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

Filière Systèmes industriels

Orientation Infotronics

Diplôme 2007

David Crettaz

Ordinateur de bord pour mini satellite

Professeur	Christophe Bianchi
Expert	Olivier Moulin

Sion, le 23 novembre 2007

HES-SO Valais

SI	TV	EE	IG	EST
X	X	X	X	

Données du travail de diplôme
Daten der Diplomarbeit

FO.0.2.02.07.DB
DD/18/05/2006

Filière / *Studiengang* : Systèmes industriels

Confidentiel / *Vertraulich* ☐

Etudiant / *Student* David Crettaz	**Année scolaire / *Schuljahr*** 2006/07	**No TD / *Nr. DA*** SI/2007/4
Proposé par / *vorgeschlagen von* HES-SO Valais, UIT		**Lieu d'exécution / *Ausführungsort*** HES-SO Valais, DSI **Expert / *Experte*** Olivier Moulin (Syderal SA)

Titre / *Titel*:

Ordinateur de bord pour mini satellite

Description / *Beschreibung*:

Petit cube de 10 cm de côté, ne pesant qu'un kilo, le premier picosatellite de l'EPFL et de la HES-SO (appelé SwissCube) sera lancé en 2008. Construit principalement par les étudiants des deux écoles mentionnées précédemment, la HES-SO Valais participe à l'étude et à la réalisation de l'ordinateur de bord du satellite, à savoir le CDMS (control & data management sub-system).

Le but du travail de diplôme est de réaliser les modèles de qualification et de vol de la carte CDMS tenant compte des premiers tests réalisés sur prototype. Les activités se décomposent de la manière suivante :

- finalisation des tests de la carte prototype
- interface avec les autres sous-systèmes
- modification de la schématique, développement et tests des modèles de qualification et de vols

Objectifs / *Ziele:*

— Développement du modèle de qualification et de vol de la carte CDMS du Swisscube
— Participation aux réunions projets du swisscube.

Signature ou visa / *Unterschrift oder Visum*	**Délais / *Termine***
Resp. de l'orientation infotronics	Attribution du thème / Ausgabe des Auftrags: 03.09.2007
Professeur/*Dozent*: Christophe Bianchi	Remise du rapport / Abgabe des Schlussberichts: 23.11.2007
Etudiant/*Student*:	Exposition publique / Ausstellung Diplomarbeiten: 30.11.2007
	Défenses orales / Mündliche Verfechtungen Semaine 49

Rapport reçu le / *Schlussbericht erhalten am* ..... Visa du secrétariat / *Visum des Sekretariats:* .....

Crettaz, David 2007

Ordinateur de bord pour mini satellite

Boardcomputer für einen Picosatelliten

Objectif

Développement d'une nouvelle carte électronique pour le sous-système CDMS (Control & Data Management System) du satellite Swisscube, sur la base des résultats de tests du modèle prototype. Ce picosatellite développé entièrement par des étudiants de différentes hautes écoles de Suisse, sera lancé vers la fin de l'année 2008.

Résultats

Le modèle de qualification du CDMS a été développé et testé. Ces différents tests ont permis de démontrer le bon fonctionnent de la carte pour les différentes fonctionnalités et spécifications requises. La carte du CDMS est entièrement opérationnelle et peu être transmise au team responsable du développement software du CDMS, pour effectuer leurs tests d'intégration.

Mots-clés

CDMS, CubeSat, Swisscube, microcontrôleur, mémoires, I2C, SPI, OpenOCD, JTAGkey, Eclipse

Ziel

Entwurf einer neuen elektronischen Karte für das Subsystem CDMS des Swisscube Satelliten, welche auf die Resultate von Tests des Prototypenmodells basiert ist. Dieser Picosatellit, welcher von Schweizer Fachhochstudenten entwickelt worden ist, wird Ende 2008 gestartet.

Resultate

Das Qualifikationsmodell des CDMS wurde entwickelt und getestet. Diese Tests erlaubten es, das gute Funktionieren der verschiedenen erforderlichen Funktionalitäten und Spezifizierungen zu beweisen. Die CDMS Karte ist einsatzfähig und kann dem verantwortlichen Team der Entwicklung der CDM -Software weitergeleitet werden, um die Integrationsteste durchzuführen.

Schlüsselwörter

CDMS, CubeSat, Swisscube, Mikrokontroller, Speicher, I2C, SPI, OpenOCD, JTAGkey, Eclipse

Phase C

CDMS

Control &
Data Management System

Prepared by:
Crettaz David

Checked by:

Approved by:

•
HES-SO
Sion
Switzerland
•
22/11/2007
•

RECORD OF REVISIONS

ISS/REV	Date	Modifications	Created/modified by
1/0	22/11/2007	First Edition	Crettaz David

1 INTRODUCTION

Swisscube is the name of a Picosatellite which will be launched at the end of the year 2008. He has the particularity to be entirely built by students of different Swiss university (EPFL, HES-SO, HE-ARC) who respect the CubeSat standard.

