Hes·so VALAIS WALLIS
Haute Ecole Spécialisée de Suisse occidentale
Fachhochschule Westschweiz
University of Applied Sciences Western Switzerland

Filière Systèmes industriels

Orientation Infotronics

Diplôme 2009

Laurent Lugon Moulin

Swisscube integration, launch and operational activities

Professeur	Christophe Bianchi
Expert	Philippe Etique

Sion, le 6 juillet 2009

HES-SO Valais

SI	TV	EE	IG	EST
X	X			

Données du travail de diplôme
Daten der Diplomarbeit

FO 1.2.02.07.DB
mam/30/06/2008

☒ FSI ☐ FTV	Année académique / *Studienjahr* **2008/09**	No TD / *Nr. DA* **it/2009/17**
Mandant / *Auftraggeber* ☒ HES—SO Valais ☐ Industrie ☒ Etablissements partenaires **EPFL, IMT, HES-SO**	Etudiant / *Student* **Laurent Lugon Moulin**	Lieu d'exécution / *Ausführungsort* ☒ HES—SO Valais ☐ Industrie ☒ Etablissement partenaire
Professeur / *Dozent* **Christophe Bianchi**	Expert / *Experte* *(données complètes)*	
Travail confidentiel / *vertrauliche Arbeit* ☐ oui / *ja* ☒ non / *nein*		

Titre / *Titel*

Swisscube integration, launch and operational activities

Description et Objectifs / *Beschreibung und Ziele*

The SwissCube satellite (Flight Model 1) is approaching launch in april 2009 in India. The project is now starting to get organized to launch the second flight model (FM2) on the ESA-VEGA launch vehicle, launch currently planned for November 2009 from Kourou, French Guyana.

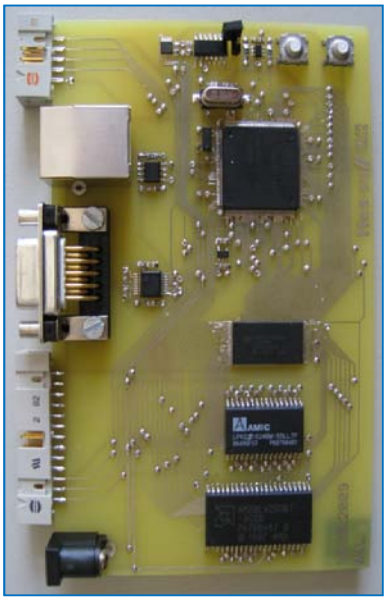
The main objectives of this bachelor project is:

1) to learn everything there is to know about the SwissCube satellite, and its respective subsystems
2) to analyse the operational data of the SwissCube FM1 in space in order to characterize the satellite performances and to detect the critical parts of the satellite (FMECA)
3) to propose a new architecture for the next generations of the satellite
4) to participate to the final integration of the FM2, prepare and perform the last tests before launch.

Signature ou visa / *Unterschrift oder Visum*	Délais / *Termine*
Resp. de la filière *Leiter des Studieng.:*	Attribution du thème / *Ausgabe des Auftrags:* 18.02.2009
Etudiant / *Student*:	Remise du rapport / *Abgabe des Schlussberichts:* 06.07.2009, 12:00
	Exposition publique / *Ausstellung Diplomarbeiten:* 04.09.2009
	Défense orale / *Mündliche Verfechtung:* Semaine / *Woche* 35

Hes·so VALAIS WALLIS
Haute Ecole Spécialisée de Suisse occidentale
Fachhochschule Westschweiz
University of Applied Sciences Western Switzerland

Swisscube integration, launch and operational activities



Diplômant/e Laurent Lugon Moulin

Travail de diplôme
| édition 2009 |

Filière
Systèmes industriels

Domaine d'application
infotronics

Professeur responsable
Christophe Bianchi
christophe.bianchi@hevs.ch

Partenaire
EPFL

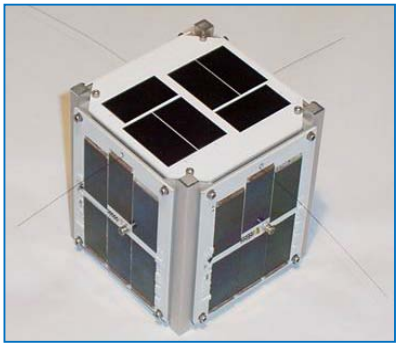
Objectif du projet

Le but de ce travail de diplôme est d'étudier et de proposer une nouvelle architecture de la carte CDMS, de réaliser et de tester une carte prototype de la nouvelle génération du sous-système CDMS du Swisscube.

Méthodes | Expériences | Résultats

Pour commencer, l'architecture de la carte CDMS a été étudiée afin de connaître les points à améliorer. Mme M. Noca, responsable de projet à l'EPFL, a également été consultée afin qu'elle définisse ses exigences pour la nouvelle carte. Avec ces informations, la nouvelle architecture a pu être dessinée.
Il a ensuite fallu choisir les composants correspondants à la nouvelle architecture. Afin de s'assurer d'utiliser le matériel le plus adéquat, les éléments de différents fabricants ont été choisi grâce à des tableaux comparatifs.
Lorsque l'architecture et les éléments à utiliser furent définis, la schématique a été réalisée grâce au logiciel ''P-CAD schematic''. Le PCB a ensuite été fabriqué par le bureau technique de l'Hes-so Valais et les composants ont été montés sur la carte.
Une fois la carte réalisée, il a fallu procéder à différents tests hardware (court-circuit, tension, consommation,…) et ensuite un programme de test a été chargé sur la carte afin de s'assurer du bon fonctionnement du microcontrôleur.

La carte de prototype est maintenant fabriquée avec les composants choisis pour la carte CDMS. Les tests hardware sont terminés et un programme de test a fonctionné sur la carte.



Le Swisscube est un CubeSat, ces types de picosatellites sont utilisés pour la recherche spatiale. Ils ont des dimensions de 10cm par côté et un poid maximal de 1kg.

Le P-Pod est le système standard de déployement pour les CubeSat. Il leur sert d'interface avec la fusée. Trois CubeSat sont chargés dans un P-Pod.

HES-SO Valais
Route du Rawyl 47
1950 Sion

Tél. 027 606 85 11
URL www.hevs.ch

CONTENTS

Contents **4**
TABLES **6**
FIGURES **7**
1 Introduction **8**
1.1 Standard 'CubeSat' definition 8
1.2 P-POD interface definition 8
1.3 Mission and science objectives 9
1.3.1 Mission objective 1 9
1.3.2 Mission objective 2 9
1.3.3 Mission objective 3 9
1.3.4 Science objective 9
1.3.5 Nightglow description 10
2 Diploma work definition **11**
2.1 General objective 11
2.2 Work organization 11
2.3 Task definition 11
2.3.1 CDMS architecture improvement 11
2.3.2 Choice of the new CDMS/prototype board components 11
2.3.3 Realization of the prototype board 11
2.3.4 Tests of the prototype board 11
3 Architecture **12**
3.1 Analysis of the current architecture 12
3.2 SPI/I^2C bridge replacement 12
3.3 Hardware redundancy 12
4 New parts choice **13**
4.1 Microcontroller 13
4.1.1 ATMEL 13
4.1.2 NXP 13
4.2 Memories 16
4.2.1 Memory architecture 16
4.2.2 Boot-loader storage 16
4.2.3 FM Software storage 16
4.2.4 Working memory 16
4.2.5 Data save 16
5 Prototype board block diagram **17**
6 Hardware development **19**
6.1 PICEBS1 board 19
6.1.1 Features 19
6.1.2 The Extension connectors 19
6.2 USB-to-CAN compact 21
6.3 JTAGkey 21
6.4 Operation modes 22
6.4.1 Standard mode 22
6.4.2 Shut down mode 22
6.4.3 Shut down and microcontroller idle mode 22
6.4.4 Microcontroller power-off mode 22
6.5 Logical level and supply voltage 22
6.6 Power-on reset 23
6.7 Schematic 23
6.7.1 Signal processing 23

6.7.2 Connectors 25
6.7.3 Microcontroller 27
6.7.4 Memories 31
6.7.5 External oscillator 33
6.7.6 Temperature Sensor 33
6.7.7 CAN transceiver 34
6.7.8 UART transceiver 34
6.8 PCB production 35
6.9 Global I/O list 35
6.9.1 Prototype board I/O 35
6.9.2 CDMS board I/O 36
7 Tests 37
7.1 Used devices 37
7.2 Hardware tests 37
7.2.1 Connections 37
7.2.2 Supply voltage 37
7.2.3 Power-on reset 38
7.2.4 Reset signal 38
7.2.5 External oscillator 38
7.2.6 Temperature Sensor 38
7.3 Software tests 39
7.3.1 Open On-Chip Debugger (openOCD) 39
7.3.2 Arm-elf-gcc 39
7.3.3 Eclipse 39
7.3.4 eCos 40
7.3.5 OpenOCD configuration file for AMEBS3 40
7.3.6 Loading an running a simple software on the ARMEBS3 board 41
7.3.7 OpenOCD configuration file for the prototype board 42
7.3.8 Loading an running a simple software on the prototype board 43
7.3.9 eCos porting on the prototype board 44
8 Prototype board to CDMS board adaptations 46
8.1 Decoupling capacitors 46
8.2 DC/DC converter 46
8.3 Temperature sensor placement 46
9 Conclusion 47
10 Future work 47
11 Abbreviations 48
12 References 49
13 Annexes 49
Annex 1: Forecast planning of diploma work 50
Annex 2: Hardware redundancy 52
Annex 3: Complete schematic of the prototype board 54
Annex 4: Bill of material 62

TABLES

Table 4-1 : ATMEL's microcontrollers 13
Table 4-2 : ARM7 NXP's microcontroller 14
Table 4-3 : ARM9 NXP's microcontroller 14
Table 4-4 : Comparison of the NXP ARM based microcontrollers 15
Table 4-5 : Result 15
Table 6-1 : Extension connector pinout 20
Table 6-2 : Binding array pinout 20
Table 6-3 : I^2C signals 21
Table 6-4 : operation modes 22
Table 6-5 : Voltage level 22
Table 6-6 : Commutation threshold of the Schmitt trigger 23
Table 6-7 : Reset list 25
Table 6-8 : First part of microcontroller signals 28
Table 6-9 : Second part of microcontroller signals 29
Table 6-10 : Control signals 32
Table 6-11 : SRAM mode selection 32
Table 6-12 : Capacitor value 33
Table 6-13 : UART signals 35
Table 6-14 : Prototype board I/O 35
Table 6-15 : CDMS board I/O 36
Table 7-1 : Used devices 37
Table 7-2 : supply voltage verification 37
Table 7-3 : Temperature sensor test 39
Table 7-4 : List of needed software and tools 44

FIGURES

Figure 1-1 : CubeSat in space 8
Figure 1-2 : P-POD interface 8
Figure 1-3 : NASA photo of the nightglow 10
Figure 3-1 : Current CDMS architecture 12
Figure 5-1 : Block diagram of the prototype board 18
Figure 6-1 : Extension connetor 19
Figure 6-2 : USB-to-CAN 21
Figure 6-3 : JTAGkey 21
Figure 6-4 : Current supply voltage 23
Figure 6-5 : New supply voltage 24
Figure 6-6 : Manual wakeup 24
Figure 6-7 : Reset part 25
Figure 6-8 : CAN connection 26
Figure 6-9 : JTAG connection 26
Figure 6-10 : I^2C connection 26
Figure 6-11 : RS232 connection 27
Figure 6-12 : Schematic of the first part of the microcontroller 27
Figure 6-13 : Schematic of the second part of the microcontroller 29
Figure 6-14 : Schematic of the third part of the microcontroller 30
Figure 6-15 : Position of the alimentation pin 30
Figure 6-16 : 8MB flash 31
Figure 6-17 : 2MB flash memory 31
Figure 6-18 : 8 bit bank external memory interface 31
Figure 6-19 : SRAM memory 32
Figure 6-20 : External oscillator 33
Figure 6-21 : Temperature sensor 33
Figure 6-22 : CAN transceiver 34
Figure 6-23 : UART transceiver 34
Figure 7-1 : power-on reset measurement 38
Figure 7-2 : OpenOCD result for the ARMEBS3 board 41
Figure 7-3 : Eclipse debug interface 42
Figure 7-4 : OpenOCD result for the prototype board 43
Figure 7-5: eCos configuration tool 44
Figure 7-6: eCos files for LPC2292 45
Figure 7-7: Compilation command 45
Figure 7-8: Compilation result 45

1 INTRODUCTION

Swisscube is the name of a Picosatellite which will be launched during the year 2009. He has the particularity to be entirely built by students of different Swiss university (EPFL, HES-SO, HE Arc) and respects the CubeSat standard.

The development of a space project is a complex and long task. This is why the construction of the various subsystems is divided into various groups. The Hes-so Valais task consists of the development of the subsystem named CDMS. The first board was elaborated during the phase B of the project, by the student Pierre André Tapparel Then, the student David Crettaz realized the second board during the phase C of the project. M. Christophe Bianchi, teacher at the HES-SO Sion was their supervisor.

Due to the distance between the different groups who work on the project, the communication between them is very important in order to share and discuss the problems, solutions and project changes. For that, meetings are regularly organized in various manners.

1.1 Standard 'CubeSat' definition

Figure 1-1 : CubeSat in space

CubeSat is a type of space research Picosatellite with dimensions of 10 cm each side and a maximum weight of one kilogram, and typically using commercial electronics components. Developed through joint efforts, California Polytechnic State University and Stanford University introduced the CubeSat to the world of academia as a means of opportunity for Universities throughout the world to enter into the realm of space science and exploration. The main advantage of this standard is to reduce the launch cost, because the standard allows 3 satellites to take place in a special box named P-POD.

1.2 P-POD interface definition

The P-POD is a standardized CubeSat deployment system. It is capable of carrying three standard CubeSat and serves as the interface between the CubeSat and the launch vehicle.

This rectangular box is made of aluminium with a door and a spring mechanism to permit the CubeSat ejection when the orbit height as been reached.

Figure 1-2 : P-POD interface

1.3 Mission and science objectives

The motivation for the overall SwissCube project development is primarily to educate students in space technologies and space system engineering. This motivation has several impacts:

- The project involves undergraduate and postgraduate students and young engineers through its whole life cycle.
- The project cost is relatively low, in accordance with a university type of development.
- Compared to an industry type space project, decisions are taken to simplify the design or design for low-cost and thus might not comply with the usual standards.

Keeping these aspects in mind, the mission and science objectives for the project are summarized in the following requirements.

1.3.1 Mission objective 1

The project shall design, build, and test a satellite. The success criterion is to deliver a fully tested satellite to the launch site.
This objective assumes the development of both a ground and space system.

1.3.2 Mission objective 2

The project shall launch the satellite and communicate with it using the ground and space systems. The success criterion is to establish a radio connection with the developed ground system and download telemetry.

1.3.3 Mission objective 3

The project shall operate a scientific or technology demonstration payload. The success criterion is to receive data from the payload and confirm operations.

1.3.4 Science objective

After discussions with several partners of the project, it was decided that the SwissCube mission should focus on the observation of the airglow phenomena. The motivation for these observations is to demonstrate the feasibility of using the airglow as basis for development of a low cost Earth Sensor (ES). A model of the airglow emissions as a function of intensity, latitude, longitude and time has been established and the objective the science mission is to collect data that will validate, or at least bring additional information to the model. The development of the Earth Sensor is a separate activity to SwissCube led by the EPFL-LMTS laboratory.

In addition, at the project level and as a technology demonstration, it was decided to develop a payload that has the most commonality/synergy as possible with the Earth Sensor.

1.3.5 Nightglow description

The nightglow is a photoluminescence of the atmosphere at night, occurring at approximately 100 km altitude (see Figure 1-3). It is principally due to the recombination of the atomic oxygen, which is dissociated during the day. To study variations of the emissions as a function of time, the minimum science duration is 3 months, with an extended science mission of duration up to 1 year.

Figure 1-3 : NASA photo of the nightglow

2 DIPLOMA WORK DEFINITION

2.1 General objective

In this diploma work, I have to study, realize and test a new CDMS board. This board will be a prototype of the FM3 CDMS board. In annex 1, there is a forecast planning of my diploma work.

2.2 Work organization

1. CDMS architecture improvement
2. Choice of the new CDMS/prototype board components
3. Realization of the prototype board
4. Test of the prototype board

2.3 Task definition

2.3.1 CDMS architecture improvement

Some points of the actual architecture have to be changed. I have to replace the SPI/I^2C bridge by a CAN bus. With this bus, we can simplify the actual architecture without a higher consumption. The memory architecture will also be changed.

Besides, after the programming of the FM1 CDMS board, it has been see that an UART interface will be helpful for the test/debug.

Finally, I have to add a hardware redundancy because the CDMS board is considered as a critical component.

2.3.2 Choice of the new CDMS/prototype board components

The microcontroller has to be changed to support the new peripherals. This microcontroller has to be supported by the OS (eCos). After test, the programming team has seen that they need higher capacity memories. Besides, the actual memory doesn't have driver for this OS. I have to check the driver for the new memories.

