation conditionnelles

```

#ifndef UPSD3254A
#ifndef DALLAS89C450
#ifndef SPI.DALLAS
#ifndef I2C.DALLAS
#ifndef PCF.5.CAPTEURS
#ifndef PCF.3.CAPTEURS

```

5.6 Procédure de vérification

Pour les vérification, rien de plus simple. Il suffit d'envoyer le fichier .hex dans le microcontrôleur et d'observer visuellement le fonctionnement du montage. Certains appellent cette technique « débogage à la chandelle⁹. » Cela étant dit, afficher des trames et autres données sur un écran LCD aide grandement.

9. Salut Étienne !!!

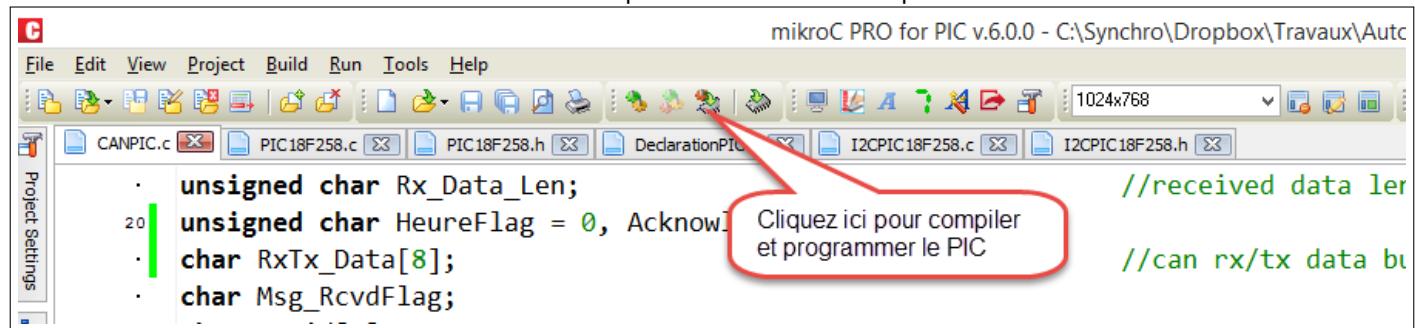
6 Logiciel du module PIC18F258

6.1 Description du fonctionnement du programme

La structure du programme est on ne peut plus classique, car il s'agit d'un programme écrit en C et disposant d'un bon vieux `while(1)` à l'intérieur du `void main(void)`. En bref, le programme initialise d'abord les différents registres du PIC, puis le bus CAN, I2C et RS232. Quant à la boucle, elle lit le poids du bloc, traduit les trames CAN en données RS232 et expédie le tout au... Où déjà ?