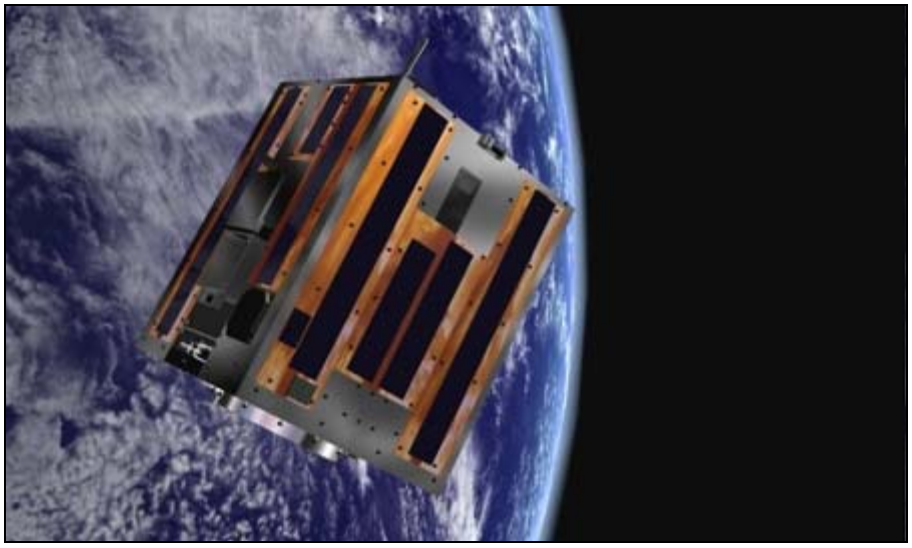


Figure 1 : CubeSat in Space

The development of a space project being a complex and long task is why the construction of the various subsystems was divided into various groups. The HES-SO Sion task consists of the development of the subsystem named CDMS (Control and Data Management System). A prototype board was elaborated during the phase B of the project, by the student Pierre André Tapparel during his semester and diploma work. M. Christophe Bianchi, teacher at the HES-SO Sion was is 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 CubeSat Standard definition

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 Poly Picosatellite Orbital Deployer 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 2 : P-POD interface

2 DIPLOMA OBJECTIVES

The goal of this diploma work on this project is to finish the tests of the CDMS engineering model (EM) that I've began during the semester project. With the result of these tests, a second PCB will be build, the qualification model (QM). This board should be delivered at the end of my diploma.

Here is a description of the main planned tasks to realize during this diploma project:

- Finish the tests of the CDMS engineering model (EM) to validate the actual design
  - Check the communication with external peripherals (FLASH, SRAM)
  - Check the Shut Down , Wakeup and the Watchdog feature
  - Consumption tests
- Functional analysis modification to perform for the Qualification Model (QM)
- Schematic modification for the Qualification Model (QM)
- Listing and command of all the components to build the new PCB
- Assembly of the new PCB
- Tests of the Qualification Model (QM)
- Make the PSA analysis (Part Stress Analysis)

3 DESCRIPTION OF THE SWISSCUBE SUBSYSTEMS

3.1 Swisscube architecture

Swisscube is composed of 5 various subsystems communicating via a bus I2C. All the functions of the various subsystems which require a great computing power will be carried out by the CDMS.

The figure below shows an overall picture of the organization of the various subsystems with their principal tasks.