2.3.3 Realization of the prototype board

When the choice of the architecture and the component will be validate by the project manager, I should realize this board. This is made in three general parts:

1. Realization of the schematic
2. Realization of the PCB
3. Mounting the component

2.3.4 Tests of the prototype board

In the end, several tests must be done to check the board functionalities. First, hardware tests will check the eventual problem of the card (short-circuit, supply voltage, consumption…). Then simple software will be loaded on the board for checking the microcontroller behavior and finally a porting of the OS have to be done.

3 ARCHITECTURE

3.1 Analysis of the current architecture

In the figure 3-1, we can see the actual CDMS architecture. There is two principal points to improve:

1. The SPI / I^2C bridge will be replace by a CAN bus;
2. The CDMS board will have a hardware redundancy;

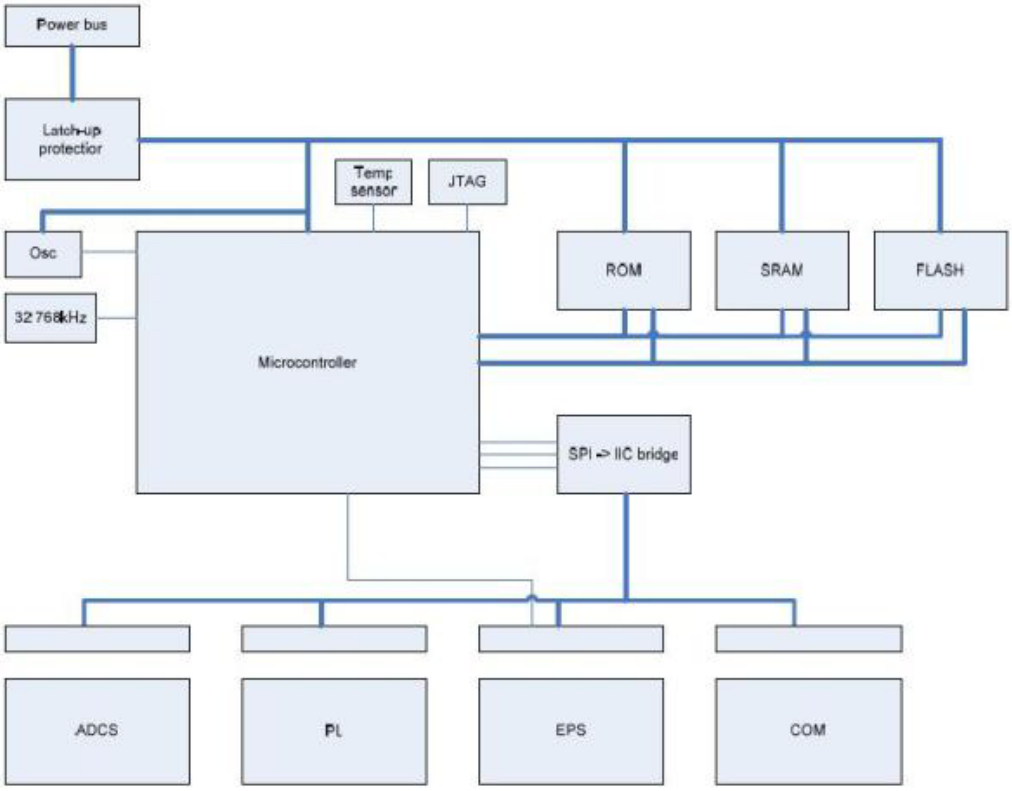


Figure 3-1 : Current CDMS architecture

3.2 SPI/I^2C bridge replacement

In the last diploma work, the I^2C has been chosen to assure the communication between the microcontroller and the subsystems. Unfortunately, the chosen microcontroller doesn't have an I^2C interface, so the SPI/I^2C bridge correct this problem.
In the way to simplify the architecture and keep a low consumption, this bridge will be replace by a direct CAN bus usage. This parameter is one of the most significant for the future microcontroller choice.

3.3 Hardware redundancy

The CDMS is a critical system, to protect the SwissCube against a failure of the CDMS we have to implement a hardware redundancy. A bloc diagram of the new CDMS architecture is represented on the annex 2.

This is a cold-redundancy of whole board. Only the subsystem interfaces are shared between the two microcontrollers. The EPS have to check that the microcontroller is working by sending a periodical message. If there is no response, the principal microcontroller is considering as down, and the supply microcontroller is using instead.

4 NEW PARTS CHOICE

4.1 Microcontroller

The microcontroller for the FM1 and FM2 is an AT91M55800A based on ARM7TDMI processor. There are three determinant parameters for the choice of the new microcontroller:

- CAN bus implementation
- External memory interface
- Spatial compatibility

For a better software recovery, the new microcontroller will be ARM processor based. I have search in two of the most important microcontroller provider.

- ATMEL
- NXP (Philips)

4.1.1 ATMEL

ATMEL provide a parametric product table of their microcontrollers. In the table 4-1, we can see a part of the ATMEL's table with the three determinant parameters. There are only microcontrollers who have CAN interface. The only one who have a external memory interface is the AT91SAM9263, but its package is not spatial compatible (BGA package).

Device	External Bus Interface	CAN	Packages	Green Packages
AT91SAM9263	2	1	TFBGA 324	TFBGA 324
AT91SAM7X128	--	1	LQFP 100	LQFP 100 TFBGA 100
AT91SAM7X256	--	1	LQFP 100	LQFP 100 TFBGA 100
AT91SAM7XC128	--	1	LQFP 100	LQFP 100 TFBGA 100
AT91SAM7XC256	--	1	LQFP 100	LQFP 100 TFBGA 100
AT91SAM7A3	--	2	LQFP 100	LQFP 100
AT91SAM7X512	--	1	LQFP 100	LQFP 100 TFBGA 100
AT91SAM7XC512	--	1	LQFP 100	LQFP 100 TFBGA 100

Table 4-1 : ATMEL's microcontrollers

There is no microcontroller which can be used from ATMEL.

4.1.2 NXP

NXP provide an online part selection guide. In table 4-2, we can see the NXP's microcontroller based on ARM7 who has an external memory controller. The four last have a CAN implementation but the 2292FET doesn't have a spatial compatible package. The three other are very similar, so the first one will be study more in details.
In table 4-3, we can see the NXP's microcontroller based on ARM9 who has an external memory controller. There are four with CAN implementation but the LPC2927 and the LPC2929 still not in production. The two other are very similar, so the first one will be study more in details.

LPC2210FBD144	SOT486-1	2xUART, 1xSPI,1xSSP, 1xI2C	Ext. Mem. Ctrl.
LPC2212FBD144	SOT486-1	2xUART, 1xSPI,1xSSP, 1xI2C	Ext. Mem. Ctrl.
LPC2214FBD144	SOT486-1	2xUART, 1xSPI,1xSSP, 1xI2C	Ext. Mem. Ctrl.
LPC2220FBD144	SOT486-1	2xUART, 1xSPI,1xSSP, 1xI2C	Ext. Mem. Ctrl.
LPC2220FET144	SOT569-2	2xUART, 1xSPI,1xSSP, 1xI2C	Ext. Mem. Ctrl.
LPC2290FBD144	SOT486-1	2xCAN,2xUART, 1xSPI,1xSSP, 1xI2C	Ext. Mem. Ctrl.
LPC2292FBD144	SOT486-1	2xCAN,2xUART, 1xSPI,1xSSP, 1xI2C	Ext. Mem. Ctrl.
LPC2292FET144	SOT569-2	2xCAN,2xUART, 1xSPI,1xSSP, 1xI2C	Ext. Mem. Ctrl.
LPC2294HBD144	SOT486-1	4xCAN,2xUART, 1xSPI,1xSSP, 1xI2C	Ext. Mem. Ctrl.

Table 4-2 : ARM7 NXP's microcontroller

LPC2917FBD144	SOT486-1	2xCAN,2xLIN, 2xUART,3xQSPI	16KB ITCM/DTCM, Ext.Mem.Ctrl.
LPC2919FBD144	SOT486-1	2xCAN,2xLIN, 2xUART,3xQSPI	16KB ITCM/DTCM, Ext.Mem.Ctrl.
LPC2927FBD144	SOT486-1	USB 2.0 FS O/D, 2xCAN,2xLIN/UART, 2xUART,3xQSPI, 2xI2C	QEI, GPDMA, 16KB EEPROM, 32KB ITCM/DTCM, Ext.Mem.Ctrl.
LPC2929FBD144	SOT486-1	USB 2.0 FS O/D, 2xCAN,2xLIN/UART, 2xUART,3xQSPI, 2xI2C	QEI, GPDMA, 16KB EEPROM, 32KB ITCM/DTCM, Ext.Mem.Ctrl.

Table 4-3 : ARM9 NXP's microcontroller

In Table 4-4 you can see the comparison between the two families (ARM7 and ARM9) and finally in table 4-5 the result.

The chosen microcontroller is the **NXF LCP2292FBD144**.

Parameters	peak power comsumption	Standby Power consumption	Temperature range	Power voltage core	Power voltage I/O	I/O compatibility	Max operating frequeny	Digital I/O number	Analog I/O number	USART	OS	Standard timer number	multimedia instructions
Needed	as little as possible	as little as possible	-40 to +85	-	-	5V tolerant	-	12	8	1	eCos compatibility	-	if possible
Unit	-	-	°C	V	V	-	MHz	-	-	-	-	-	-
AT91M558800A (used in FM1 & FM2)	6.55mW/MHz	1.78mW/MHz	-40 to +85	3.3	3.3	5V tolerant	33	58	8	3 USART	ARM7 supported	2*16bit	no
ARM7 based family	90mW (peripherals not active)	18µW (at 1.8V 25°C)	-40 to +85	1.8	3.3	5V tolerant	60	76	8*10bit	2 USART	ARM7 supported	2*32bit	no
ARM9 based family	158.4mW (peripherals active)	54µW (at 1.8V 25°C)	-40 to +85	1.8	3.3	5V tolerant	80	108	16*10bit	2 USART	ARM9 supported	4*32bit	no
Comments	at max frequency, 1.8V, 25°C	they can respond to a wake-up event	-	-	nominal voltage	-	-	-	-	-	-	-	useful for the image processing

Table 4-4 : Comparison of the NXP ARM based microcontrollers

Parameters	Peak power comsumption	Standby Power consumption	Temperature range	Power voltage core	Power voltage I/O	I/O compatibility	Max operating frequency	Digital I/O number	Analog I/O number	USART	OS	Timer number	multimedia instructions	Result
weighting factor (1-4)	4	3	2	3	4	3	1	3	3	2	4	1	3	
AT91M558800A (used in FM1 & FM2)	1	1	3	1	3	3	1	3	3	3	3	2	1	79
ARM7 family	3	3	3	3	3	3	2	2	3	2	3	2	1	95
ARM9 family	2	2	3	3	3	3	3	1	1	2	3	3	1	81

Table 4-5 : Result

4.2 Memories

4.2.1 Memory architecture

There are 4 principal usages of the memories:

1. Boot-loader storage
2. FM software storage
3. Working memory
4. Data save

The memory architecture has been done in order to respect these functionalities.
The external memory controller of the chosen microcontroller supports SRAM, ROM, flash EPROM and burst ROM memories. It is separate in 4 independently configurable memory banks each of 16Mb capacity.

4.2.2 Boot-loader storage

For the boot-loader, we need a non-volatile memory. This memory will be read at the start of the microcontroller in order to begin the principal software. The chosen microcontroller has an internal flash (256Kb) which can be used for the boot-loader.
Besides, a little soft for erase the flash memory (§4.2.3) will be also store in this memory.

4.2.3 FM Software storage

The FM software will be stored in a non-volatile memory. The microcontroller will load this software and run it. Another flash memory will be used for this application.
If the FM software is modified during the mission, they can be uploading to the SwissCube. First, we need to erase the flash memory with the software stored in microcontroller internal memory (§4.2.2). Then we can upload the new FM software part by part.
The density of this flash have to be at least as big as the working memory density which is 1Mb (§4.2.4). . eCos provide a list of supported hardware which defines the compatible flash memories. There is no 1Mb flash memory which is compatible. So a 2Mb flash memory from AMD Spansion has been chosen (Am29LV200B).

4.2.4 Working memory

We need a volatile memory used as working memory, for the stack and for saving temporary data. The LCP2292FBD144 provide only 16Kb of internal SRAM which is not enough. In accordance with the FM software development team request, a 1Mb SRAM will be used. The chosen memory is the AMIC LP 62S1024BM-55LLTF)

4.2.5 Data save

The picture take by the payload and the housekeeping have to be saved in a non-volatile memory. We need a density of 4Mb for this job. For this memory, the same type as the FM2 flash memory has been chosen in order to simplify the development. There is no 4Mb flash memory from AMD Spansion compatible with eCos, so an 8Mb flash memory has been chosen (AM29LV081B).

5 PROTOTYPE BOARD BLOCK DIAGRAM

The Block diagram of the prototype board is shown in the figure 5-1. A laboratory power supply is using for the alimentation. For the programming, the JTAG bus is used but the UART is also implemented for debug facilities.

The CAN bus is used for the communication with the subsystem (for the prototype board, the CAN communications are simulated with a USB-to-CAN module, see §6.2). The I^2C is also implemented in order to test the prototype board first with the actual software (which works with the I^2C bus). The I^2C communications are tested with the PICEBS1 board (see §6.1).

The two external signals (reset and wakeup) which are controlled by the EPS in the CDMS board are here simulated with two switches.

On the right are the three kinds of memory (flash 2MB, flash8MB and SRAM 1MB). They are controlled by the external memory controller of the LPC2292.

The microcontroller drives the shut-down signal (SHDN) which can turn off the temperature sensor and the UART transceiver. The goal of this signal in the prototype board is to test the shut-down operation modes.

Finally in the middle on the right is the 12MHz quartz. This frequency has been chosen in order to be high enough for an eventual PLL usage (minimum 10MHz).

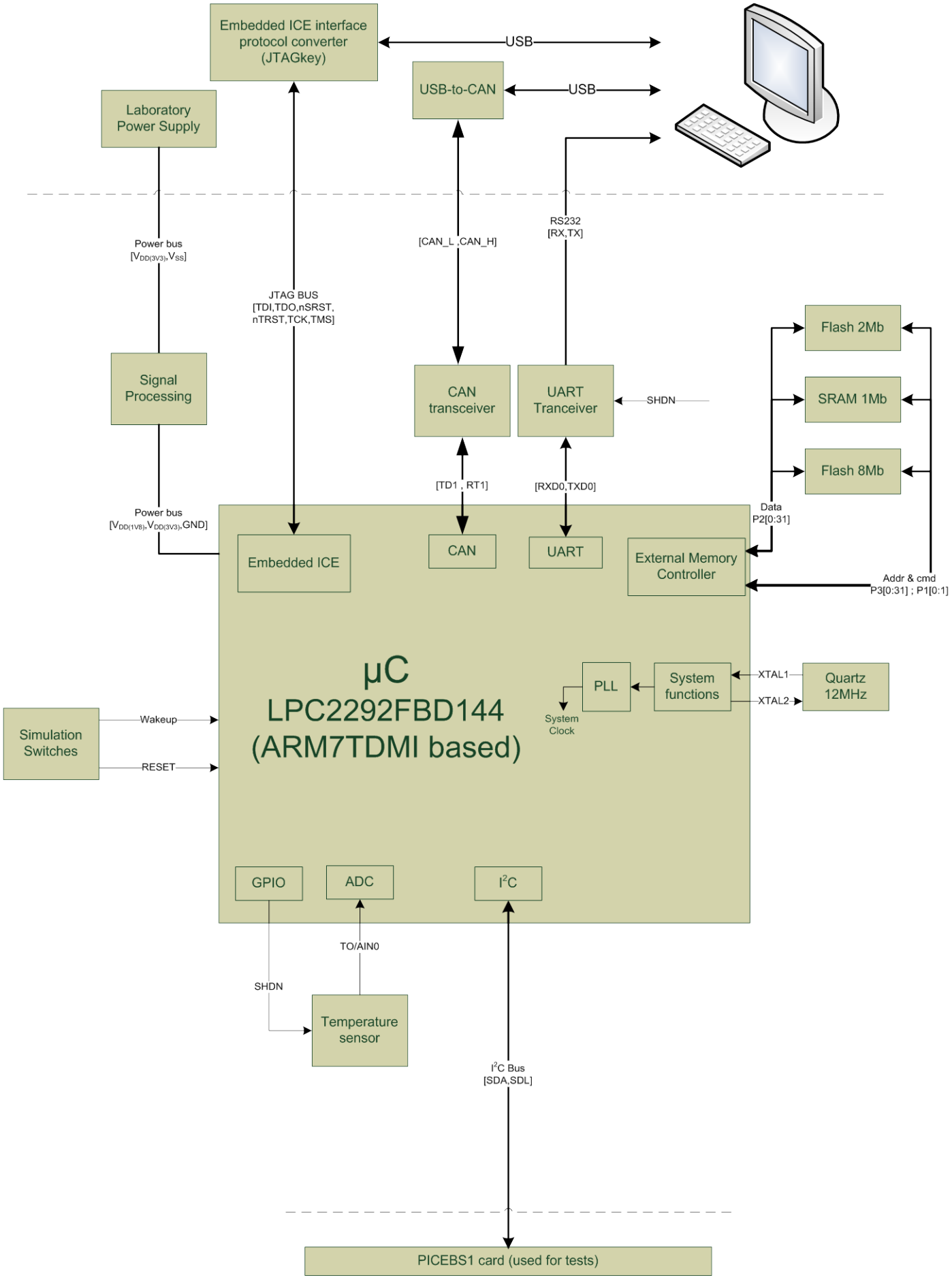


Figure 5-1 : Block diagram of the prototype board

6 HARDWARE DEVELOPMENT

This prototype of the new generation of CDMS board will be used to prove the good behavior of all the chosen components and the software. The general components for the CDMS board (capacitors, resistors…) are from SMD 806 type. In order to simplify the board, all the components are in SMD technology (excepted the switches and the connectors). In annex 4, there is a bill of all the material used on the prototype board.