6.2 Procédure de compilation sur MPLAB

FIGURE 13 – Compiler avec MicroC PRO pour PIC



6.3 Procédure de vérification

7 Schémas OrCAD

FIGURE 14 – Schémas carte IO I2C, page 1

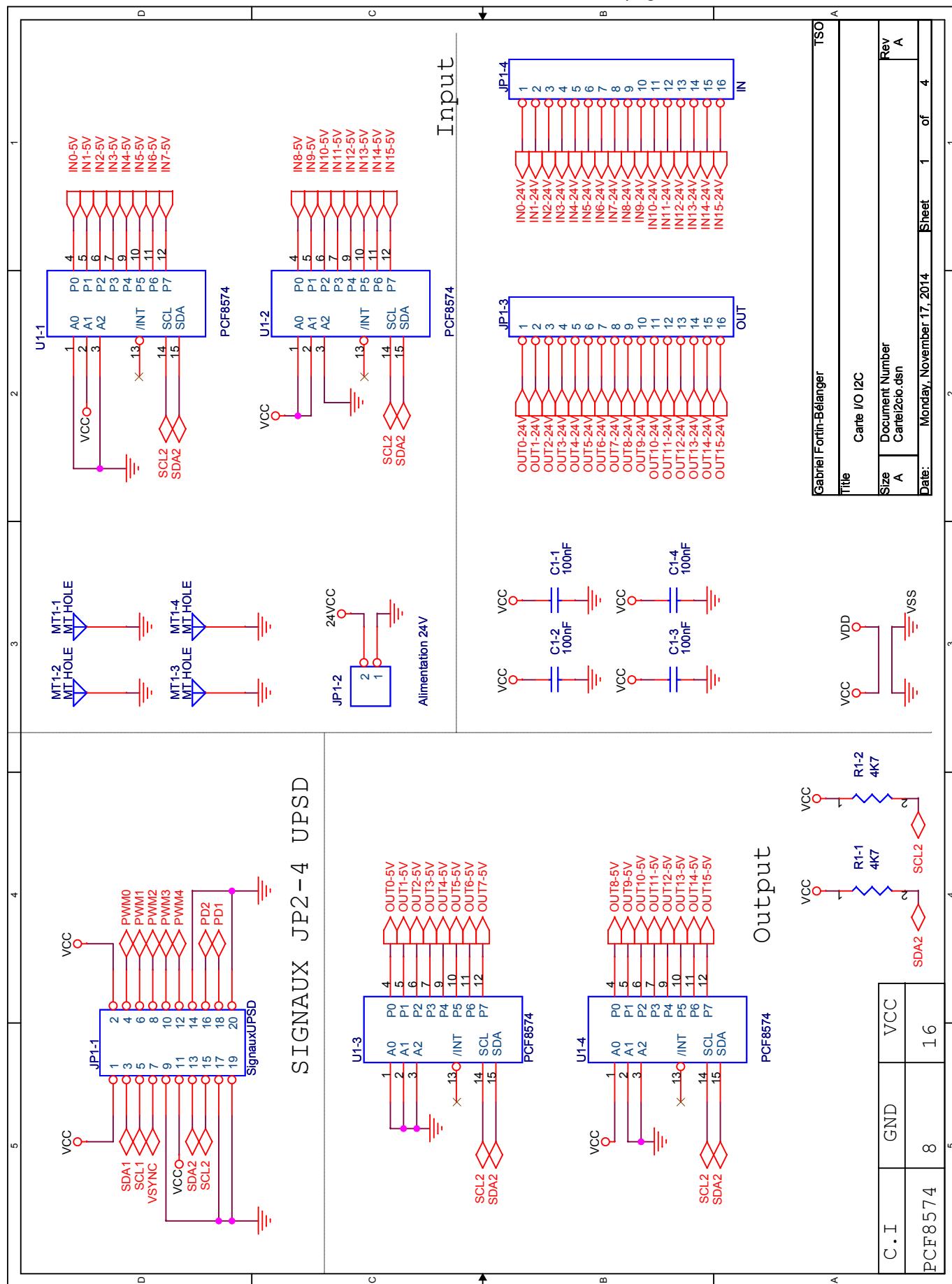


FIGURE 15 – Schémas carte IO I2C, page 2