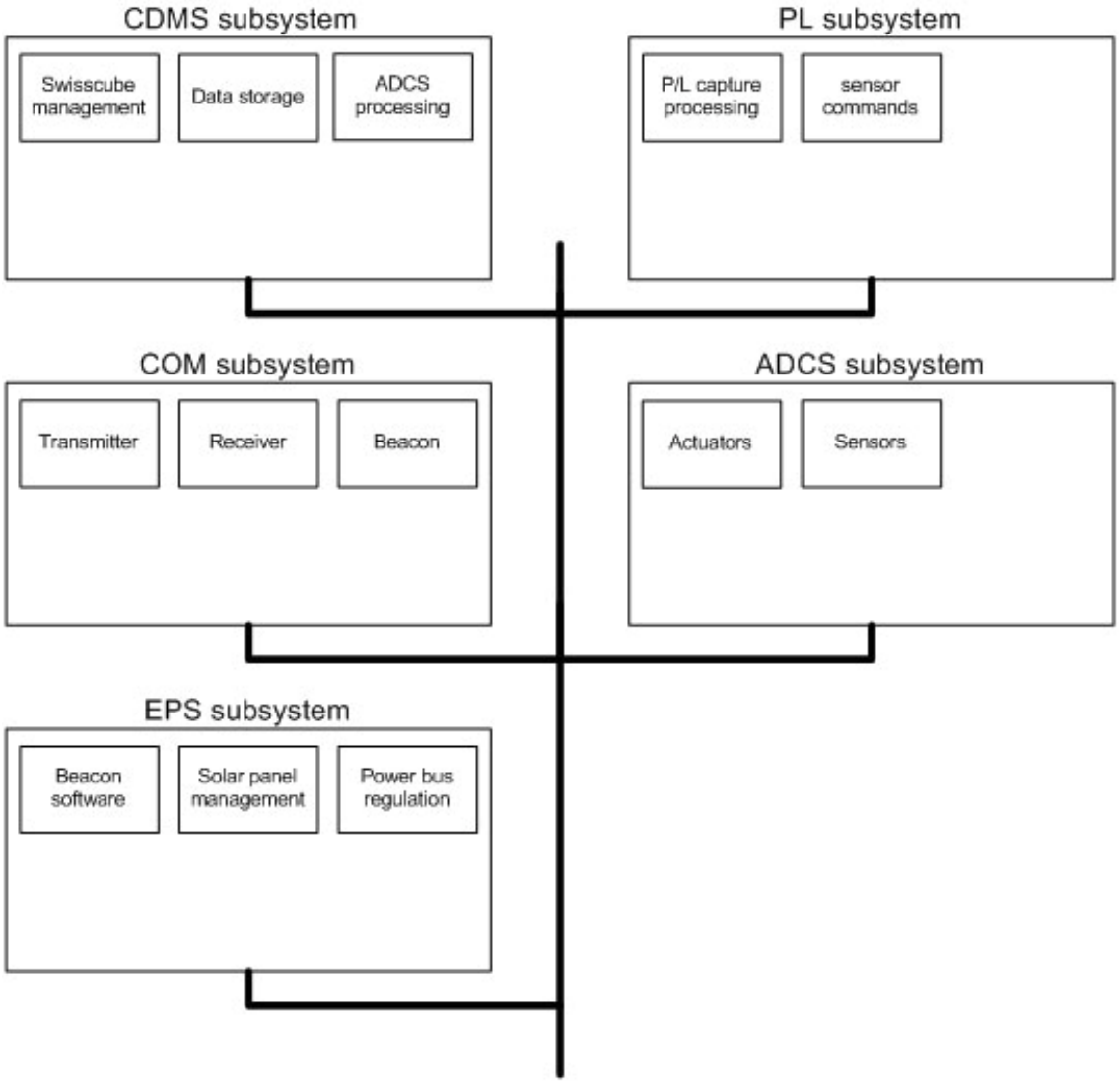


Figure 3 : Swisscube architecture

3.2 Brief description of the subsystems

The goal of this part is to provide a brief description of 5 subsystems, to permit the reader to understand the different functions performed on the Swisscube satellite.

3.2.1 Control and data-management (CDMS)

The CDMS has two main functions. The first is to execute the flight software in order to schedule the spatial mission and moreover to supply computing services for other subsystems. The second is to provide communication between the satellite and the ground station for the purpose of command and control, obtaining spacecraft safety and systems status as well as sensor data transfer.

3.2.2 Attitude determination and control system (ADCS)

The microcontroller included on the ADCS board will manage the sensors and actuators in the way to control the attitude of the satellite, all the processing will be done on the CDMS ARM microcontroller.

3.2.3 Communication subsystem (COM)

The COM must be able to send and receive data from the ground station.

3.2.4 Payload (PL)

The payload will take pictures of nightglows effect and transmit them on the CDMS via the I2C bus for the storage, until the satellite will be able to transmit those pictures to the ground station.

Figure 4 : Nightglows phenomenon

3.2.5 Electrical power subsystem (EPS)

The EPS supplies all the subsystems. This task will be made by capturing solar energy during the daylight and store it in batteries for later power usage.

4 CDMS SPECIFICATION (LEVEL 4)

In this chapter, only the main specifications will be presented. For further information about the entire specification of the CDMS board see the following document:

→S3-B-SE-2-0-CDMS_specification

4.1 Functional Specification

The main activities of the CDMS are:

- To perform the scheduling of the space system functions
- To perform data storage for the space system
- To maintain life statuses of the space system
- To process data coming out of temperature sensor
- To execute the attitude determination and attitude control algorithms (ADCS)

4.2 Modes

The different operational modes on the CDMS are:

- **OFF:** In this mode, the whole subsystem shall be turned off (no function available)
- **STAND-BY**: In this mode, only the CDMS shall be in low power
- **INITIALISATION:** This mode is a transient mode after a power on and the CDMS shall initiate all low-level functions
- **OPERATIONAL:** In this mode the CDMS shall be fully operational
- **SHUTDOWN:** This mode is a transient mode which led to off mode