6.1 PICEBS1 board

The PICEBS1 is an experimentation board with different peripherals. It will be used for testing the I^2C interface bus of the prototype board.

6.1.1 Features

- PIC18LF6680 at 25MHz, 3.3V (32kWord instruction, 3328 bytes RAM, 1024 bytes EEPROM)
- 12 A/D 10 bits
- SPI/I2C
- UART
- 4 Timers
- Enhanced CAN
- Connectivity :
  - 1 x Serial RS-232
  - 2 x CAN (1.2, 2.0A, 2.0B compatible)
  - 1 x SD/MMC memory board slot
  - 1 x Nunchuck (Nintendo) slot
  - 5 x Buttons and 4 x user LEDs
  - 2 x Extension (17 - IO individually connectable (quasi any CPU pin))

6.1.2 The Extension connectors

On the right of the PICEBS1 board are two extension connectors. They are useful to integrate any new system on this board or to use the already existing extension modules. This extension connector will be used for testing the I^2C bus.

Figure 6-1 : Extension connetor

The user can bind any I/O extension pin with most of the microcontroller pin. The table below describes the extension pinout. Some of the extension pin are fixed by the Hes-so Valais standard.

Pin number	Description	Pin number	Description
1	P1	2	P2
3	P3	4	P4
5	P5	6	P6
7	P7	8	P8
9	P9	10	GND
11	P11	12	GND
13	P13	14	GND
15	P15	16	GND
17	P17	18	GND
19	P19	20	GND
21	P21	22	GND
23	P23	24	GND
25	P25	26	VCC (3.3V)

Table 6-1 : Extension connector pinout

The table below describes the binding array corresponding to the extension connector (right row of binding array) and the processor available I/O (left and center rows of binding array).

Pin number	*Row left (CPU)*	*Row center (CPU)*	*Row right (extension)*
1	RC2/PWMA	RD0	P1
2	RC5/SDO	RD1	P2
3	RC3/SCK	RD2	P3
4	RC4/DSI	RD3	P4
5	n.c.	RD4	P5
6	RB2/INT2	RD5	P6
7	RG1	RD6	P7
8	RE6/PWMB	RD7	P8
9	RG3	RB3/INT3	P9
10	RA5/AN4	RE5/PWMC	P11
11	RF7/SS	RA0/AN0	P13
12	RF6/AN11	RA1/AN1	P15
13	RF5/AN10	RA2/AN2/REF-	P17
14	RF4/AN9	RA3/AN3/REF+	P19
15	RF3/AN8	RA4/T0CKI	P21
16	RF2/AN7	3.3V	P23
17	RF1/AN6	RG4/PWMD	P25
18	RF0/AN5	5V	GND

Table 6-2 : Binding array pinout

The I^2C signal are connected to the pin RC3 and RC4 of the PIC (see table 6-3).

RC3/SCK/SCL	34	45	44			
RC3				I/O	ST	Digital I/O.
SCK				I/O	ST	Synchronous serial clock input/output for SPI mode.
SCL				I/O	ST	Synchronous serial clock input/output for I^2C mode.
RC4/SDI/SDA	35	46	45			
RC4				I/O	ST	Digital I/O.
SDI				I	ST	SPI data in.
SDA				I/O	ST	I^2C data I/O.

Table 6-3 : I^2C signals

So the signal RC3 and RC4 of the row left (from CPU) have to be connected in any pin of the row right (extension connector) in this case with the pin 3 (RC3) and 4 (RC4).

6.2 USB-to-CAN compact

The USB-to-CAN compact is a low-cost, active CAN module for connection to the USB bus. The 16-bit microcontroller system enables reliable, loss-free transmission and reception of messages in CAN networks with both a high transmission rate and a high bus load. In addition, messages are provided with a time-stamp and can be filtered and buffered directly in the USB-to-CAN compact. This module will be used to test the CAN interface of the prototype board.

Figure 6-2 : USB-to-CAN

6.3 JTAGkey

The Amontec JTAGkey is an advanced hardware debug devices that enable software to load, control, debug and test a target system.

Figure 6-3 : JTAGkey

6.4 Operation modes

The four possible modes are short described below. In table 6-4 are the respective typical and maximal global consumptions for each mode.

Mode	Typical consumption	Maximal consumption
Standard	40mA	100mA
Shut down	20mA	35mA
Shut down and idle	15mA	20mA
Power-off	10mA	20mA

Table 6-4 : operation modes

6.4.1 Standard mode

This is the standard operation mode. All the components are in their normal utilization mode.

6.4.2 Shut down mode

All the components are in standby mode and the microcontroller generate the SHDN signal.

6.4.3 Shut down and microcontroller idle mode

All the components are in standby mode and the microcontroller is in idle mode.

6.4.4 Microcontroller power-off mode

The microcontroller is in power-off mode. In this mode, the microcontroller can't drive its output, so the other components are in their standard operation mode (no shut down signal). The microcontroller needs an external wakeup signal (interrupt signal) to terminate its power-off mode.

6.5 Logical level and supply voltage

For each logical component, there is in table 6-5 the maximum and minimum voltage check for both high and low level. The supply voltage tolerance is also checked below.

Component	High level voltage [V]	Low level voltage [V]	V_{CC} [V]
Microcontroller	+2.0 to +5.5	Lower than +0.4	+1.65 to +1.95
Flash 8Mb	+2.31 to +3.6	-0.5 to +0.8	+2.7 to +3.6
Flash 2Mb	+2.31 to +3.6	-0.5 to +0.8	+2.7 to +3.6
SRAM 1MB	+2.2 to +3.6	-0.3 to +0.6	+2.7 to +3.6
Temperature sensor	+2.31 to +3.8	-0.5 to +0.99	+2.7 to +5.5
CAN transceiver	+2.0 to +7.5	Lower than +0.8	+3.0 to +3.6
3 input AND Gate	Higher than +2V	Lower than +0.8	+1.65 to +5.5
UART transceiver	Higher than +2V	Lower than +0.8	+3.0 to +3.6

Table 6-5 : Voltage level

We can use +3.3V for the logical high level voltage and V_{SS} (0V) for the logical low level voltage. The supply voltage will be +3.3V except for the microcontroller (+1.8V). A DC/DC converter is needed to provide the +1.8V. For this job, a MSP1080 from Microchip has been chosen because of its very low current consumption (typical25µA).

6.6 Power-on reset

At the starting of the microcontroller, the reset signal has to still low until the supply voltage has raise to its nominal value. First, the raising time of the supply voltage have to be measured. For the prototype board, a laboratory power supply (TTi EL302Tv) is used. The longest of ten raising time measurements is 18ms. The reset has to be done during minimum this 18ms at the power-on of the microcontroller.
To hold the reset signal low, a low-pass filter and two inverter Schmitt triggers are used. In the table below, there are the commutation thresholds of the Schmitt trigger.

Parameter	Description	Worst case	Typical case	Unit
V_{T+}	Positive-going threshold	1.32	2.2	V
V_{T-}	Negative-going threshold	1.98	1.3	V

Table 6-6 : Commutation threshold of the Schmitt trigger

In order to assure the reset time, the low-pass filter will be calculated with the worst case positive-going threshold of the trigger. The τ of the RC filter can easily be calculated, and then with the triggers, we can toggle the output value at the good moment. With a capacitor of 1µF and a resistor of 18K (normalized value), the reset time of 18ms is assured.

6.7 Schematic

The schematic of the prototype board is detailed here. The complete schematic is in annex 3.

6.7.1 Signal processing

6.7.1.1 Supply voltage

The current supply voltage is in figure 6-4. The microcontroller controls the alimentation of the entire CDMS component. By clearing the SHDN signal, the supply voltage is cut excepted for the microcontroller backup power. When it sets the SHDN signal, the components are supply again.

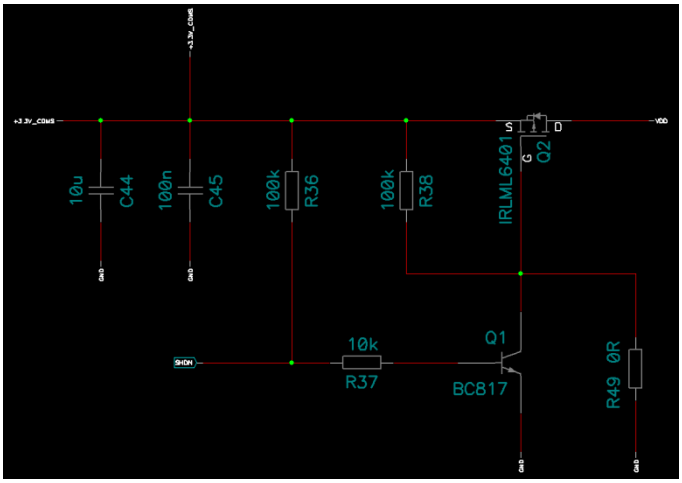


Figure 6-4 : Current supply voltage

In the new version, the power supply isn't drive by the microcontroller because it doesn't have a backup power. For save current, the microcontroller can go to power-down mode (with software instruction). In this mode, the current consumption is reducing to nearly zero. Any of the other chips have also a power-down mode controlled by the microcontroller. A DC/DC converter produces the 1.8V which is used for the power supply of the

microcontroller (the output capacitor is specified by the documentation of the chip). The new schematic is on the figure below.

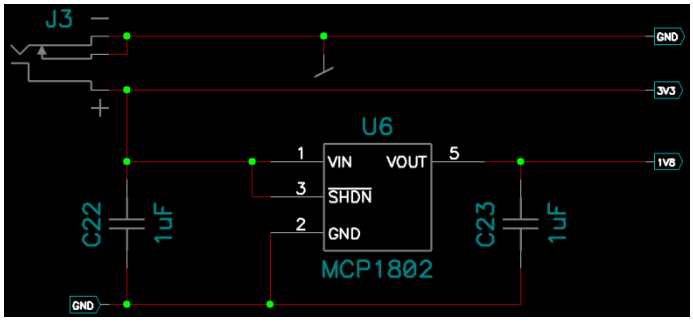


Figure 6-5 : New supply voltage

6.7.1.2 Manual wakeup

The wakeup signal is simulated with a simple push button and a pull-down resistor. The wakeup is an active high signal unlike the older microcontroller wakeup signal. The manual wakeup is connected to an external interrupt pin of the microcontroller. With this signal, the microcontroller terminated its power-down mode.

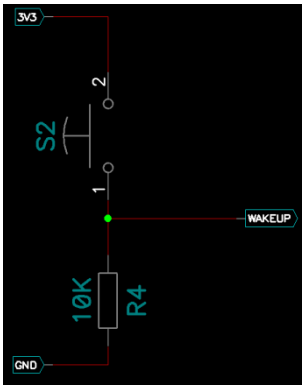


Figure 6-6 : Manual wakeup

6.7.1.3 Reset part

In the current version, a MAX823 reset IC is used to manage both the manual reset and the watchdog. The other reset source comes from the JTAG. Besides, the power-on reset is directly connected to a specific input of the microcontroller.

For the new version, there is no specific input for the power-on reset, so the three sources are handling together with the OR gate (U9). With the new microcontroller, the watchdog is an internal signal, so the MAX823 is useless. The new schematic is shown in the figure 6-7. The list of reset is shown in the table 6-7.

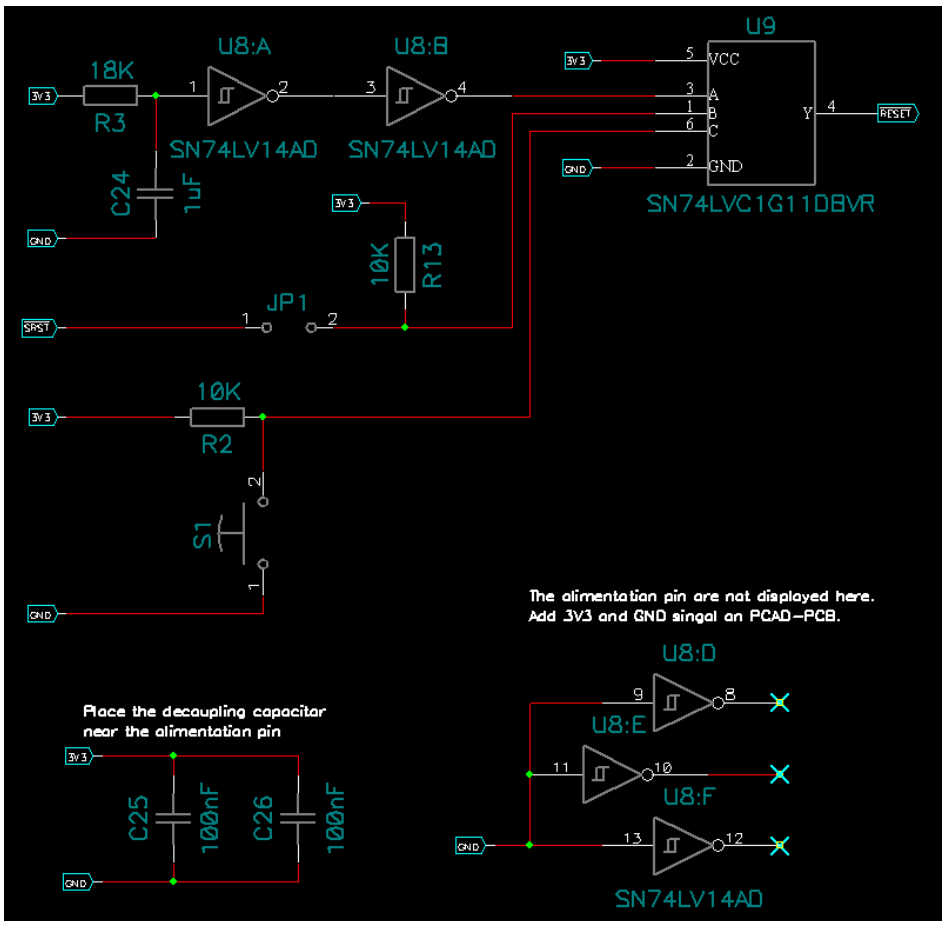


Figure 6-7 : Reset part

OR Gate entry	Reset type
A	Power-on reset (§6.6)
B	JTAG reset (§6.7.2.2)
C	Manual reset

Table 6-7 : Reset list

The JTAG system reset is connected through a jumper in order to test if this reset is useful or not for the new CDMS board.

6.7.2 Connectors

6.7.2.1 CAN bus

The physical layer of the CAN bus use the RJ45 connector. The same pin as the USB-to-CAN module have been chosen for the tests. The connection is shown in the figure 6-8..

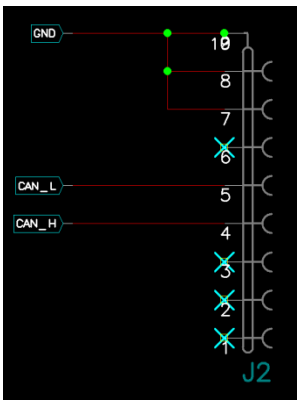


Figure 6-8 : CAN connection

6.7.2.2 JTAG Debug/Test interface

The JTAG connection has be done in order to use the Amontec JTAGkey interface protocol converter. It is the same as the actual connections (figure 6-9). RTCK has to be at low level at the power-on reset in order to use the debug port.

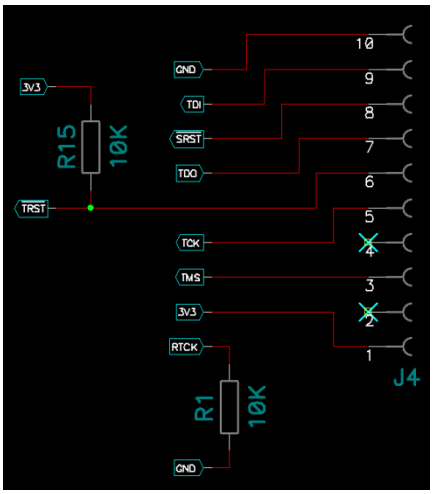


Figure 6-9 : JTAG connection

6.7.2.3 I²C interface

The connector for the I²C is compatible with the PICEBS extension connector (§6.1.2). The connection is shown in the figure 6-10.