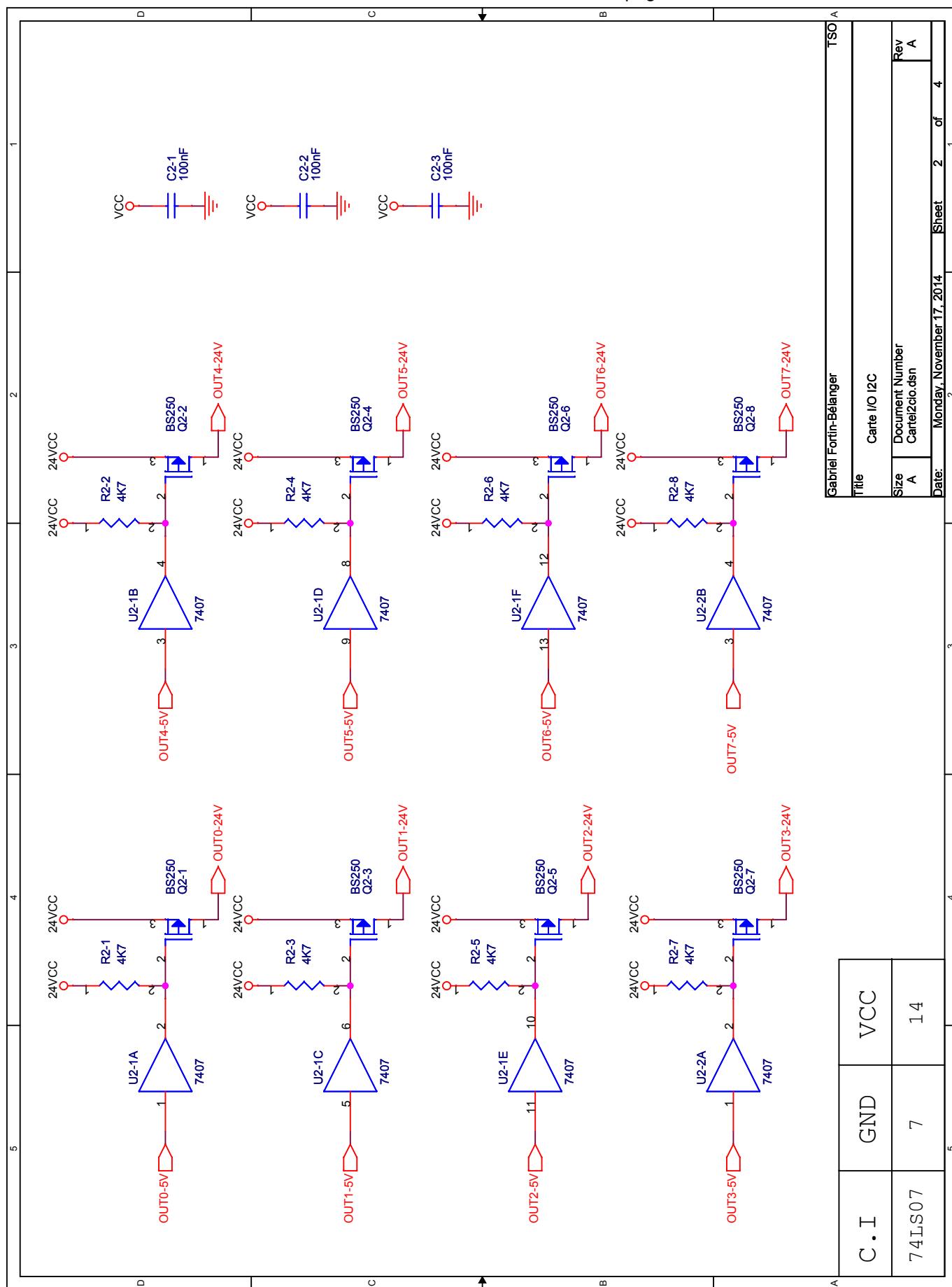


FIGURE 16 – Schémas carte IO I2C, page 3

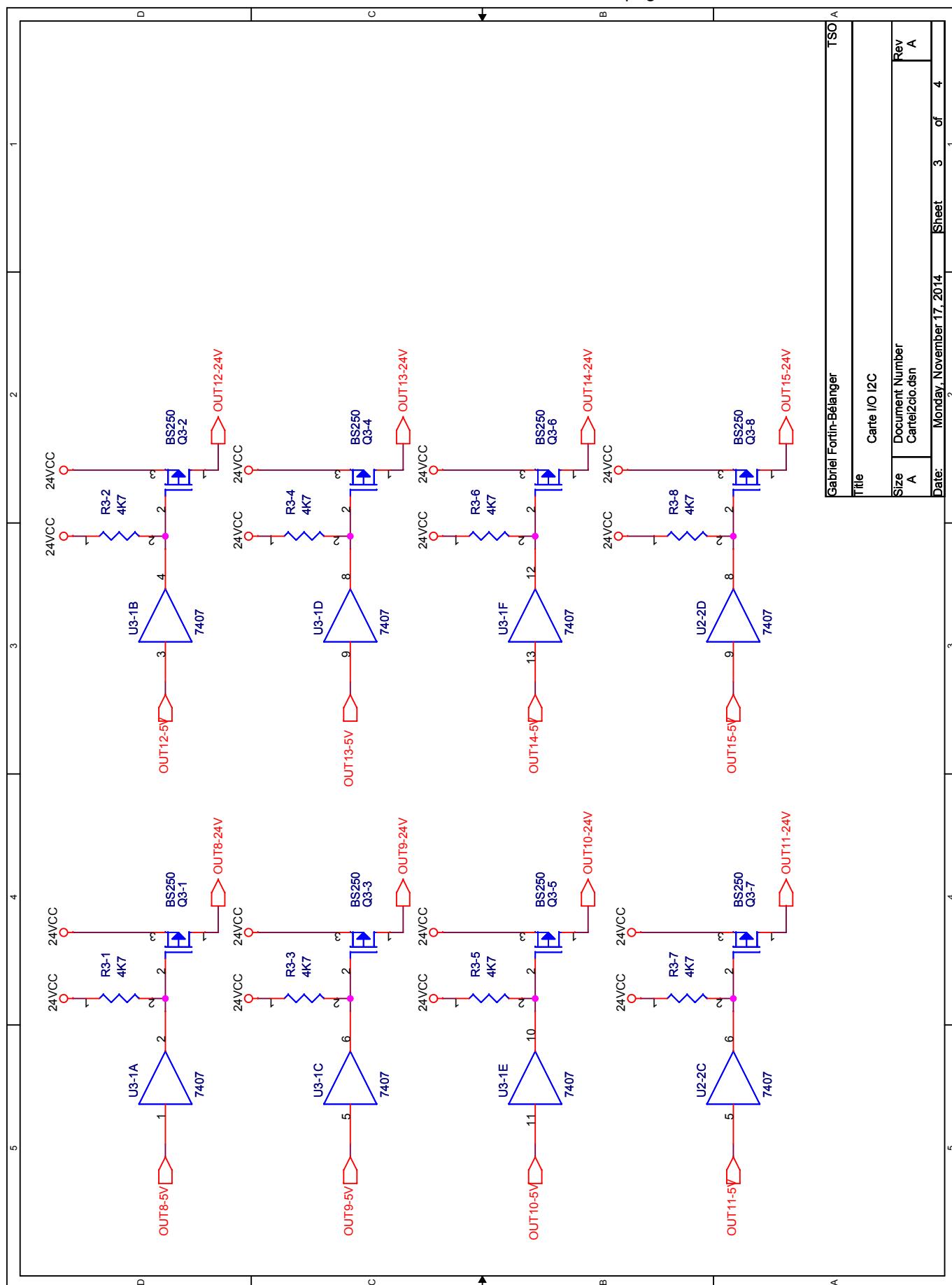
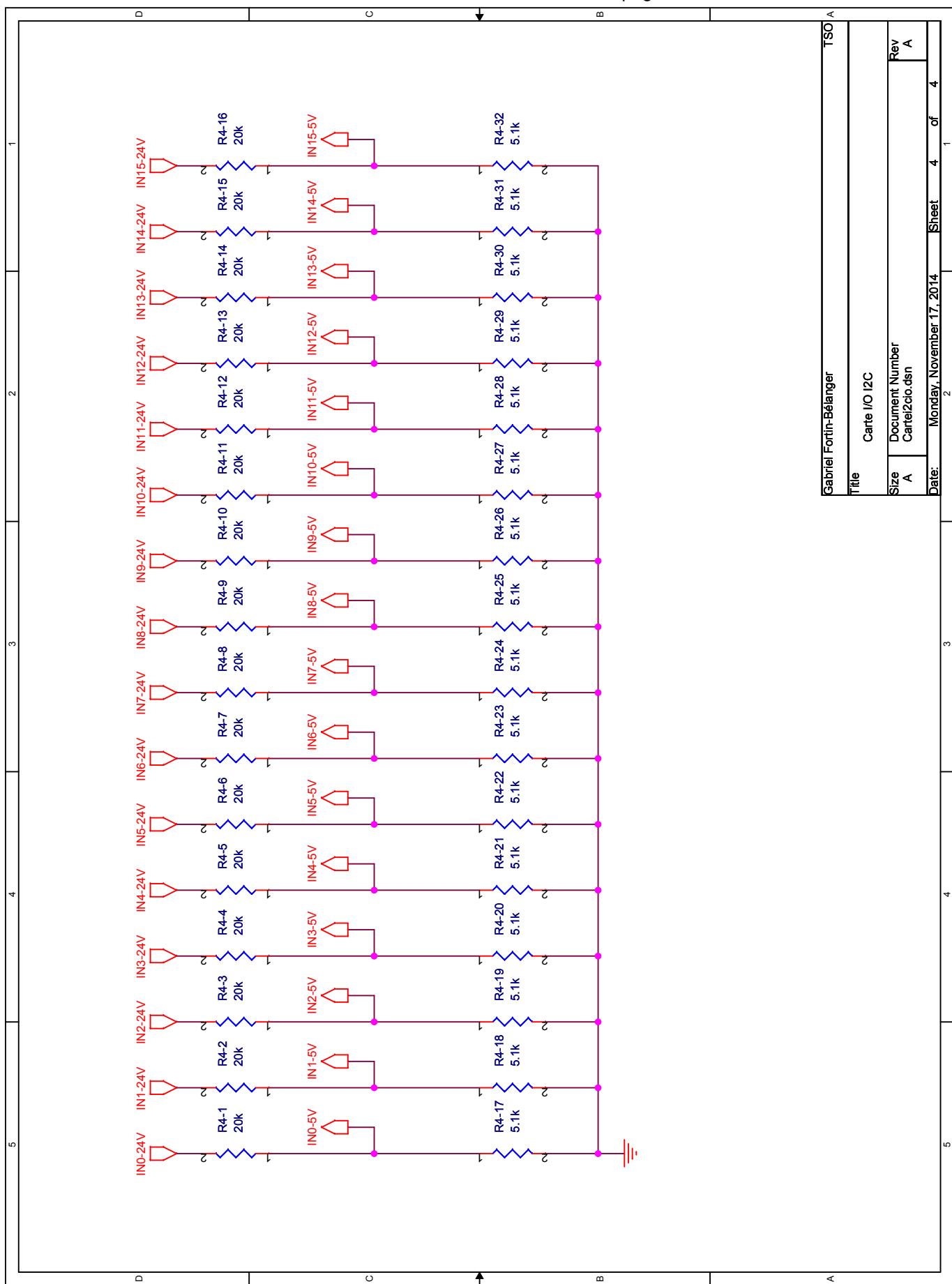


FIGURE 17 – Schémas carte IO I2C, page 4



TSO A	
Gabriel Fortin-Bélanger	
Title	Carte I/O I2C
Size	A
Date:	Monday, November 17, 2014
Rev	A
Sheet	4 of 4
2	1

8 Fichiers Gerbers

Une carte d'extension, dont voici les images GERBER¹⁰, à été réalisée avec OrCAD 16.2 et gravée à l'aide de la rutilante LPKF départementale.