4.3 Memories requirement

- **Final boot program**
  - The CDMS shall provide a 512 Kbytes memory (EPROM)
- **Application SW and data storage**
  - The CDMS shall provide a 2 Mbytes memory (FLASH)
- **Execution program memory and temporary data**
  - The CDMS shall provide a 512 Kbytes memory (SRAM)

4.4 Electrical requirement

- **Consumption in OFF mode**
  - The CDMS shall consume no power
- **Consumption in STAND-BY mode**
  - The CDMS shall consume 1 mW
- **Consumption in OPERATIONAL mode**
  - The CDMS shall consume at average 150 mW and at peak 250 mW in operational mode

4.5 Software performance

- **Timestamp**
  - To timestamp each picture coming from the payload
- **CDMS Status 1**
  - To deliver board temperature (one sensor)
- **CDMS Status 2**
  - To deliver reset source
- **CDMS Status 3**
  - To deliver mode status
- **Compression of scientific data**
  - If the pictures coming from the payload will be compressed, this compression shall be done by the CDMS
- **Orbital position determination**
  - During observation, the orbital position of the space system shall be known by the CDMS subsystem

4.6 Reliability and redundancy

- **SEU**
  - CDMS HW and SW design for critical functions shall mitigated possible SEUs
- **Tests**
  - Reliability of the electrical systems shall be demonstrated by tests

4.7 Electrical interfaces

- **Power link**
  - Two power links composed of two GND and two VDD +3.3V [+/- 7%]

4.8 Data interfaces

- **I2C**
  - This interface permits to connect all subsystems together for communication and command signals
  - The pictures taken by the payload are transmitted by I2C to the CDMS
- **TM and TC**
  - The CDMS shall manage telemetry (TM) and telecommand (TM) coming form Space System
- **Payload control**
  - The CDMS shall define the required parameters to operate the payload according to the command updates from the Ground Station (GS), if required. Modifiable parameters are:
    - Image capturing integration time
    - Image saturation intensity
    - Image intensity resolution
- **Science data product**
  - The CDMS shall format the science data product as specified
- **JTAG**
  - Interface to debug the microcontroller AT91M55800A and the MSP430F1611

4.9 Physical properties

- **CDMS board dimensions and constraints**
  - The CDMS shall fulfil the following dimensions :