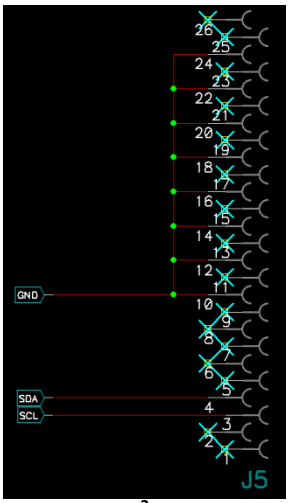


Figure 6-10 : I²C connection

6.7.2.4 RS232 UART interface

The UART use a RS232 connector. This connector can be directly connected with a PC (DCE equipement). The connection is shown in the figure below.

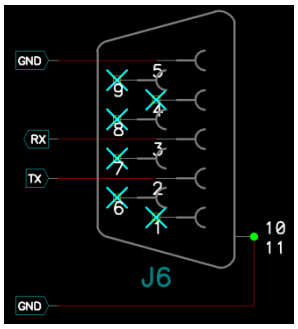


Figure 6-11 : RS232 connection

6.7.3 Microcontroller

For the schematic, the microcontroller has been separated in three blocks.

6.7.3.1 Peripherals block

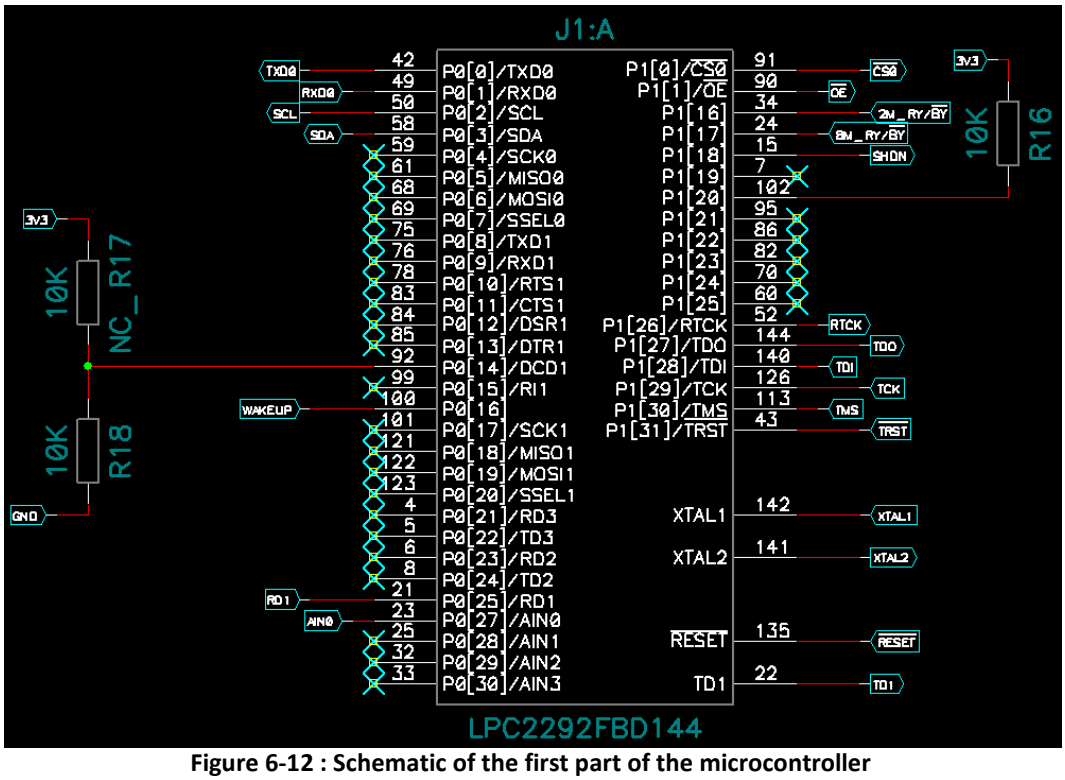


Figure 6-12 : Schematic of the first part of the microcontroller

This block's principal usage is to manage the peripheral signals. The table 6-8 defines the signals and their usage.

Name	Type	I/O	Description	Periph. Ref.
TXD0	-	O	CMOS/TTL compatible UART transmission signal	§6.7.2.4
RXD0	-	I	CMOS/TTL compatible UART reception signal	§6.7.2.4
SCL	-	O	I^2C clock	§6.7.2.3
SDA	-	I/O	I^2C data	§6.7.2.3
P0[14]	L	I	Selection of on-chip boot loader	-
WAKEUP	H	I	Wakeup signal	§6.7.1.2
RD1	-	I	Received data from CAN transceiver	§6.7.2.1
AIN0	-	I	Temperature signal	§6.7.6
CS0	L	O	Chip select 0 (flash 8MB)	§6.7.4.1
OE	L	O	Output enable	§6.7.4
2M_RY/BY	-	I	Ready busy signal from 2MB flash	§6.7.4.1
8M_RY/BY	-	I	Ready busy signal from 8MB flash	§6.7.4.1
SHDN	H	O	Shut-down signal	-
P1[20]	-	I	Trace port selection. (pull high for standard port P1 utilization)	-
RTCK	-	O	Real time clock (pull down for JTAG utilization)	§6.7.2.2
TDO	-	O	JTAG data output	§6.7.2.2
TDI	-	I	JTAG data input	§6.7.2.2
TCK	-	I	JTAG clock	§6.7.2.2
TMS	-	I	Test mode selected for JTAG	§6.7.2.2
TRST	L	I	JTAG test reset	§6.7.2.2
XTAL1	-	I	External oscillator input frequency	§6.7.5
XTAL2	-	O	External oscillator excitation	§6.7.5
RESET	L	I	General reset	§6.7.1.3
TD1	-	O	Transmitted data for CAN transceiver	§6.7.2.1

Table 6-8 : First part of microcontroller signals

H = active High, L = active Low

6.7.3.2 Address and data block

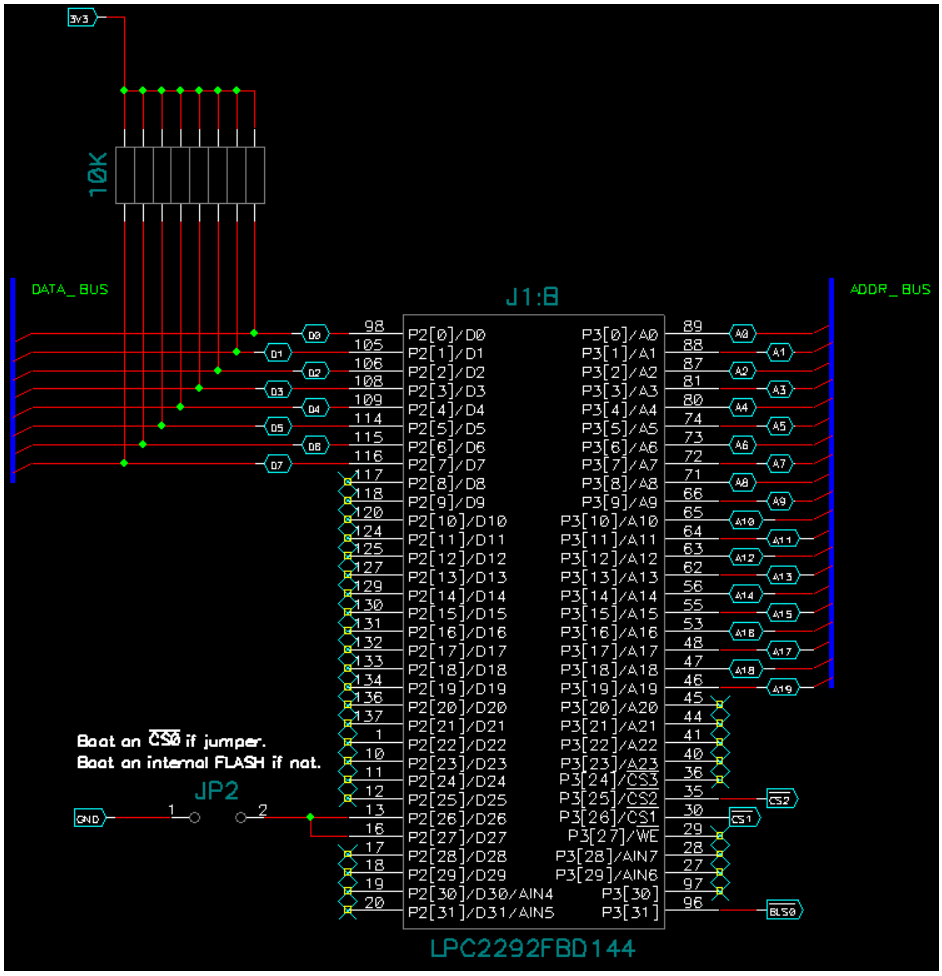


Figure 6-13 : Schematic of the second part of the microcontroller

This block is principally made of the data bus (8bit) with pull-up resistors and the address bus (20bit). The table below defines the signals and their usage.

Name	Type	I/O	Description	Periph. Ref.
DATA_BUS	-	I/O	Data bus	-
ADDR_BUS	-	O	Address bus	-
P2[26,27]	-	I	Boot select	-
CS1,CS2	L	O	Chip select for memories	§6.7.4
BLS0	L	O	Used as write enable for memories	§6.7.4

Table 6-9 : Second part of microcontroller signals

H = active High, L = active Low

6.7.3.3 Alimentation block

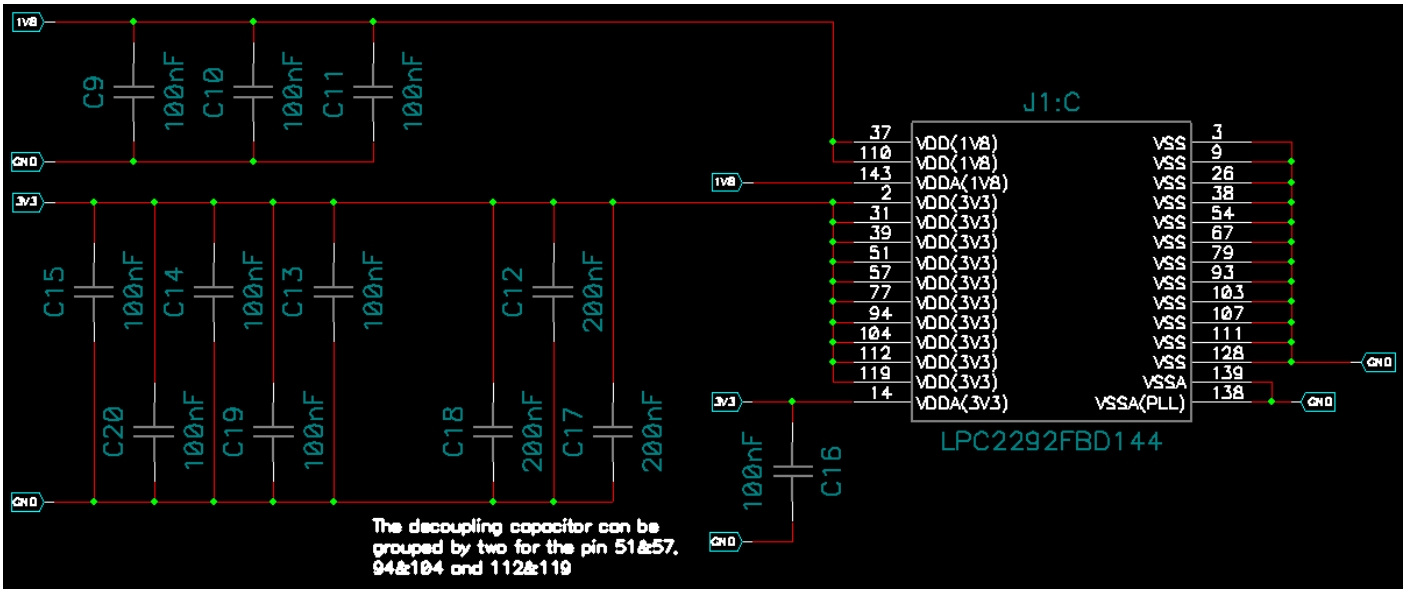


Figure 6-14 : Schematic of the third part of the microcontroller

This last block is made of the alimentations and the grounds of the microcontroller. There are two alimentation voltages (1.8V and 3.3V). V_{DDA} (1.8 and 3.3V) and V_{SSA} are the analogical reference for the ADC. These alimentations must be connected in only one point in order to minimize the noise and reduce the error. A filter can be added to improve the precision of the reference voltage but it is not needed here because the ADC is only used for the temperature measurement.

The capacitors C_9 to C_{20} are decoupling capacitors and have to be placed near the alimentation pin. The figure below shows the alimentation pin position around the microcontroller. When two pin are close enough and have the same voltage, one capacitor with double density can be used instead of two.

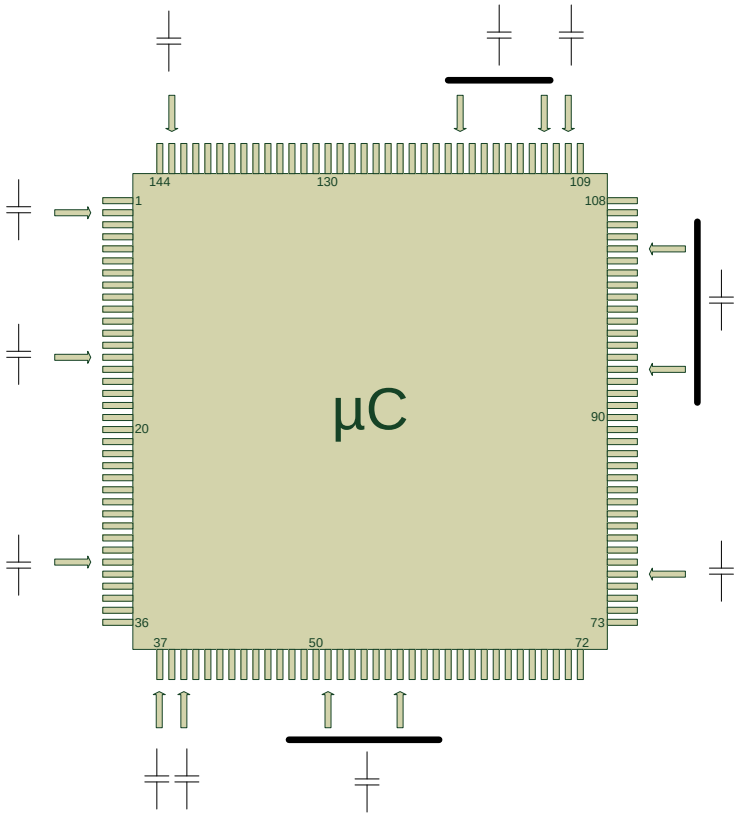


Figure 6-15 : Position of the alimentation pin

Six pin have been grouped by two (51 & 57, 94 & 104 and 112 & 119). So the capacitors C_{12}, C_{17} and C_{18} have the double density (200nF), the other decoupling capacitors have a density of 100nF.

6.7.4 Memories

6.7.4.1 8Mb and 2Mb flash memories

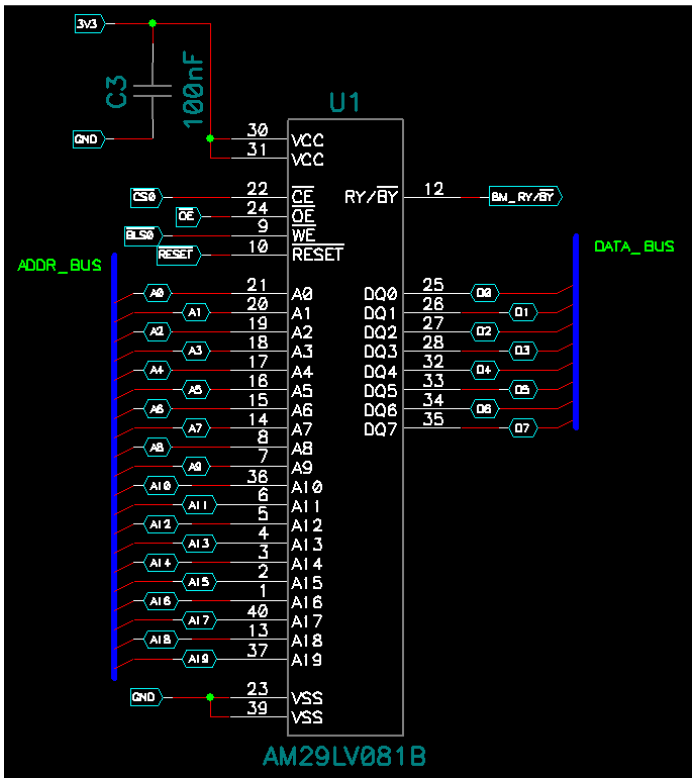


Figure 6-16 : 8MB flash

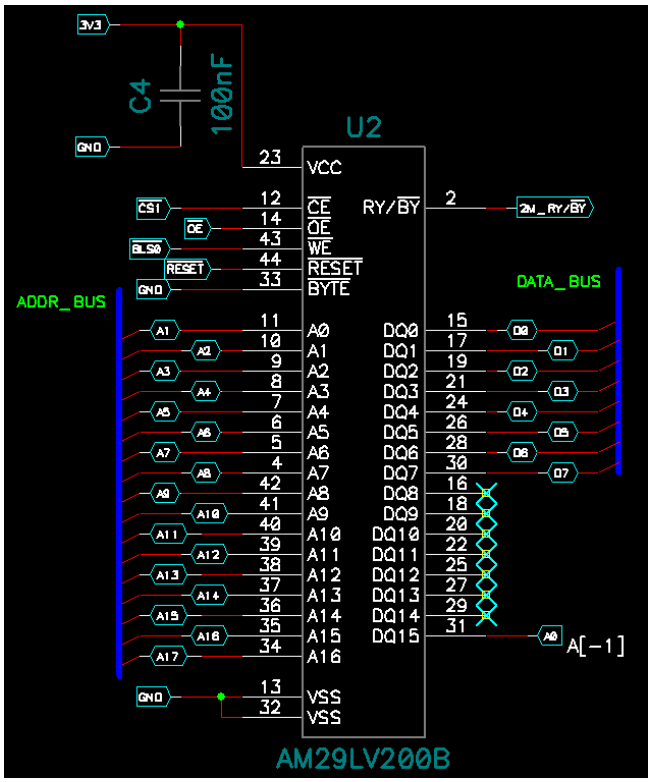


Figure 6-17 : 2MB flash memory

The connection of the memory is shown in the documentation of the microcontroller (figure 6-18).