FIGURE 18 – Couche TOP

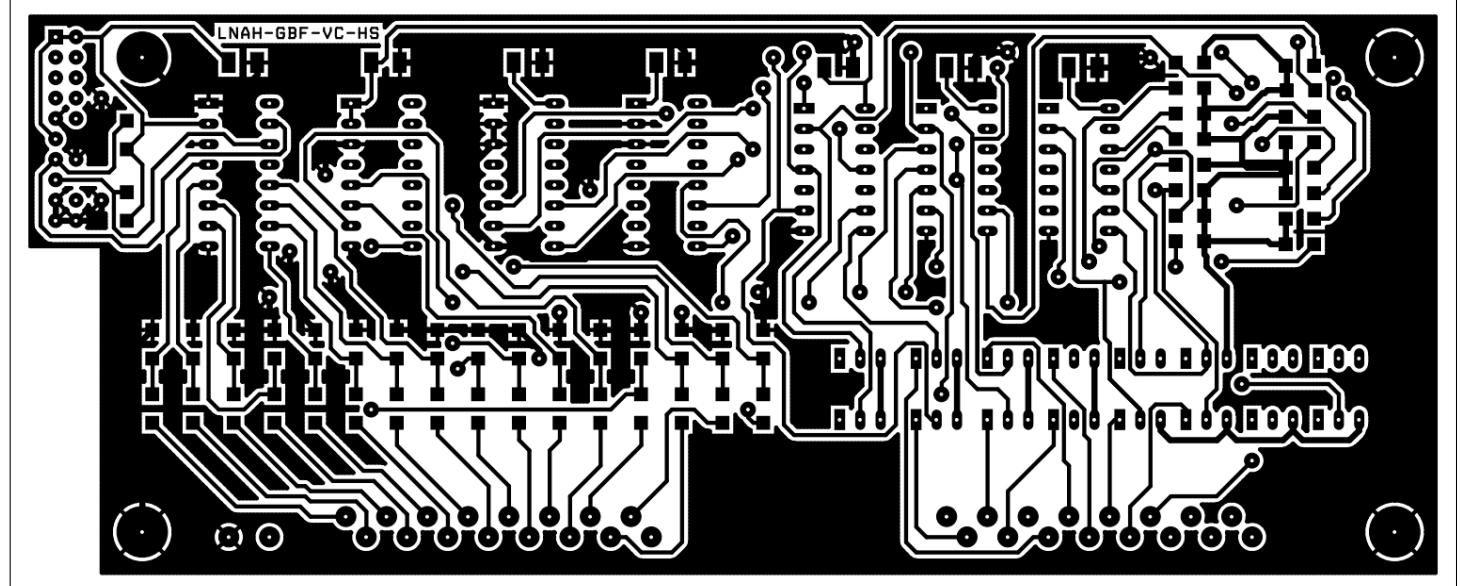


FIGURE 19 – Couche BOT