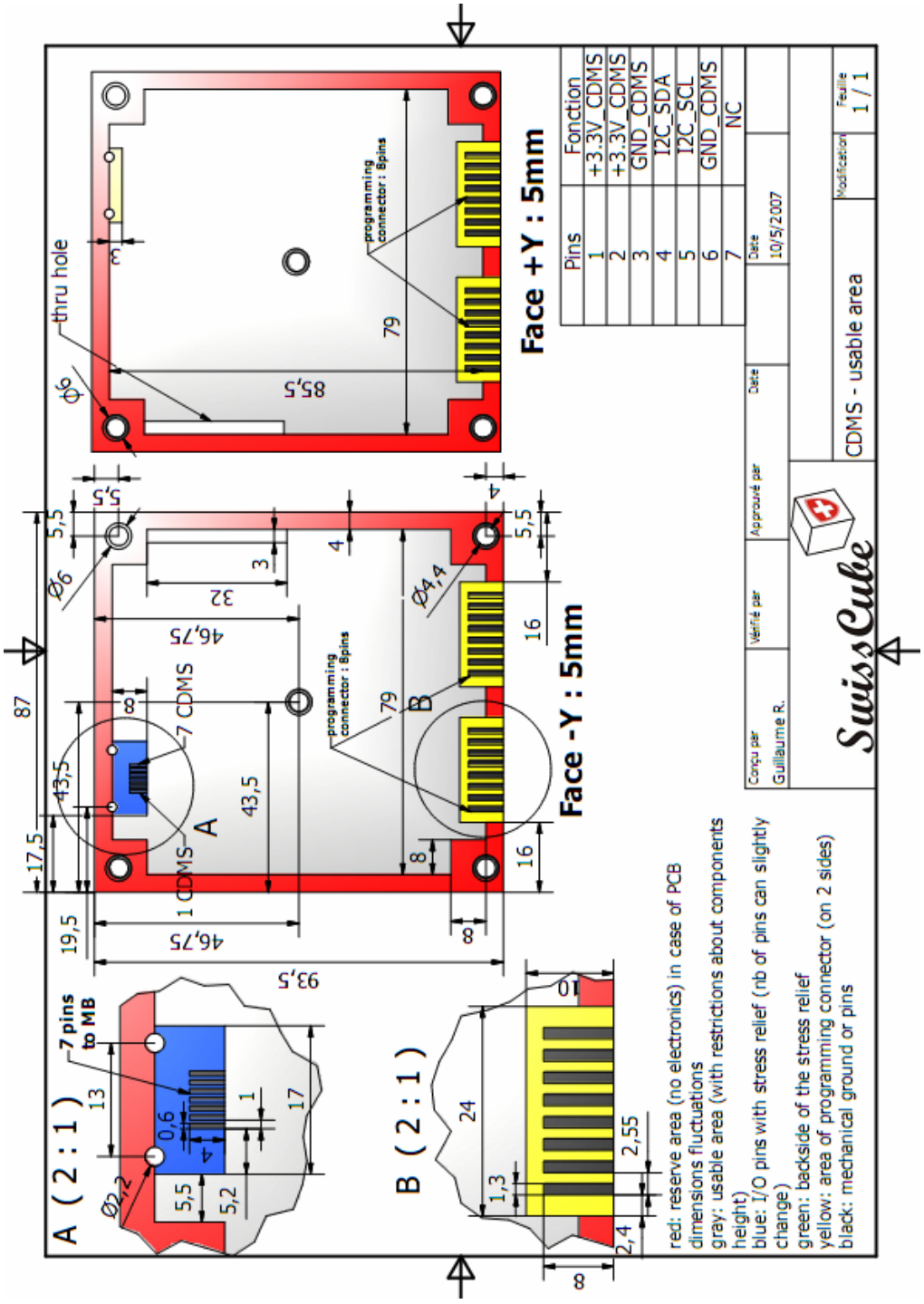


Figure 5 : Mechanical dimensions of the CDMS board

4.10 Environmental

- **Environment**
  - The CDMS shall survive under space environment at earth distances between 400 and 1000 km
- **Thermal**
  - The CDMS shall survive at temperature range from -30 to +60 °C in OFF and OPERATIONAL mode
- **Acceleration**
  - The CDMS shall survive an acceleration of 10.4 g
- **Random and Sine vibration**
  - The Space System shall be able to sustain the random and Sine vibration qualification and acceptance tests for the frequency range specified
- **Vacuum**
  - The CDMS shall operate under vacuum conditions
- **Radiation (Total dose)**
  - The CDMS shall survive to a TID of maximum 20 kRad

4.11 Operational

- **Autonomy**
  - The CDMS shall have an operational life time of 1 year on earth and 3 months in orbit
- **Control**
  - The power on of the CDMS shall be directly performed with main power bus
  - The power off of the CDMS shall be given by a specific I2C command
- **Failure propagation**
  - Failure of one part or element of the CDMS shall not result in consequential damage to the equipments or other satellite components
- **Recovery plans**
  - For the nominal phase, possible failure scenarios and recovery plans shall be elaborated

5 TESTS OF THE CDMS ENGINEERING BOARD

During the semester project, a problem of JTAG connection with the board was identified. A lot of time was spent to resolve this problem, because it was not possible to debug the board and also to check all the functionalities. At the end of the project, the problem was found, but it was too late to finish all the tests on the CDMS Engineering Model (EM). The first point that was made at the beginning of this diploma work was to finish all the tests on the engineering board of the CDMS, in order to validate the functionalities done by the specifications, to build a second board, the Qualification Model.

5.1 Effectuated tests on the CDMS Engineering Model

Here are the lists of the effectuated tests during the semester project on the CDMS:

- **Electrical and Hardware tests**
  - Shortcut test on the power supply
  - Power check of all components
  - Test of the reset circuitry to ensure a knows state of the microcontroller when a reset is to perform
  - JTAG signals check between the interface and the ARM microcontroller
  - JTAG Connexion to the board with OpenOCD
  - Software test performed with a development environment (SDK4ARM)

5.2 I2C problem

A problem of I2C was detected on the CDMS board with the I2C Bridge that was chosen for the engineering board. The problem is that the I2C Bridge SC18IS600 of Philips can only operate in master or multi-master mode. So the other subsystems can't directly contact the CDMS by I2C because all transfers are initiated by the master (ARM microcontroller).

Some solutions were proposed at this problem:

- A new interruption line between each subsystem and the CDMS, but with this solution the CDMS must always be supplied.
- The second was to implement an I2C multi-master / multi-slave in software, but it's very delicate and the time for the implementation is not known by the software team.
- The best solution seems to replace the actual I2C Bridge with a MSP430F1611 from Texas Instruments, which is the same microcontroller, used in others subsystems. This microcontroller has the advantage to manage the I2C in multi-master / multi-slave mode. He also has an SPI interface to communicate with the ARM microcontroller. The other advantage is that all the I2C will be managed by the same microcontroller for all subsystems. It will be better for the debug tests and to make the software. So for the next design, it will be necessary to integrate this MSP430F1611 on the CDMS Qualification Model board.

5.3 External peripheral access on AT91M55800A

To testing the functionalities of the CDMS Engineering Model, it's necessary to check the communication with the different externals peripherals present on the CDMS board like Flash and Sram.