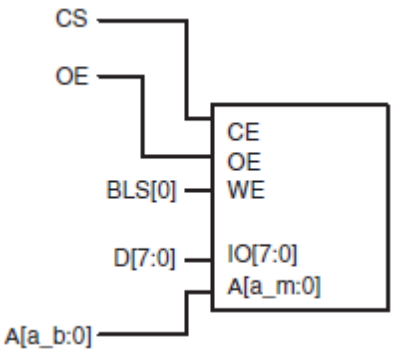


Figure 6-18 : 8 bit bank external memory interface

For both memories, there is an 8bit data bus and an address bus. For the 8Mb flash, the address bus (20bit) is connected from A_0 to A_{19}. For the 2Mb flash, the mode have to be selected with the BYTE pin. In order to have the same size as the others memories, the chosen mode is 8bit (byte mode). The 2Mb flash memory need one more address bit in order to work in byte mode. This bit is taken on the DQ15 pin and used as A_{-1}. V_{CC} are 3.3V alimentation pin, decoupling with C_3 and C_4. In table 6-10, there is the description of each control signal.

Name	Type	I/O	Description
CE	L	I	Chip enable
OE	L	I	Output enable
BLS0	L	I	Write enable
RESET	L	I	Reset signal
BYTE	L	I	Byte mode selection
RY/BY	-	O	Memory's status (1 = ready, 0 = busy)

Table 6-10 : Control signals

6.7.4.2 1Mb SRAM

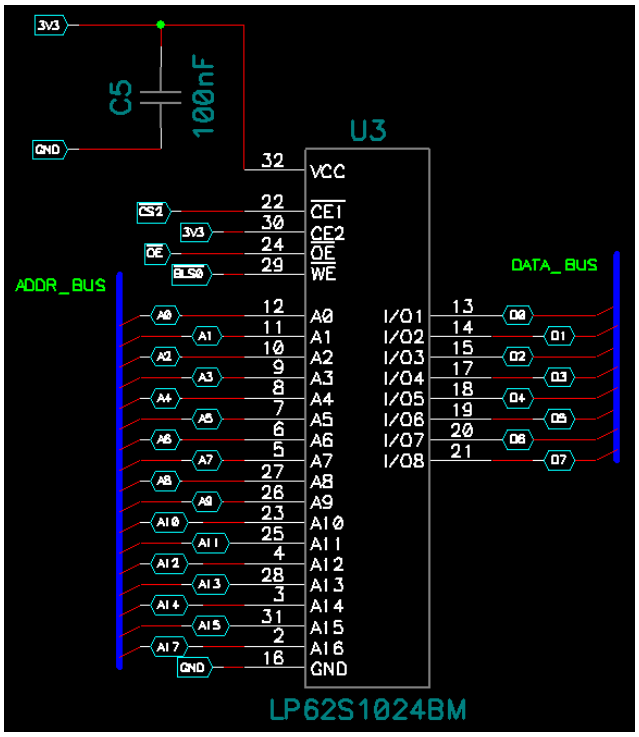


Figure 6-19 : SRAM memory

For the SRAM memory, there is just one more signal (CE2, active high). This signal is used for select the Standby mode if it is as low level (Table 6-11), for this assembly, the standby mode will be controlled by setting an high level on $\overline{CE1}$. So the CE2 pin is always at high level.

Mode	$\overline{CE1}$	CE2	$\overline{OE}$	$\overline{WE}$	I/O Operation	Supply Current
Standby	H	X	X	X	High Z	I_{SB}, I_{SB1}
	X	L	X	X	High Z	I_{SB}, I_{SB1}
Output Disable	L	H	H	H	High Z	I_{CC}, I_{CC1}, I_{CC2}
Read	L	H	L	H	D_{OUT}	I_{CC}, I_{CC1}, I_{CC2}
Write	L	H	X	L	D_{IN}	I_{CC}, I_{CC1}, I_{CC2}

Table 6-11 : SRAM mode selection

6.7.5 External oscillator

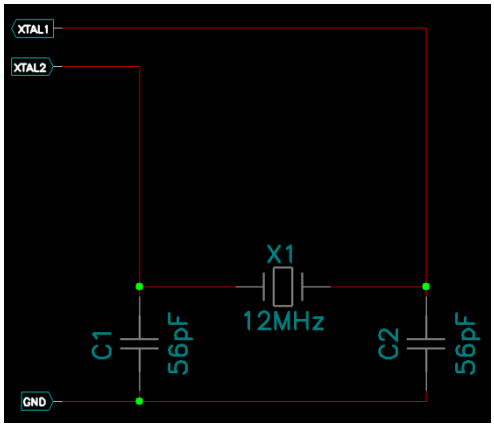


Figure 6-20 : External oscillator

The new external oscillator has a frequency of 12MHz. With this frequency, the PLL can be used if it is needed. The oscillator is handled by the system control block of the LPC2292. The value of the two external load capacitors can be chosen with the table 6-12 from the LPC2292 documentation. The crystal has a load capacitance of 12-32 pF with a maximum series resistance of 50Ω, so it needs external load capacitors of 58pF (56pF normalized).

Fundamental oscillation frequency F_{OSC}	Crystal load capacitance C_L	Maximum crystal series resistance R_S	External load capacitors $C_{X1,\ CX2}$
1 MHz - 5 MHz	10 pF	NA	NA
	20 pF	NA	NA
	30 pF	< 300 Ω	58 pF, 58 pF
5 MHz - 10 MHz	10 pF	< 300 Ω	18 pF, 18 pF
	20 pF	< 300 Ω	38 pF, 38 pF
	30 pF	< 300 Ω	58 pF, 58 pF
10 MHz - 15 MHz	10 pF	< 300 Ω	18 pF, 18 pF
	20 pF	< 220 Ω	38 pF, 38 pF
	30 pF	< 140 Ω	58 pF, 58 pF
15 MHz - 20 MHz	10 pF	< 220 Ω	18 pF, 18 pF
	20 pF	< 140 Ω	38 pF, 38 pF
	30 pF	< 80 Ω	58 pF, 58 pF
20 MHz - 25 MHz	10 pF	< 160 Ω	18 pF, 18 pF
	20 pF	< 90 Ω	38 pF, 38 pF
	30 pF	< 50 Ω	58 pF, 58 pF
25 MHz - 30 MHz	10 pF	< 130 Ω	18 pF, 18 pF
	20 pF	< 50 Ω	38 pF, 38 pF
	30 pF	NA	NA

Table 6-12 : Capacitor value

The new microcontroller doesn't need a 32KHz external frequency for starting unlike the older one.

6.7.6 Temperature Sensor

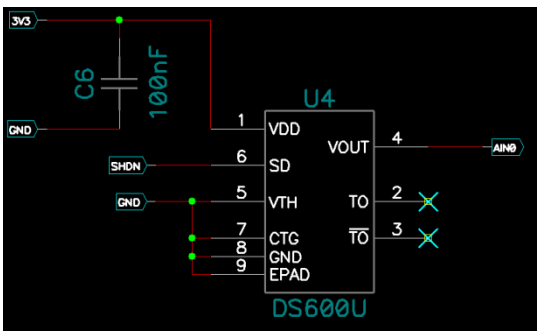


Figure 6-21 : Temperature sensor

SD put the temperature sensor in shut-down mode if it is set (controlled by the microcontroller with the SHDN signal). The V_{OUT} signal gives an analogical voltage which represents the temperature (6.45mV/°C, offset of 509mA at 0°C). V_{TH} fix the thermostat trip-point temperature which active TO and $\overline{TO}$, but these signals aren't used.

6.7.7 CAN transceiver

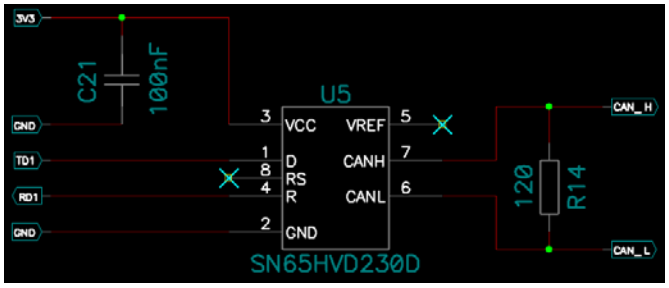


Figure 6-22 : CAN transceiver

The transceiver used in the practice is the SN65HVD232D. The pin 5 and 8 of this chip aren't connected. The signals TD1 and RD1 come from the microcontroller, they are the signal to transmit on the CAN bus (TD1) and the signal received from the CAN bus (RD1). The Transceiver creates these signals from CANH and CANL. The resistor R_{14} has to have a minimum power dissipation of 80mW.

$P = \frac{U^2}{R} = \frac{3.1^2}{120}$ (3.1V is the maximal voltage difference between CANH and CANL)

6.7.8 UART transceiver

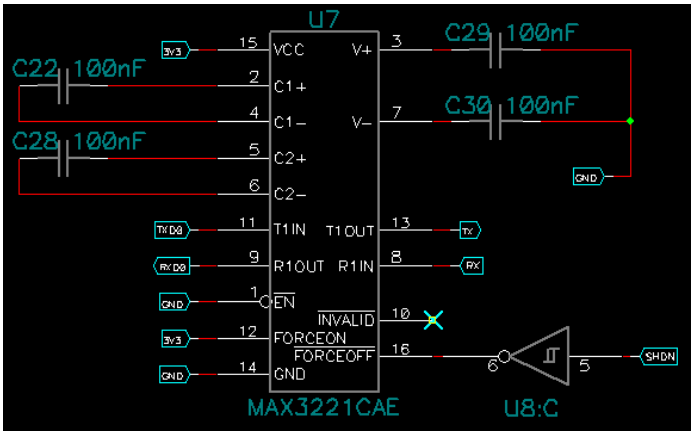


Figure 6-23 : UART transceiver

This chip transforms the signal of the UART with CMOS/TTL voltage level to signal with RS232 voltage level. The values of the capacitors are fixed in the documentation of the MAX3221. Its signals are describe in the table below.

Name	Type	I/O	Description
T1IN	-	I	Transmission signal input from microcontroller
R1OUT	-	O	Received signal output to microcontroller
OEN	L	I	Output enable
FORCEON	H	I	Force the chip to be in standard mode
V+	-	O	Positive output supply voltage
V-	-	O	Negative output supply voltage
T1OUT	-	O	Transmission signal output to RS232
R1IN	-	I	Received signal input from RS232
INVALID	L	O	Input invalid (if voltage on RS232 is too low)
FORCEOFF	L	I	Force the chip to be in sleep mode

Table 6-13 : UART signals

6.8 PCB production

Based on this schematic, the technical department of the Hes-so has made the PCB. In order to simplify their work, no size restriction has been given. After that, I have mounted all the components on the board.

Finally, the regrouping of the decoupling capacitors will not be done because there is enough space on the prototype board.

6.9 Global I/O list

The goal of this section is to list all the I/Os that are needed for the new generation CDMS board. In order to do that, the first step is to list the I/Os that are in the prototype board and to define which are used for the final CDMS board and which will be suppress (table 6-14). Then the I/Os for the CDMS board are listed in table 6-15.

6.9.1 Prototype board I/O

Name	I/O	Description	Needed in the CDMS board
$V_{DD(3V3)}$	I	Power supply	X
V_{SS}	I	Ground	X
CAN_H	I/O	CAN high	X
CAN_L	I/O	CAN low	X
2M_RY/BY	O	Ready/Busy signal from the 2MB flash memory	-
8M_RY/BY	O	Ready/Busy signal from the 8MB flash memory	-
SDA	I/O	I^2C data	-
SCL	O	I^2C clock	-
RESET	O	Reset signal of the prototype board	X
TDI	I	JTAG data input	X
TDO	O	JTAG data output	X
TCK	I	JTAG test clock	X
TMS	I	JTAG Test mode select	X
SRST	I/O	JTAG System reset	X
TRST	I	JTAG Test reset	X
RX	I	UART received data	-
TX	O	UART transmitted data	-

Table 6-14 : Prototype board I/O

6.9.2 CDMS board I/O

2M_RY/BY and 8M_RY/BY are on output only in order to verify the good behavior of the memories. The reset is on output to control that the logic of this signal is correct. These three signals plus the I^2C signals and the UART signals are not needed in the CDMS board.

Name	I/O	Description
$V_{DD(3V3)1}$	I	Power supply for the first microcontroller
V_{SS1}	I	Ground
$V_{DD(3V3)2}$	I	Power supply for the second microcontroller
V_{SS2}	I	Ground
CAN_H	I/O	CAN high
CAN_L	I/O	CAN low
RESET1	I	Reset signal of the first microcontroller
RESET2	I	Reset signal of the second microcontroller
WAKEUP1	I	Wakeup signal of the first microcontroller
WAKEUP2	I	Wakeup signal of the second microcontroller
TDI	I	JTAG data input
TDO	O	JTAG data output
TCK	I	JTAG test clock
TMS	I	JTAG Test mode select
SRST	I/O	JTAG System reset
TRST	I	JTAG Test reset

Table 6-15 : CDMS board I/O

The signals RESET1, RESET2 and WAKEUP1, WAKEUP2 are coming from the EPS (see annex 1).

7 TESTS

At this point, the board is realized and the soft is ready for the tests. The goal yet is to check that the prototype board is ready to be used with the final FM soft.

7.1 Used devices

Material	Type
Multimeter	Agilent U1252A
Oscilloscope	Agilent 54622D
Thermocouple probe	-
Industrial heat gun	-

Table 7-1 : Used devices

7.2 Hardware tests

7.2.1 Connections

The connection lines of the PCB are simply checked visually. After that, a multimeter in beep mode is used for checking the switches and to verify that there is no short-circuit between 3V3 and GND, and between 1V8 and GND.

Result:

- The connection lines of the PCB are alright.
- The switches work correctly.
- There is no short-circuiting.

7.2.2 Supply voltage

The supply voltage and the signal processing are checked with a multimeter for each component. The two voltage have to be measured (3V3 and 1V8). Then the current have to be measured to verify if there is a too high consumption.

Result:

In the table below is the result of the supply voltage measurement.

Component name	Value [V]	Needed [V]	Correct
DC/DC converter	3.3	3.3	OK
AND gate	3.3	3.3	OK
Microcontroller	3.3	3.3	OK
Microcontroller	1.8	1.8	OK
Flash 2MB	3.3	3.3	OK
Flash 8MB	3.3	3.3	OK
SRAM 1MB	3.3	3.3	OK
Temperature sensor	3.3	3.3	OK
UART transceiver	3.3	3.3	OK
CAN transceiver	3.3	3.3	OK

Table 7-2 : supply voltage verification

Then the current of the prototype board has been measured in the standard mode. The result is 24.5mA.

7.2.3 Power-on reset

The power-on reset is checked with an oscilloscope. The duration of the reset is measured in order to be sure that it covers the starting time of the microcontroller (18ms).