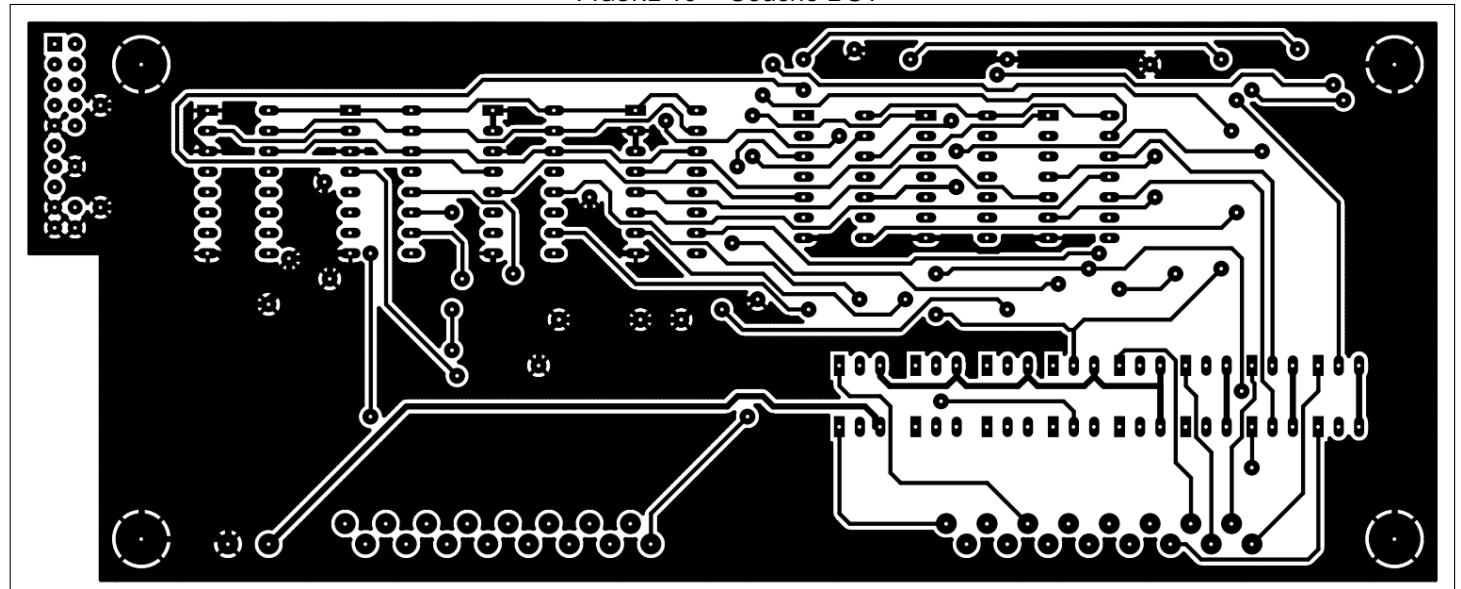


FIGURE 20 – Silk Screen TOP

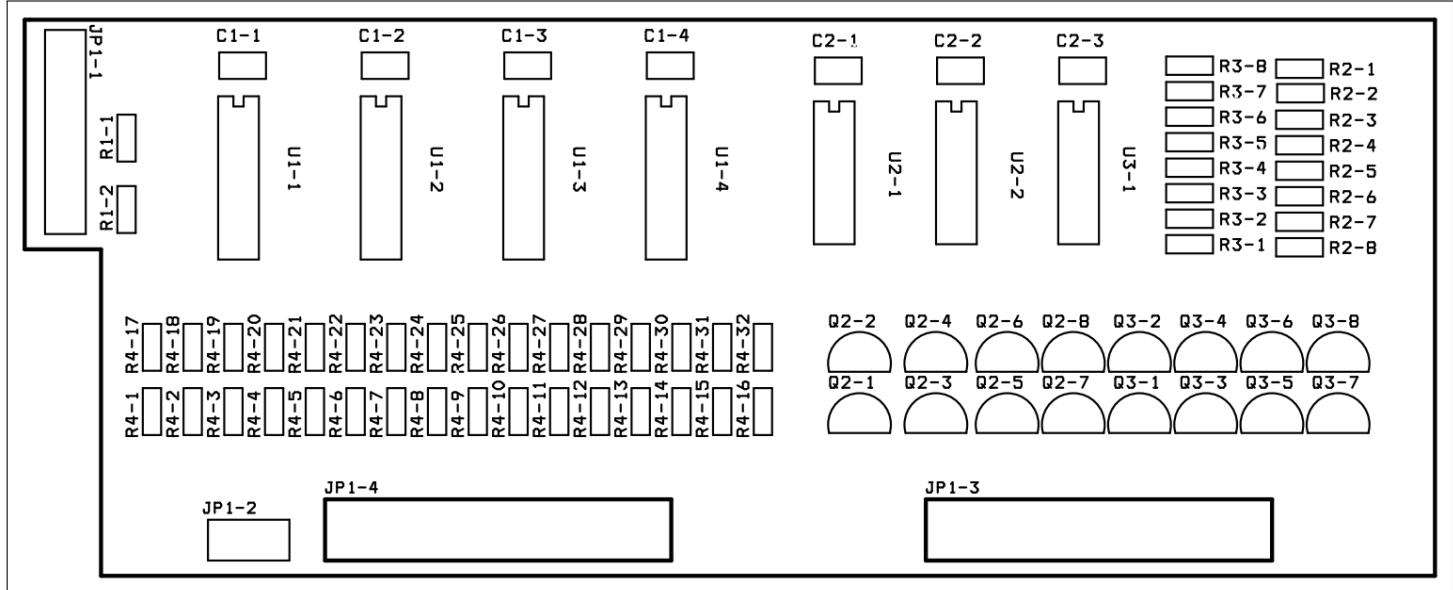