5.3.1 External Bus Interface (EBI)

The EBI generates the signals that control the access to the external memory or peripheral devices. The EBI is fully-programmable and can address up to 128M bytes. It has eight chip selects and a 24 bit address bus.

The 16-bit data bus can be configured to interface with 8 or 16 bit external devices. Separate read and write control signals allow for direct memory and peripheral interfacing.

The EBI supports different access protocols allowing single-clock cycle memory accesses.

The main features are:

- External memory mapping
- 8 active-low chip select lines
- 8 or 16-bit data bus
- Byte-write or byte-select lines
- Remap of boot memory
- Two different read protocols
- Programmable wait state generation
- External wait request
- Programmable data float time

5.3.2 External Memory Mapping

The memory map associates the internal 32-bit address space with the external 24-bit address bus. The memory map is defined by programming the base address and page size of the external memories. Note that A0 - A23 is only significant for 8-bit memory; A1 - A23 is used for 16-bit memory.

5.3.3 EBI User Interface

To access the external memories on the CDMS board, it's necessary to program the EBI using the corresponding registers. The Remap Control Register (EBI_RCR) controls exit from Boot Mode. The Memory Control Register (EBI_MCR) is used to program the number of active chip selects and data read protocol. Eight Chip-select Registers (EBI_CSR0 to EBI_CSR7) are used to program the parameters for the individual external memories. Each EBI_CSR must be programmed with a different base address, even for unused chip selects.

5.3.4 Configuration of chip select register NCS0-NCS3

On the CDMS board, we have four external devices connected on the EBI, so it's necessary to configure four Chip-select.

- NCS0 EPROM 512 kB Final Code (For tests we use a Flash 2MB)
- NCS1 Flash 2 MB Device code
- NCS2 Flash 2 MB Command code
- NCS3 SRAM 512 kB Pictures from payload

After the remap command the external device is accessible at the following addresses

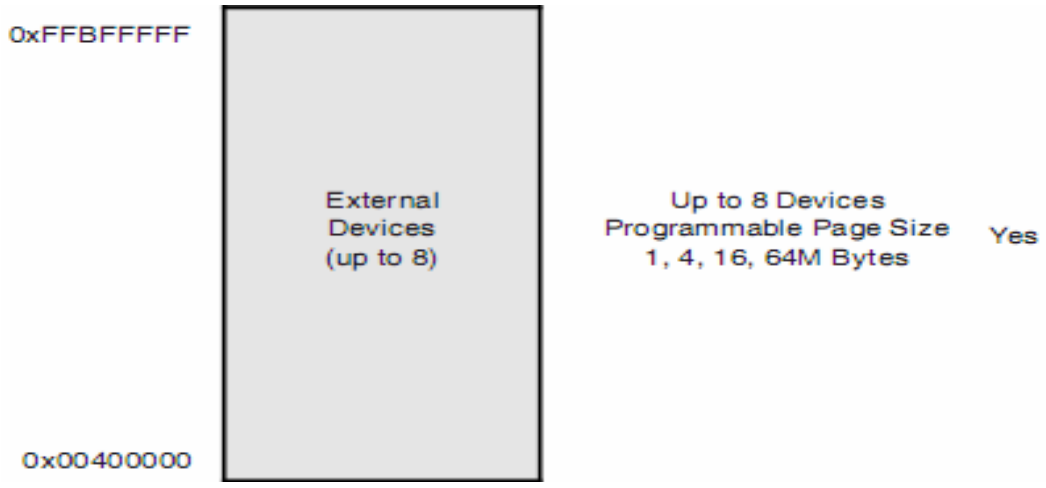


Figure 6 : External devices mapping

The NCS0, NCS1, NCS2 are all connected to a 2MB Spansion Flash. So the value of these three Chip-Select Registers will be the same except for the base address. The NCS3 is connected to the SRAM of 512kB

The following parameters must be configured from the following base address:

- 0xFFE00000 for Chip Select 0 (NCS0) Flash 2 MB
- 0xFFE00004 for Chip Select 1 (NCS1) Flash 2 MB
- 0xFFE00008 for Chip Select 2 (NCS2) Flash 2 MB
- 0xFFE0000C for Chip Select 3 (NCS3) SRAM 512 kB

Here are the values to configure the four Chip Select at their corresponding base address:

- **Value of Chip Select 0 :** 0x100032B1 **at base address** 0xFFE00000
- **Value of Chip Select 1 :** 0x200032B1 **at base address** 0xFFE00004
- **Value of Chip Select 2 :** 0x300032B1 **at base address** 0xFFE00008
- **Value of Chip Select 3 :** 0x400032BD **at base address** 0xFFE0000C

Each Chip Select register must be programmed with a different base address, even for unused chip selects.