Result:

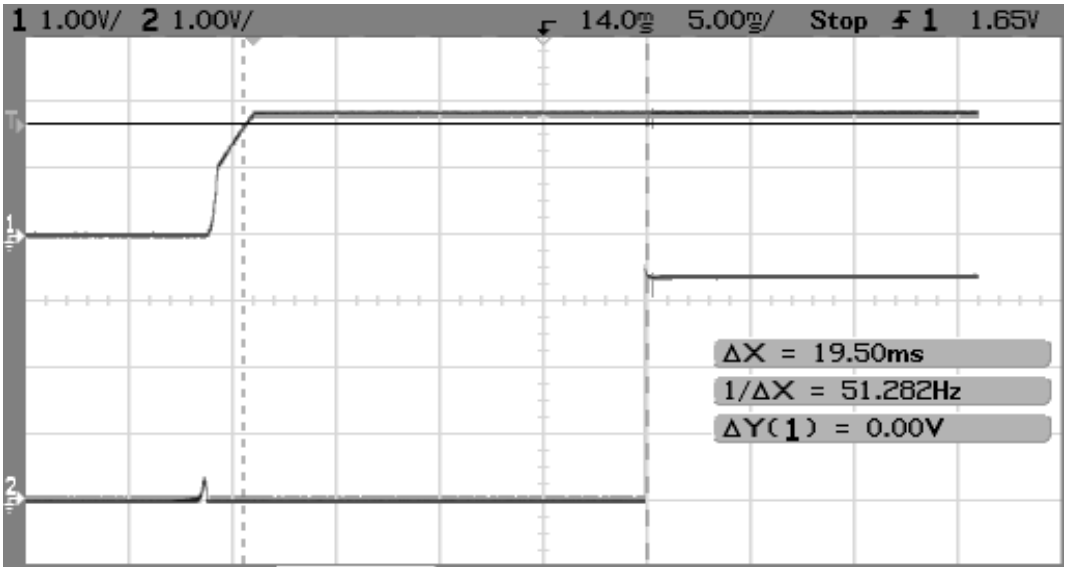


Figure 7-1 : power-on reset measurement

At the power-on of the microcontroller (minimum supply voltage = 1.65V), the signal reset is down during 19.5ms (see figure 7-1, the microcontroller alimentation signal is on the top of the figure, the reset signal is on the bottom). That's longer that the microcontroller power-on time.

7.2.4 Reset signal

The reset signal is checked after the AND gate with a multimeter.

Result:
The reset signal responds correctly.

7.2.5 External oscillator

The oscillation frequency is checked with an oscilloscope. The normalized frequency is 12MHz.

Result:
The oscillation frequency on the XTAL1 pin of the microcontroller is 12MHz. That is correct.

7.2.6 Temperature Sensor

The temperature sensor output values are checked with a multimeter. The measure is doing with different temperatures. Normally, the temperature sensor should have an output offset of 509mV at 0°C and a gain of 6.45mV/°C.

Result:

Temperature [°C]	Output voltage [mV]	Gain [mV/°C]	Correct
22	280	11.57	No
50	-	-	-
80	-	-	-

Table 7-3 : Temperature sensor test

Since the first measurement, the gain of the temperature sensor is not good. The connection has been checked one more time and the supply voltage too. The problem comes not from that. Then I have measured all the chip signal voltages. On the pin shut-down (SD), the voltage was 2.3V. This voltage is neither a high level nor a low level voltage output of the microcontroller.

After discussion with a Hes-so professor, this instable state is due to the fact, that the microcontroller is not initialized yet. And so the 2.3V on the shut-sown pin implicate a random output of the temperature sensor. The test of the temperature sensor will be done again after the loading of the OS.

7.3 Software tests

The goal of the software tests is to assure that all the peripherals are ready to be used and that the OS works alright. The used debugger is openOCD which is describe more in detail at §7.3.1. The compilator is arm-elf-gcc (§7.3.2) and Eclipse is used for the programming (§7.3.3).

First, the JTAGkey is tested with an ARMEBS3 board in order to understand in details the configuration file of openOCD and to be sure that the environment is ready to work. When a program is running successfully on the ARMEBS3 board, the second step is to modify the openOCD configuration file in order to use it for the prototype board. Then a porting of eCos will be done.

7.3.1 Open On-Chip Debugger (openOCD)

OpenOCD is an open-source on-chip debug solution for targets based on the ARM7, ARM9, Cortex-M3 and XSCALE families with Embedded-ICE support via JTAG port. It enables source level debugging with the standard GNU Debugger GDB compiled for the ARM architecture.

In addition internal and external FLASH memory programmings are supported. Any GDB aware integrated development environment, example Eclipse IDE, IAR or Emacs, can benefit from OpenOCD.

7.3.2 Arm-elf-gcc

The arm-elf-gcc is a C cross compiler for ARM targets using the ELF file format.

7.3.3 Eclipse

Eclipse is a multi-language software development platform comprising an IDE and a plug-in system to extend it. It is written primarily in Java and can be used to develop applications in Java and, by means of the various plug-ins, in other languages as well, including C, C++, COBOL, Python, Perl, PHP, and others. For the test, the CDT (C/C++ Development Tools) plug-in was installed in order to work in C/C++.

7.3.4 eCos

eCos is open-source real-time embedded operating system ported to a variety of architecture. It gives developers a low-cost, royalty-free embedded software development solution that works in highly constrained hardware environments.

7.3.5 OpenOCD configuration file for AMEBS3

For the tests, the configuration file done by the programming team of He-arc was recovered. Unfortunately, it was unusable because the syntax of openOCD has change and the configuration file was written in the old syntax.

On yagarto.de, there is a tutorial about how to set up a GNU ARM toolchain which explains shortly how to write an openOCD configuration file. With this tutorial and the openOCD user's guide (also found on yagarto.de) the configuration file for the ARMEBS3 board with JTAGkey can be define:

```
#Daemon configuration
#------------------------------------------------------------------------------
telnet_port 4444
gdb_port 3333
tcl_port 6666


#interface definition
#------------------------------------------------------------------------------
interface ft2232
ft2232_device_desc "Amontec JTAGkey A"
ft2232_layout jtagkey
ft2232_vid_pid 0x0403 0xcff8


#JTAG configuration
#------------------------------------------------------------------------------
jtag_khz 400
jtag_nsrst_delay 100
jtag_ntrst_delay 100
reset_config trst_and_srst srst_pulls_trst


#Tap creation
#jtag newtap [CHIPNAME] [TAPNAME] [OPTIONS...]
#------------------------------------------------------------------------------
jtag newtap AT91 cpu -ircapture 0x01 -irlen 4 -irmask 0x0f -expected-id 0x05b0203f


#Target creation
#target create [NAME] [TYPE] [OPTIONS...]
#------------------------------------------------------------------------------
target create AT91.cpu arm920t -endian little -chain-position AT91.cpu
targets


#Work area definition
#External 128MB flash @ 0x2000'0000
#------------------------------------------------------------------------------
AT91.cpu configure -work-area-virt 0x20000000
AT91.cpu configure -work-area-phys 0x20000000
AT91.cpu configure -work-area-size 0x08000000
AT91.cpu configure -work-area-backup 1


#Use the chip DCC
#------------------------------------------------------------------------------
arm7_9 dcc_downloads enable

init
reset
```

When openOCD is running with this configuration file, it found the interface (JTAGkey) and the target (AT91, ARMEBS3 CPU) as we can see in the figure below. So this configuration file is correct for the ARMEBS3 board.

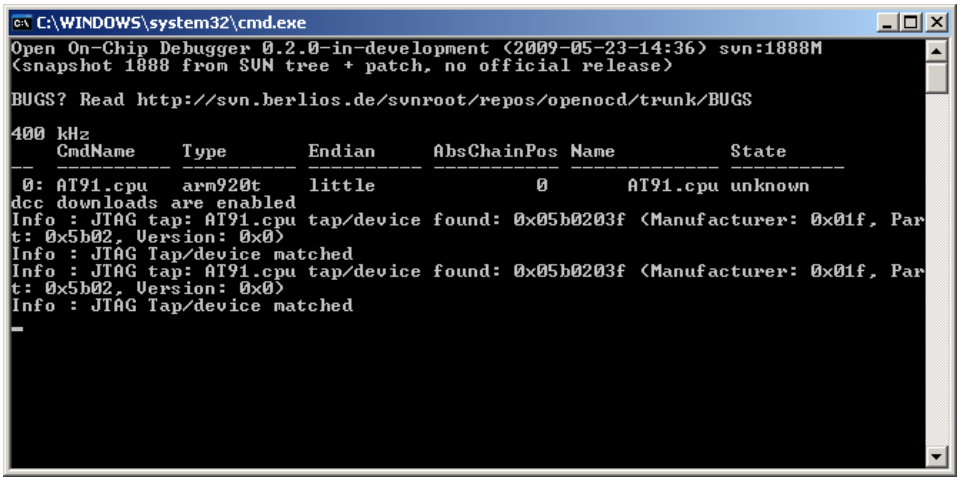


Figure 7-2 : OpenOCD result for the ARMEBS3 board

7.3.6 Loading an running a simple software on the ARMEBS3 board

Now this simple C software has been written with eclipse in order to try to run something on the ARMEBS3 board through the JTAGkey interface:

```
static const DWORD d = 7;

int main (void)
{
  DWORD a = 1;
  DWORD b = 2;
  DWORD c = 0;

  a = a + d;

  while (1)
  {
    a++;
    b++;
    c = a + b;
  }

  return(0);
}
```

The configuration of eclipse has been changed in order to use arm-elf-gcc for the compiling instead of the default compiler. Then the software is running with the debug interface of eclipse (see figure 7-3). On the top of the window is the variable value, on the middle is the main.c and on the bottom is the console. The result of this test assures that the programming environment (debugger, compiler, development software and interface) is functional.

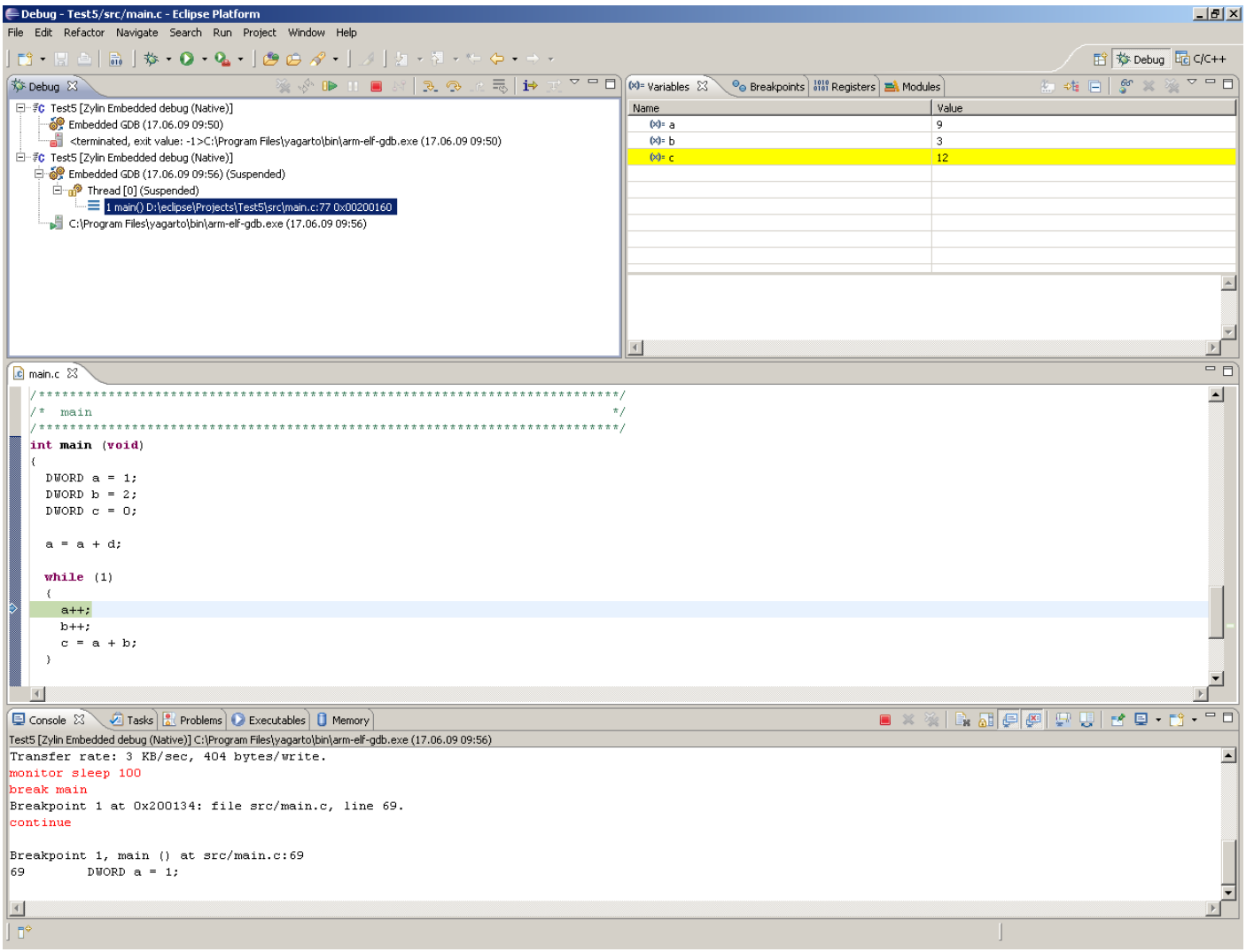


Figure 7-3 : Eclipse debug interface

7.3.7 OpenOCD configuration file for the prototype board

The configuration file for the prototype board is similar as the ARMEBS3 configuration file. The only difference is the target. The new target is an LPC2292 instead of an AT91RM9200. There is the new configuration file:

```
#Daemon configuration
#------------------------------------------------------------------------------
telnet_port 4444
gdb_port 3333
tcl_port 6666


#interface definition
#------------------------------------------------------------------------------
interface ft2232
ft2232_device_desc "Amontec JTAGkey A"
ft2232_layout jtagkey
ft2232_vid_pid 0x0403 0xcff8


#JTAG configuration
#------------------------------------------------------------------------------
jtag_khz 400
jtag_nsrst_delay 100
jtag_ntrst_delay 100
reset_config trst_and_srst srst_pulls_trst
```

```
#Tap creation
#jtag newtap [CHIPNAME] [TAPNAME] [OPTIONS...]
#-------------------------------------------------------------------------------
jtag newtap LPC2292 cpu -ircapture 0x01 -irlen 4 -irmask 0x0f -expected-id
0x4f1f0f0f

#Target creation
#target create [NAME] [TYPE] [OPTIONS...]
#-------------------------------------------------------------------------------
target create LPC2292.cpu arm7tdmi -endian little -chain-position LPC2292.cpu
targets


#Work area definition
# LPC2292 has a 16KB block of sram @ 0x4000'0000
#-------------------------------------------------------------------------------
LPC2292.cpu configure -work-area-virt 0x40000000
LPC2292.cpu configure -work-area-phys 0x40000000
LPC2292.cpu configure -work-area-size 0x00004000
LPC2292.cpu configure -work-area-backup 1

init
reset
```

When openOCD is running with this configuration file, it found the interface (JTAGkey) and the target (LPC2292, prototype board CPU) as we can see in the figure below. So this configuration file is correct for the prototype board.

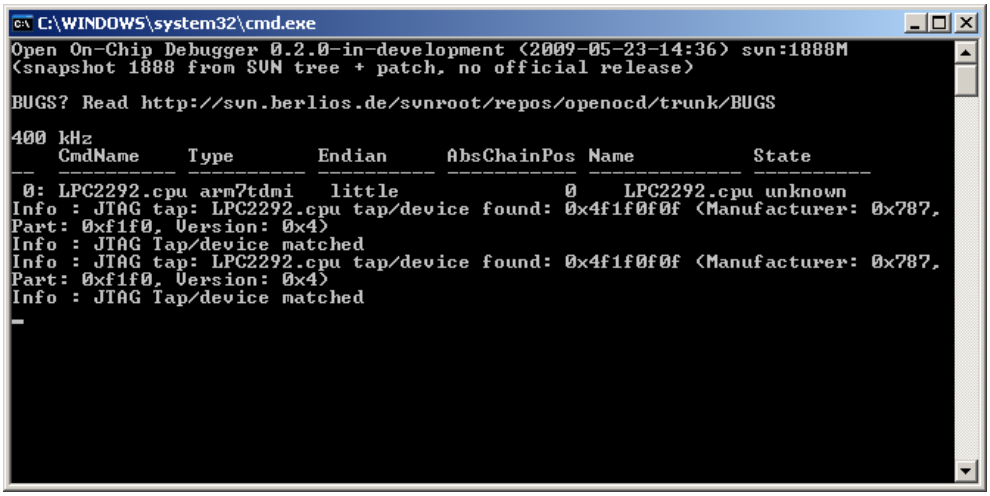


Figure 7-4 : OpenOCD result for the prototype board

7.3.8 Loading an running a simple software on the prototype board

The same simple software as the one in §7.3.6 has been loaded in the prototype baord using eclipse. The software is running properly. The result of this test assures that the programming environment is functional with the prototype board.

7.3.9 eCos porting on the prototype board

The first step is to download all the software and/or tools needed for running eCos. They are listed in the table below:

Name	Short Description
Cygwin	Linux-like work environment for Window
Pre-built toolchain	Pre-built toolchain for ARM7TDMI-based board; used for eCos
GNU binary utilities	Collection of programming tools for the manipulation of code in various formats
GCC	GNU compiler collection
Insight	Real-mode debugger for DOS
eCos	Operating system used for the Swisscube

Table 7-4 : List of needed software and tools

Then, on ecos.sourceware.com, there is some explanation about how to correctly configure these tools for an eCos usage.