FIGURE 21 – Solder mask TOP

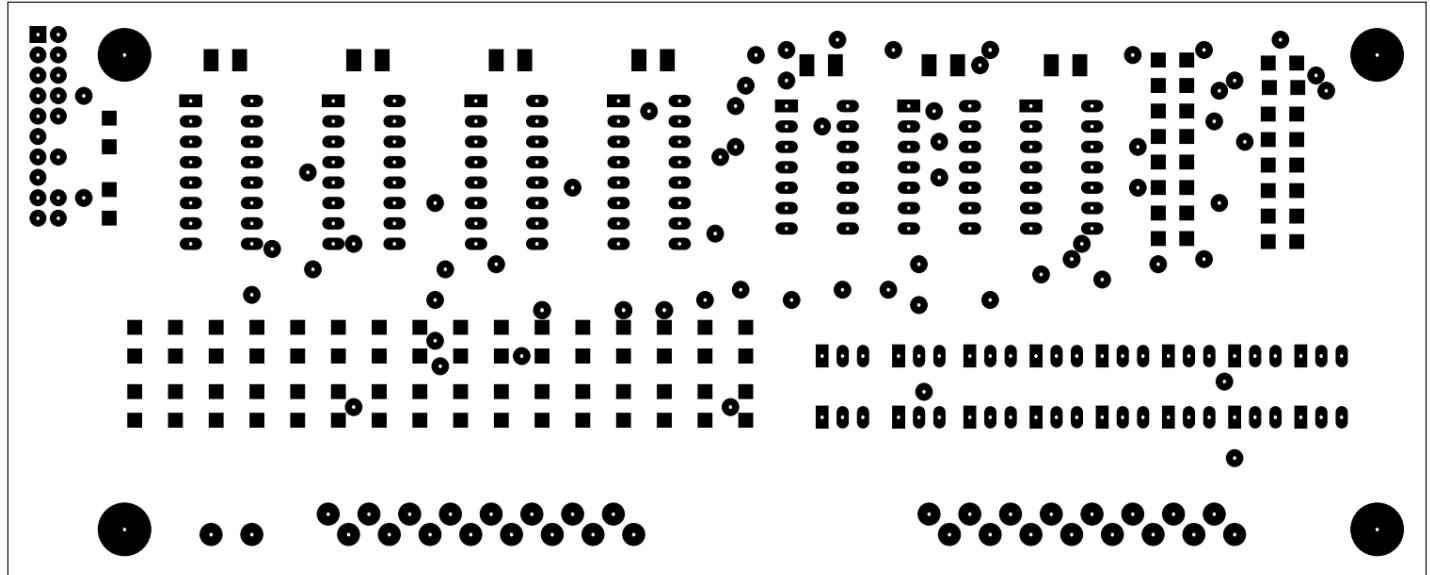
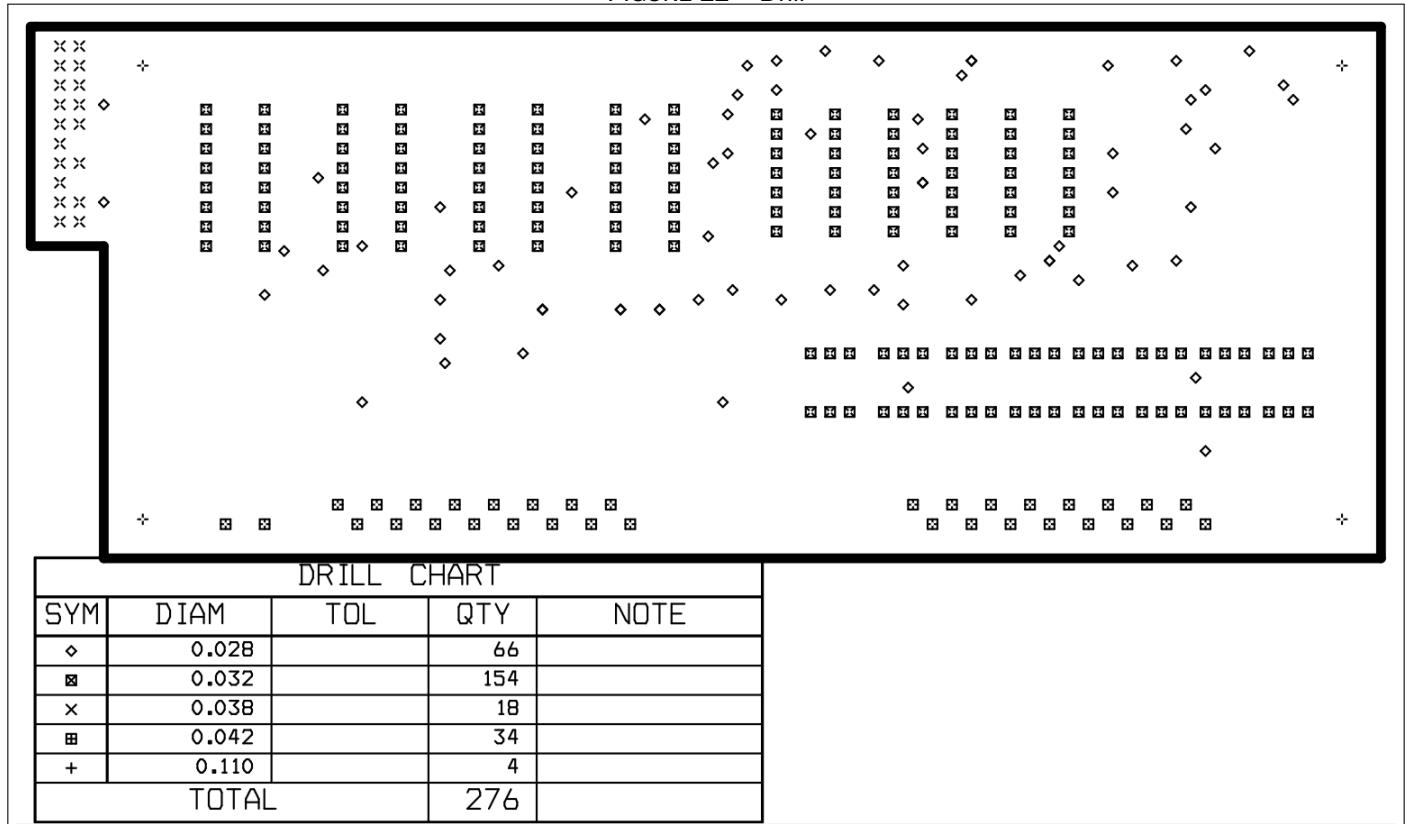