- **Value of Chip Select 4 :** 0x50000000 **at base address** 0xFFE00010
- **Value of Chip Select 5 :** 0x60000000 **at base address** 0xFFE00014
- **Value of Chip Select 6 :** 0x70000000 **at base address** 0xFFE00018
- **Value of Chip Select 7** : 0x80000000 **at base address** 0xFFE0002C

For further information about the configuration of these registers see:

→EBI: Chapter 11 on datasheet "AT91M55800A.pdf"

5.4 OpenOCD

OpenOCD created by Dominic Rath is a GDB-Server and flash utility for ARM microcontrollers. It supports ARM7 and ARM9 microcontrollers.

Wiggler-type and FTDI2232-based device (JTAGKEY of Amontec) can be used as hardware-interfaces.

The latest OpenOCD version can be downloaded at the following address:

http://www.yagarto.de

To be executed, OpenOCD need:

- A configuration file witch specify the type of hardware interface and controller used (*.cfg). For our microcontroller ARM7TDMI we need the following configuration file:

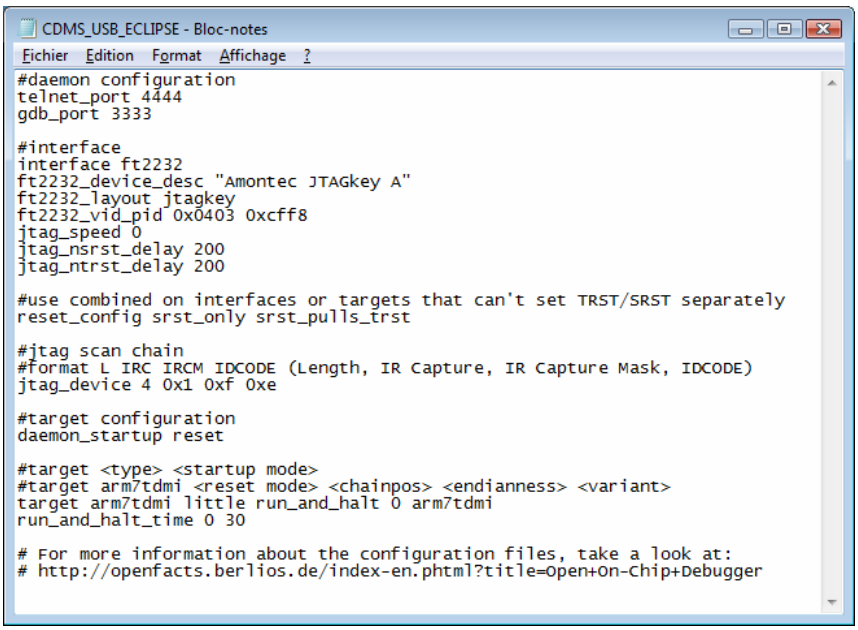


Figure 7: OpenOCD configuration file

- Eventually a file command which contain to list of command to be executed by OpenOCD for the microcontroller ARM (*.ocd) or (*.script). In this case, this file must be pointed in the configuration file.

For detailed explications about the commands of OpenOCD see:

→ **Appendix A**

5.5 Launch OpenOCD

To launch OpenOCD automatically, I wrote a (*.bat) file which contain the link to the executable OpenOCD file and his needed config file in parameter.

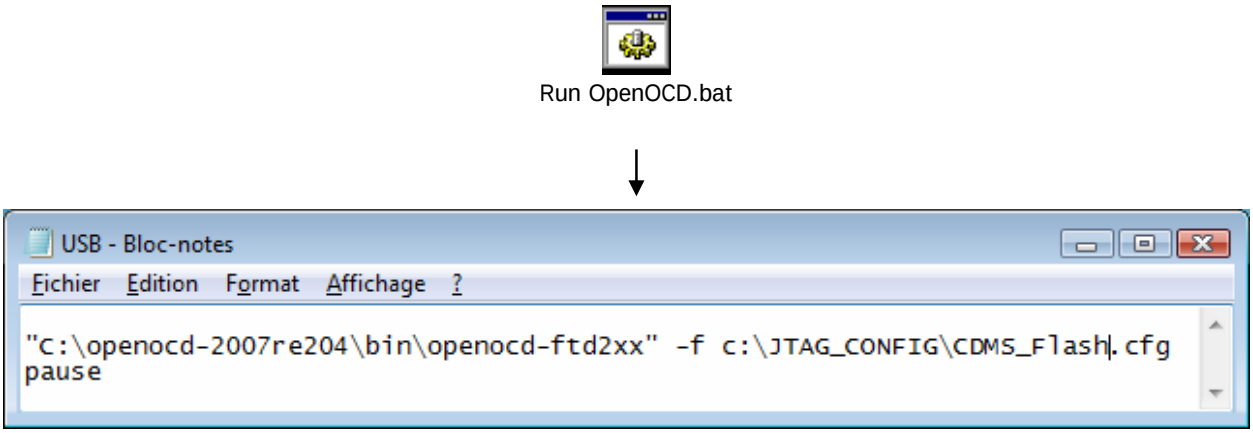


Figure 8 : Batch file for OpenOCD

After to have launched OpenOCD, the following DOS command appear, with the OpenOCD version and some other debug information's.