Yet, the entire environment for eCos is installed and correctly configure. The next step is to configure eCos itself. For that, the configuration tool is used. The principally work is to configure and compile an adapted HAL for the prototype board. For that, the HAL for the LPC2294 has been used as a base for the LPC2292. In the figure below, there is the eCos configuration tool used for the configuration of the HAL:

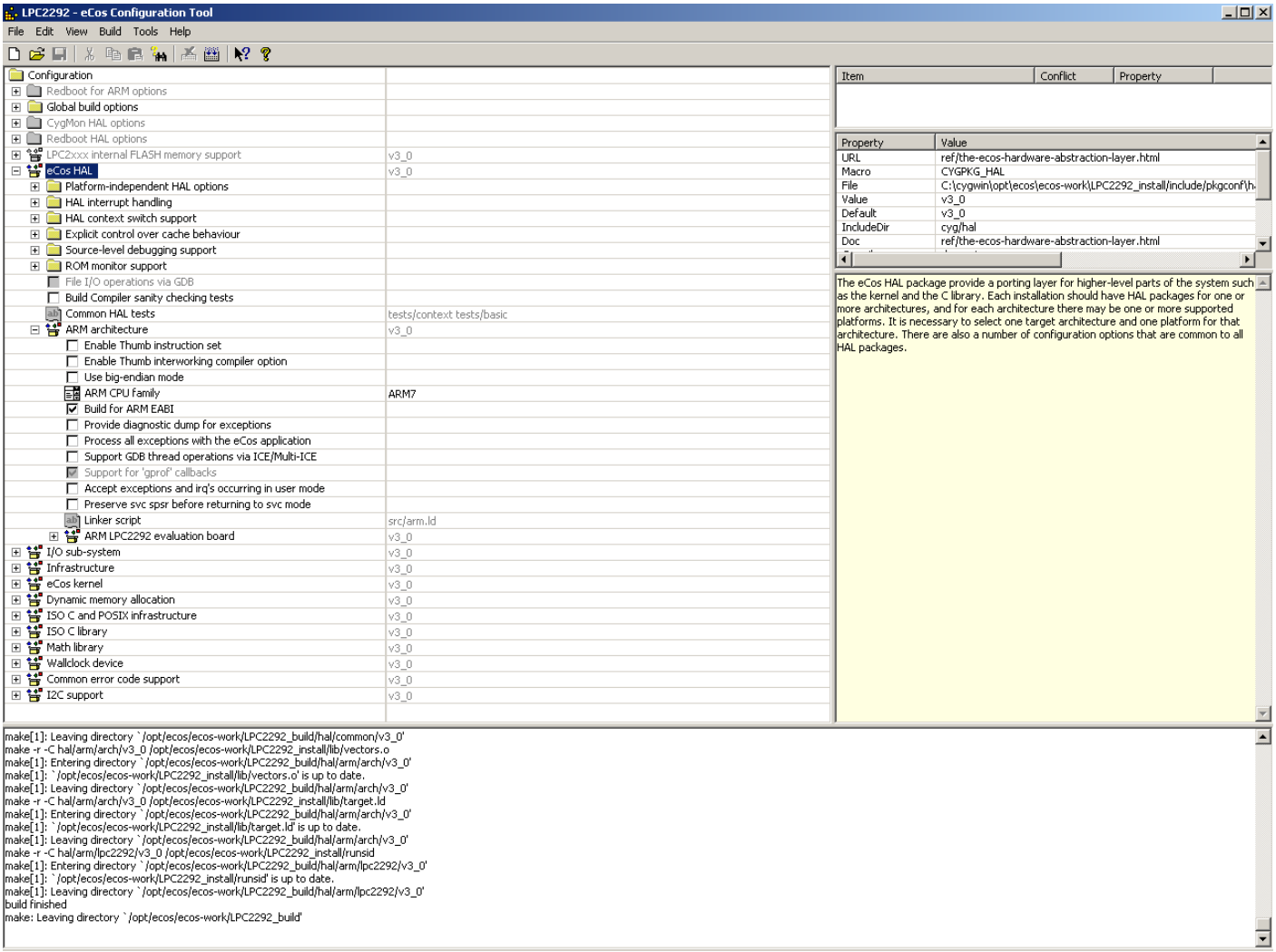


Figure 7-5: eCos configuration tool

On the left is the configuration tree of eCos. As we can see, the HAL is configured for the LPC2292. When eCos is build, the install and build folder are created with all the needed files for running eCos.

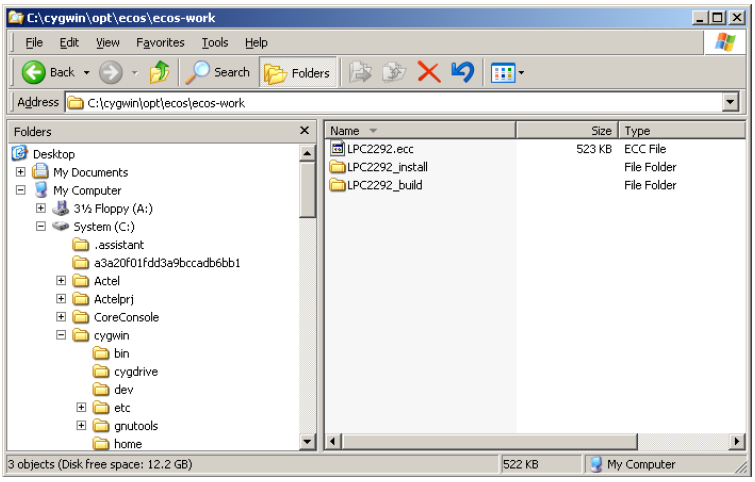


Figure 7-6: eCos files for LPC2292

Now, the HAL is correctly configure, the next step is to test a “hello world” programming with eCos on the prototype board. The compilation of this program with eCos is made with arm-elf-gcc with the following command:

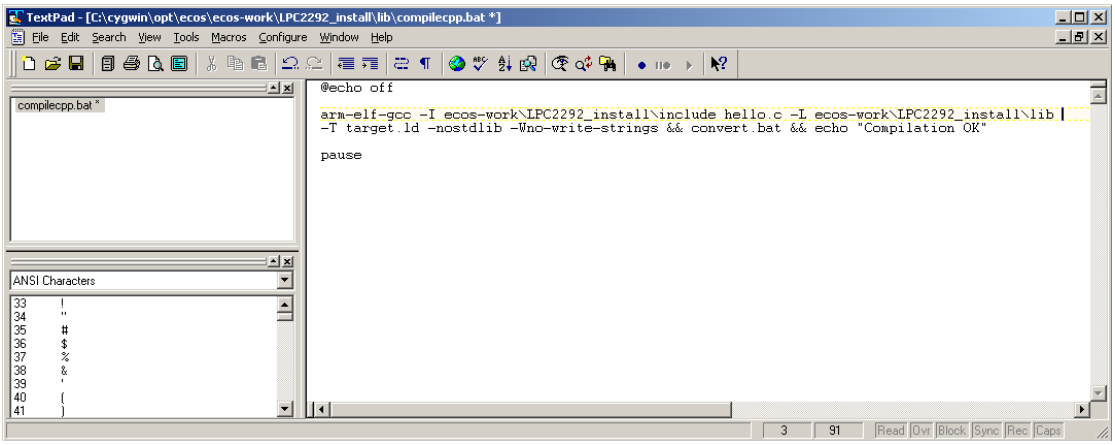


Figure 7-7: Compilation command

The result of this command is shown in the figure 7-8:

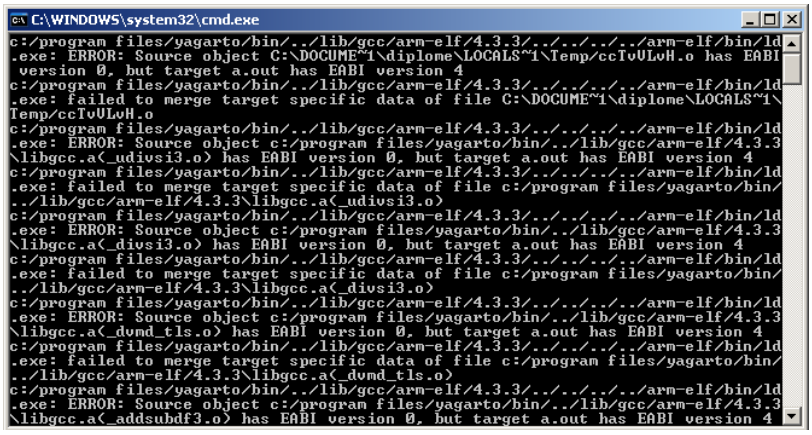


Figure 7-8: Compilation result

There is some problem of EABI version compatibility between the OS and the “hello world” software. Until now, this problem still not resolved. Unfortunately, I don’t have more time in my diploma work to resolve this problem.

8 PROTOTYPE BOARD TO CDMS BOARD ADAPTATIONS

The goal of this section is to list the adaptations that are needed to use the elements of the prototype board on the CDMS board.

8.1 Decoupling capacitors

According to the dimension of the new CDMS card, the regrouping of the decoupling capacitor (as shown on §6.7.3.3) may be useful.

8.2 DC/DC converter

The converter documentation specifies that if the converter is powered by a battery, they need a 1µF capacitor in its supply voltage pin.

8.3 Temperature sensor placement

For the prototype board, the position of the temperature on the PCB was only chosen in order to simplify the routing. For the CDMS board, the temperature sensor should be in the center of the PCB.

9 CONCLUSION

I now understand in details the architecture of the current CDMS board and its functionalities. I have suggest a new architecture which suppress the I^2C and the SPi and use the CAN bus. Besides, I have also proposed a cold redundancy for the new generation of the CDMS board. In collaboration with the technical department of the Hes-so Valais, I have made the prototype board of the CDMS. I have made the hardware test and I have run a simple software on the CDMS board. These tests which have been made demonstrate that the board is functional.

I could not terminate the porting of the OS (eCos) on the prototype board. The HAL is correctly configured but there still have an EABI compatibility problem between the test software and the OS. I have not had enough time to resolve this problem.

10 FUTURE WORK

Obviously, the next work to do is to resolve the last problem of the eCos porting. Then the flight software should be integrated on the prototype board. The next step is to test my prototype board with the other subsystems.

Finally when all the tests will be successful terminated, the next generation of the CDMS board will be realize with the last modification of the prototype board.

Sion, the 05 July 2009 Laurent Lugon Moulin

11 ABBREVIATIONS

ADC	:	Analog to Digital Converter
CAN	:	Controller Area Network
CDMS	:	Command & Data Management System
CMOS	:	Complementary Metal-Oxide-Semiconductor
CPU	:	Central Processing Unit
DCE	:	Data Communication Equipement
DOS	:	Disk Operating System
EABI	:	Embedded-Application Binary Interface
ECOS	:	Embedded Configurable Operating System
EPROM	:	Erasable Read-Only Memory
EPFL	:	Ecole Polytechnique Fédérale de Lausanne
EPS	:	Electrical Power System
FM	:	Flight Model
GCC	:	GNU Compiler Collection
GPIO	:	General Purpose parallel Input Output
GPL	:	General Public Licence
HAL	:	Hardware Abstraction Layout
HE	:	Haute Ecole
HES-SO	:	Haute Ecole Spécialisée de Suisse Occidentale
I/O	:	Input/Output
I^2C	:	Inter-Integrated Circuit
IC	:	Integrated Circuit
ICE	:	In-Circuit Emulator
JTAG	:	Join Test Action Group
LMTS	:	Laboratoire des Microsystèmes pour les Technologies Spatiales
OS	:	Operating System
PC	:	Personal Computer
PCB	:	Printed Circuit Board
PLL	:	Phase Locked-Loop
P-POD	:	Poly-Picosatellite Orbital Deployer
ROM	:	Read-Only Memory
SMD	:	Surface-Mount Devices
SPI	:	Serial Peripheral Interface
SRAM	:	Static Random Access Memory
TTL	:	Transistor-Transistor Logic

12 REFERENCES

There is a list of the principal documents or websites which help me to realize this work:

http://swisscube.epfl.ch/page17388.html
EPFL Swisscube page.

http://swisscube.epfl.ch/webdav/site/cubesat/shared/technical_documentation/s3-c-cdms-report_and_tests.pdf
Diploma work of David Crettaz

http://www.nxp.com/#/ps/ps=[i=45994]|pp=[t=pfp,i=45994
Online part selection guide from NXP

http://www.nxp.com/acrobat_download/datasheets/LPC2292_2294_7.pdf
Data sheet of the LPC2292

http://www.nxp.com/acrobat_download/usermanuals/UM10114_3.pdf
User manual of the LPC2292

http://openocd.berlios.de/doc/pdf/openocd.pdf
OpenOCD user's guide

http://www.yagarto.de/howto.html
Explanations about a GNU ARM toolchain implementation

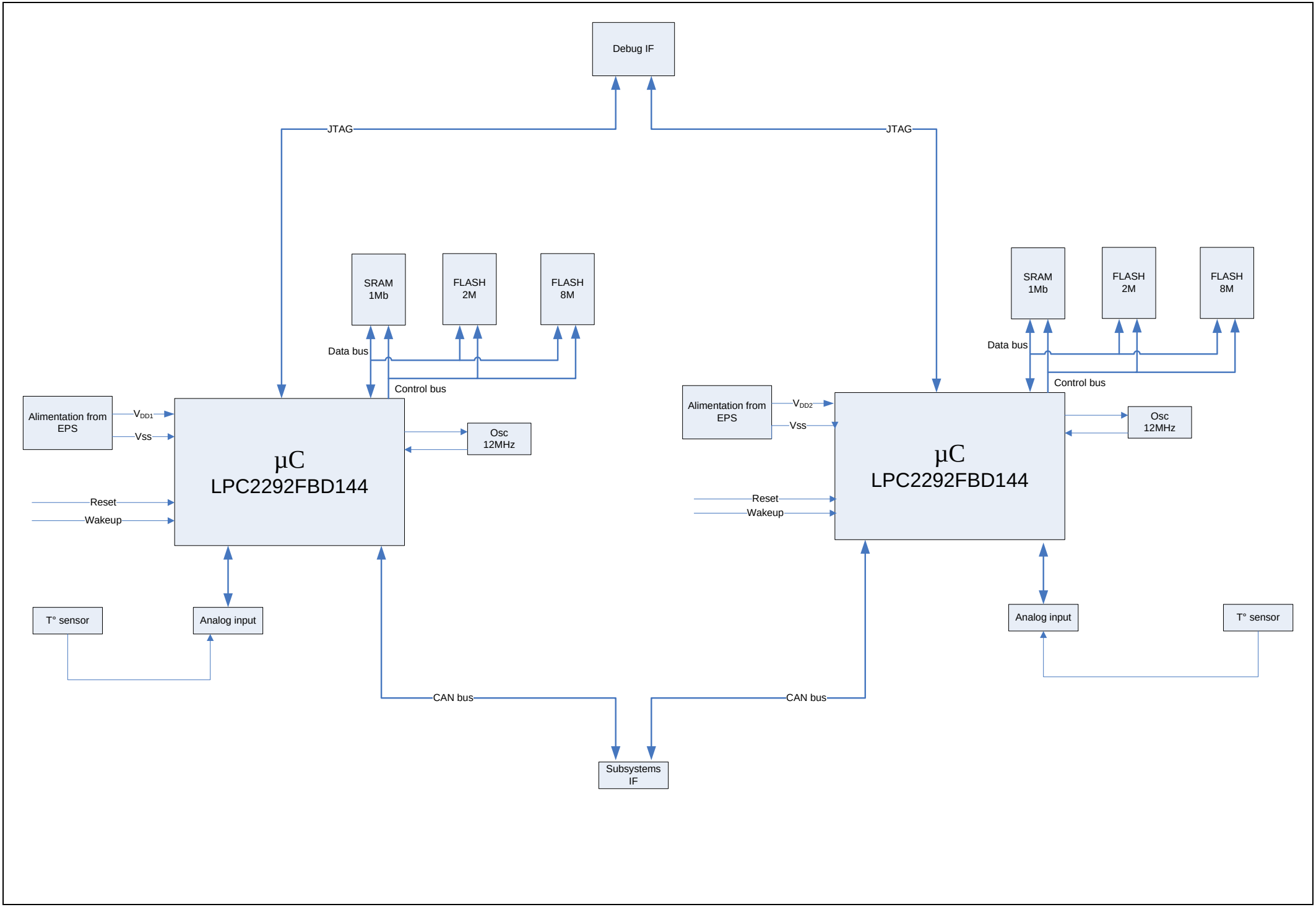
http://ecos.sourceware.org/docs-latest/user-guide/ecos-user-guide.html
eCos user's guide

13 ANNEXES

Annex 1	:	Forecast planning of diploma work
Annex 2	:	Hardware redundancy
Annex 3	:	Complete schematic of the prototype board
Annex 4	:	Bill of material