FIGURE 22 – Drill



9 Calculs

9.1 Calcul du pas de conversion de la pile

$$V_{MAX} = 10.8V \Rightarrow MAX1236 = 12bit \Rightarrow Pas = \frac{4K\Omega \cdot \left(\frac{10.8V}{10K\Omega} \right)}{2^{12}(pas)} = 1.33(mV/pas)$$

9.2 Calcul du baudrate

$$Baud = \frac{2^{SMOD}}{32} \cdot \frac{Crystal(Hz)}{12 \cdot (256 - TH1)}$$

Alors...

$$\overbrace{9600 = \frac{2^1}{32} \cdot \frac{24 \cdot 10^6(Hz)}{12 \cdot (256 - 243)}}^{uPSD3254} \Leftarrow \& \Rightarrow \overbrace{9600 = \frac{2^0}{32} \cdot \frac{11.0597 \cdot 10^6(Hz)}{12 \cdot (256 - 253)}}^{DS89C450}$$

10 Conclusions

10.1 Ce que le projet m'a apporté

10.1.1 Vincent

dd

10.1.2 Hicham

dd

10.1.3 Gabriel

dd

10.1.4 Louis-Norman

dd

10.2 Difficultés et corrections

10.2.1 Vincent

ss

10.2.2 Hicham

ss

10.2.3 Gabriel

ss

10.2.4 Louis-Norman

ss

10.3 Ce que j'ai aimé ou pas

10.3.1 Vincent

a

10.3.2 Hicham

a

10.3.3 Gabriel

a

10.3.4 Louis-Norman

a

11 ANNEXE 1 : Code source du programme pour PC

```
using System;
```

12 ANNEXE 2 : Code source du Bolide et de la station 1

```
using System;
```

13 ANNEXE 4 : Code source de la table FESTO

```
using System;
```

14 ANNEXE 5 : Code source du programme PIC

```
void main( void )  
{  
}
```

15 ANNEXE 4 : Script Shell du SOC8200

prog.sh

```
compteur=0
```

read.sh

```
using System;
```

tcp.sh

```
using System;
```

can.sh

```
using System;
```

PortSerie.sh

```
while true
do
    head -1 /dev/ttysCMA0 > ./junk
    echo "Allo" > ./hbeat
done
```

RxCAN.sh

```
candump can0 >> ./histocan &
nbligne=1
ancienvar=0

while true
do
    var='tail -1 ./histocan'
    if [ "$var" != "$ancienvar" ]
    then
        echo $var
        echo $var >> ./histocandate
        date >> ./histocandate
        ancienvar=$var
    fi
# nbligne='wc -l ./histocan'
done
```

MachinChouette.sh

```
using System;
```

Bonus

*54 · 43. $\vdash .\alpha, \beta \in 1. \supset: \alpha \cap \beta = \Lambda. \equiv .\alpha \cup \beta \in 2$

Dem.

$$\begin{aligned}
 & \vdash . * 54 \cdot 26. \supset \vdash .\alpha = \iota'x.\beta = \iota'y. \supset: \alpha \cup \beta \in 2. \equiv .x \neq y. \\
 & [\ast 51 \cdot 231] \quad \equiv .\iota'x \cap \iota'y = \Lambda. \\
 & [\ast 13 \cdot 12] \quad \equiv .\alpha \cap \beta = \Lambda \tag{1} \\
 & \vdash .(1). * 11 \cdot 11 \cdot 35. \supset \\
 & \quad \vdash: .(\exists x, y).\alpha = \iota'x.\beta = \iota'y. \supset: \alpha \cup \beta \in 2. \equiv .\alpha \cap \beta = \Lambda \tag{2} \\
 & \vdash .(2). * 11 \cdot 54. * 52 \cdot 1. \supset \vdash .Prop
 \end{aligned}$$

From this proposition it will follow, when arithmetical addition has been defined, that $1 + 1 = 2$.

Cette démonstration mathématique prouve hors de tout doute que $1 + 1 = 2$.
Si les mathématiques sont vrais, alors notre projet devrait aller.