ANNEX 1: FORECAST PLANNING OF DIPLOMA WORK

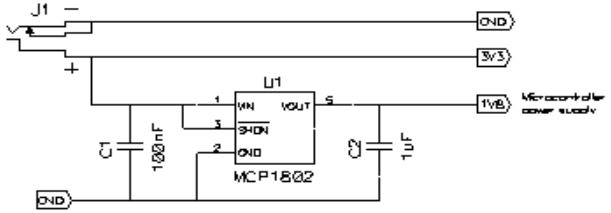
ANNEX 2: HARDWARE REDUNDANCY


Debug IF
JTAG
JTAG
SRAM 1Mb
FLASH 2M
FLASH 8M
Data bus
Control bus
Alimentation from EPS
V_{DD1}
Vss
µC LPC2292FBD144
Osc 12MHz
Reset
Wakeup
T° sensor
Analog input
SRAM 1Mb
FLASH 2M
FLASH 8M
Data bus
Control bus
Alimentation from EPS
V_{DD2}
Vss
µC LPC2292FBD144
Osc 12MHz
Reset
Wakeup
Analog input
T° sensor
CAN bus
CAN bus
Subsystems IF

ANNEX 3: COMPLETE SCHEMATIC OF THE PROTOTYPE BOARD

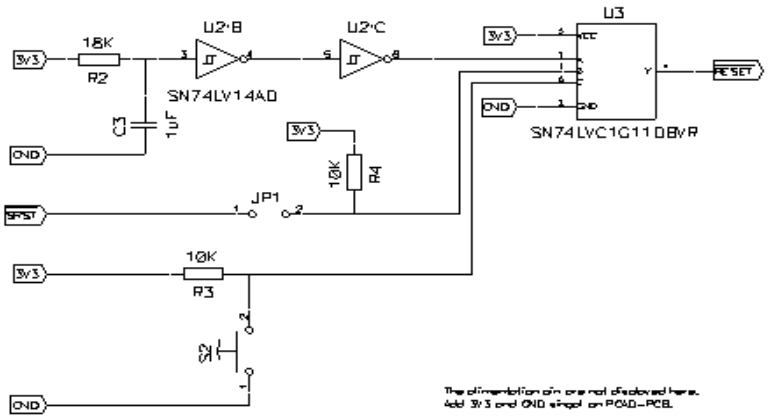
Supply voltage

There is the supply voltage for all the CDMS card

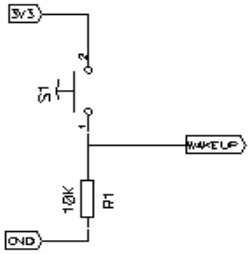


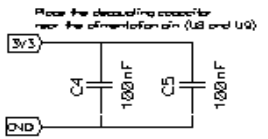
Reset part

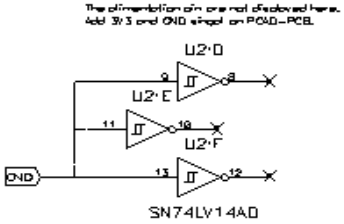
The reset can come from three different sources
The power-on reset, the manual reset or the JTAG system reset



Manual Wakeup



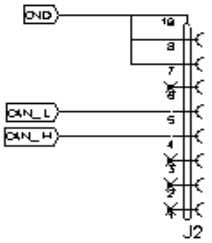




Swisscube CDMS prototype board Signal processing	DES	june 2009 Laurent Lugan Maulin
	REV	V1.0
HAUTE ECOLE VALAISANNE	1/10	schema_V3.SCH

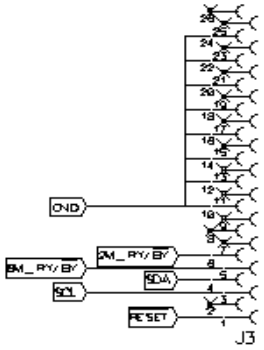
CAN bus

The connection of the CAN signals are the same as the USB-to-CAN connection



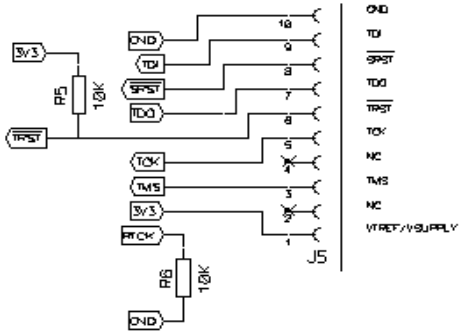
I2C

The connection of the I2C signals are the same as the PICEBS connection (usefull for the debug)

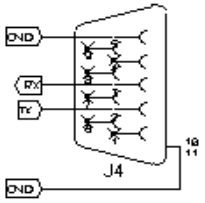


JTAG

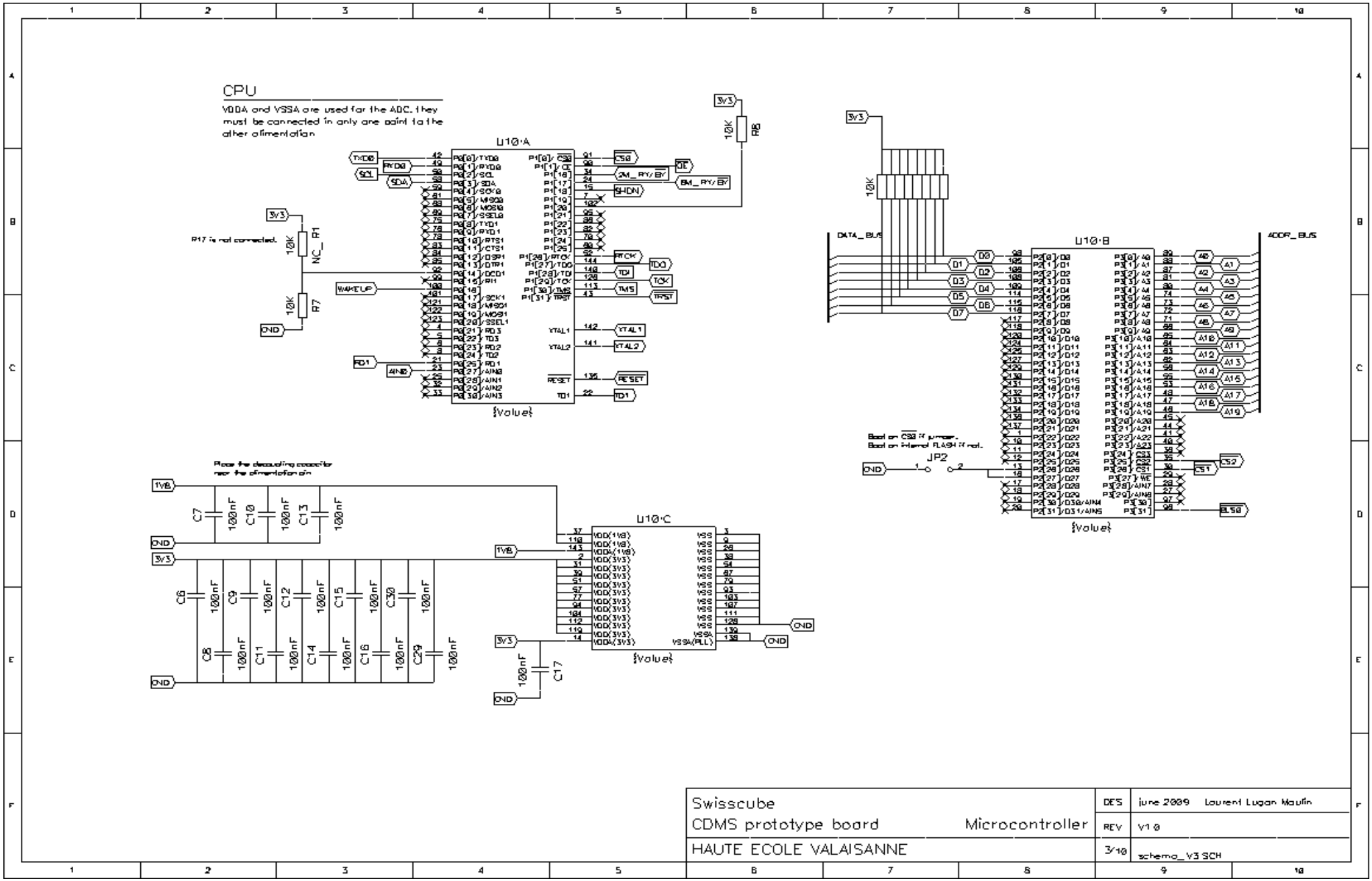
Debug and test interface

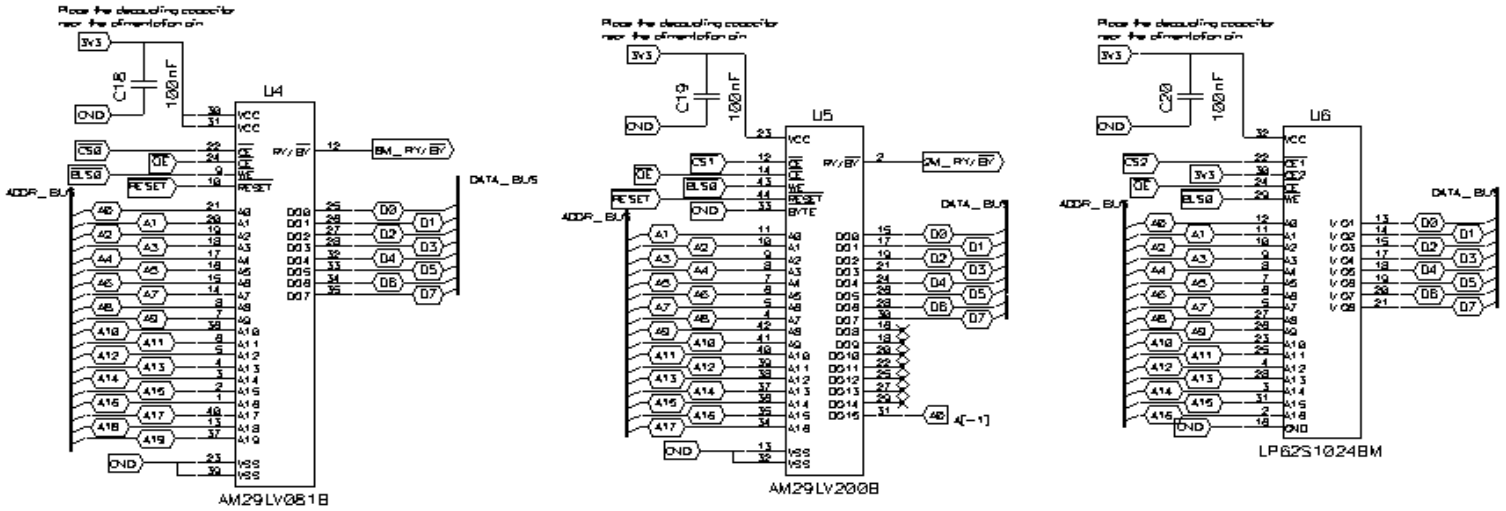


RS232



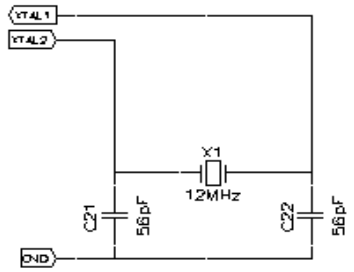
Swisscube CDMS prototype board Connectors		DES	june 2009 Laurent Lugon Moulin
		REV	V1.0
HAUTE ECOLE VALAISANNE		2/10	schema_V3.SCH


CPU
VDDA and VSSA are used for the ADC. they must be connected in only one point to the other alimentation
U10·A
U10·B
U10·C
R17 is not connected.
NC_R1
R7
10K
R8
WAKEUP
TXD0
RXD0
SCL
SDA
CS0
OE
SM_RY/BY
BM_RY/BY
SHDN
RTCK
TDO
TDI
TCK
TMS
TRST
XTAL1
XTAL2
RESET
TDI
RD1
AIN2
DATA_BUS
ADDR_BUS
JP2
Boot on CS0 if jumper.
Boot on Internal FLASH if not.
CS2
CS1
BLS0
Place the decoupling capacitor near the alimentation pin
C7
C10
C13
C6
C9
C12
C15
C30
C8
C11
C14
C16
C29
C17
100nF
1V8
3V3
GND
{Value}
Swisscube
CDMS prototype board
Microcontroller
HAUTE ECOLE VALAISANNE
DES
june 2009 Laurent Lugon Moulin
REV
V1.0
3/10
schema_V3.SCH



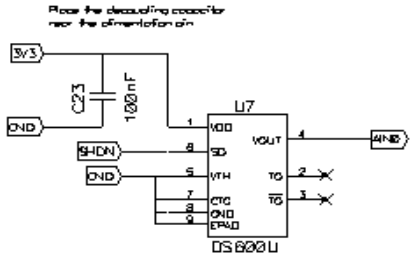
Swisscube CDMS prototype board — Memories		DES	june 2009 Laurent Lugon Moulin
		REV	V1.0
HAUTE ECOLE VALAISANNE		A/10	schema_V3.SCH

Oscillator



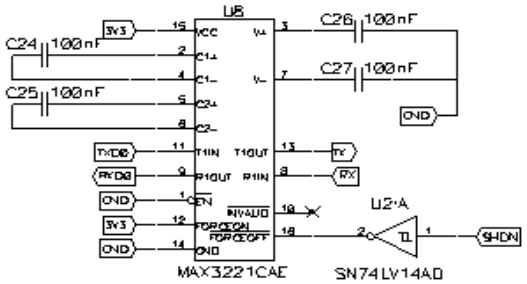
Swisscube CDMS prototype board — Oscillator	DES	june 2009 Laurent Lugon Moulin
	REV	V1.0
HAUTE ECOLE VALAISANNE	5/10	schema_V3.SCH

Temperatur sensor



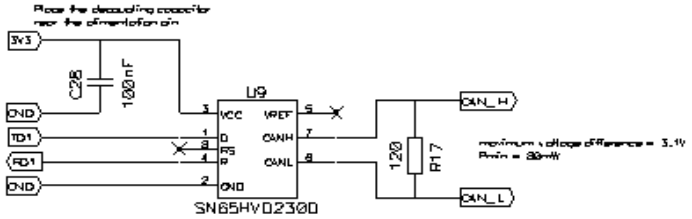
Swisscube CDMS prototype board T Sensor	DES	june 2009 Laurent Lugon Moulin
	REV	V1.0
HAUTE ECOLE VALAISANNE	B/10	schema_V3.SCH

UART Transceiver



CAN Transceiver

The CAN transceiver used in the prototype card will be a SN65HVD2320 Hs pin 5 and 8 are not connected



Swisscube CDMS prototype board Transceivers	DES	june 2009 Laurent Lugon Moulin
	REV	V1.0
HAUTE ECOLE VALAISANNE	7/10	schema_V3.SCH

ANNEX 4: BILL OF MATERIAL

Provider	Q	Article	Price/u	Link
Digikey	1	Microcontroller LPC2292FBD144	$12.66	http://www.nxp.com/acrobat_download/datasheets/LPC2292_2294_7.pdf
Farnell	1	FLASH 2Mb AM29LV200BT-90EC	CHF 2.50	http://www.spansion.com/products/21521d7.pdf
Farnell	1	FLASH 8Mb AM29LV081B	CHF 6.75	http://www.spansion.com/products/21525d7.pdf
Farnell	1	T° sensor DS600U+	CHF 9.25	http://www.farnell.com/datasheets/4511.pdf
Farnell	1	DC/DC 3.3V/1.8V converter	CHF 1.65	http://www.farnell.com/datasheets/37927.pdf
Farnell	1	3 input AND gate	CHF 0.42	http://focus.ti.com/lit/ds/symlink/sn74lvc1g11.pdf
Distrelec	1	SRAM 1Mb LP 62S1024BM-55LLTF	CHF 6.24	http://www.amictechnology.com/pdf/LP62S1024B-T.pdf
Hes-so	1	Quartz 12MHz SMU4	CHF 1.94	http://www.datasheet4u.com/html/Q/-/2/Q-25.0-SMU4-18-30_JauchQuartzGmbH.pdf.html
Hes-so	1	CAN transceiver SN65HVD232	CHF 4.80	http://focus.ti.com/lit/ds/symlink/sn65hvd230.pdf
Hes-so	1	Inverter Schmitt trigger	CHF 1.05	http://www.farnell.com/datasheets/74390.pdf
Hes-so	1	UART transceiver MAX3221	CHF 3.05	http://focus.ti.com/lit/ds/symlink/max3221.pdf
Hes-so	2	1µF capacitor	CHF 0.22	-
Hes-so	26	100nF capacitor	CHF 0.22	-
Hes-so	2	56pF capacitor	CHF 0.22	-
Hes-so	1	18kΩ resistor	CHF 0.09	-
Hes-so	16	10kΩ resistor	CHF 0.09	-
Hes-so	1	120Ω resitor	CHF 0.09	-
Hes-so	2	Jumper	CHF 0.59	-
Hes-so	2	Push button	CHF 2.91	-
Hes-so	1	Alimentation connector DC10B	CHF 1.94	-
Hes-so	1	CAN connector RJ45	CHF 1.80	-
Hes-so	1	JTAG connector CONNB 2x5	CHF 2.91	-
Hes-so	1	I^2C connector CONNB 2x13	CHF 1.80	-
Hes-so	1	UART connector SUB-D-9FC	CHF 5.60	-
Total			**CHF 80.56**	-