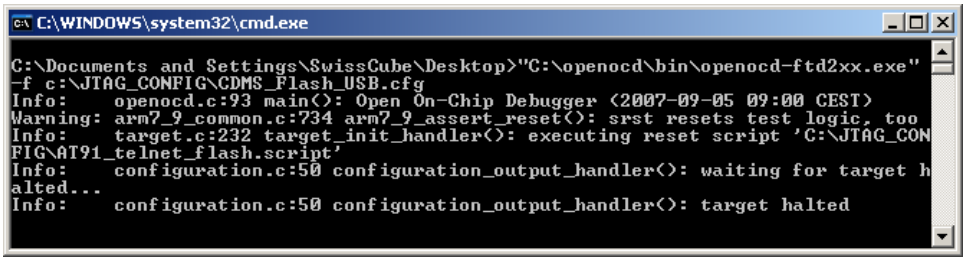


Figure 9 : OpenOCD DOS console

After that it's necessary to launch the server to execute the command in order to communicate with the ARM controller. To make that, I use the program called Putty. Putty is a program which permits to connect at server using protocol like SSH, Telnet or Rlogin. This program can be downloaded at the following address:

http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

Then it's necessary to configure the program like showing in the following picture and the click on the Open button.

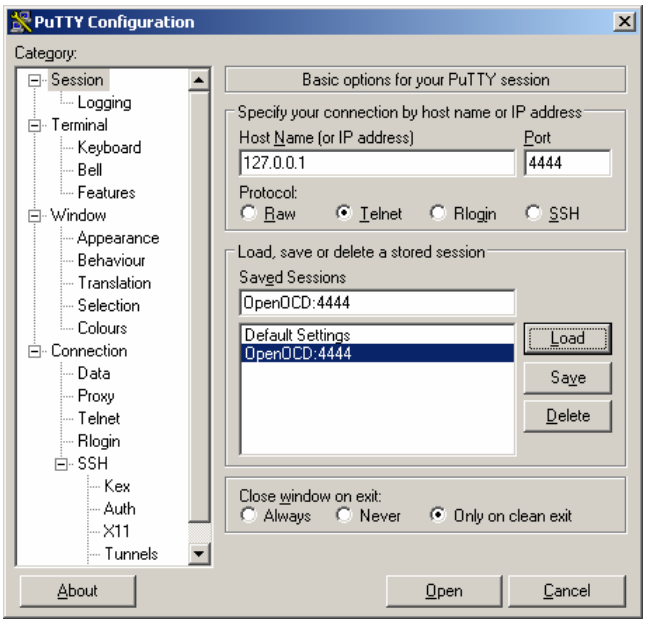


Figure 10 : Putty configuration

The connection is now established with the ARM microcontroller. OpenOCD commands could be executed in order to communicate and execute operations with the target.

Figure 11 : OpenOCD console

For example, on the following picture, the "poll" command is executed and the state of the target is returned, so the JTAG connection works.

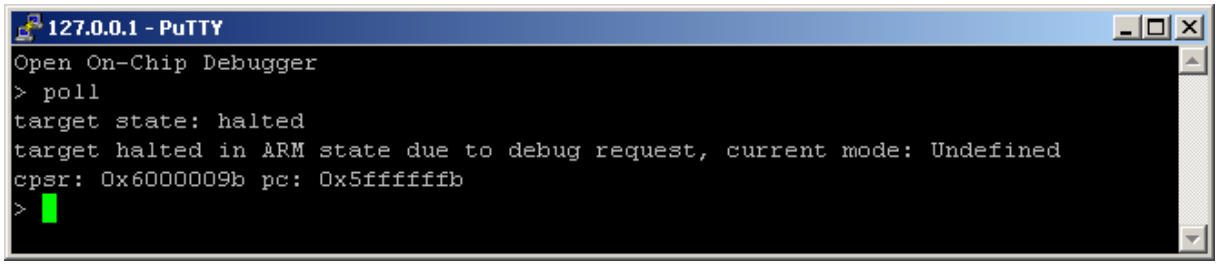


Figure 12 : OpenOCD poll command

5.6 Flash access

5.6.1 Objective

The goal of this test is to check the communication with the three external Spansion Flash connected to the NCS0, NCS1 and NCS2. To simplify, these access will be managed by OpenOCD's commands in order to make access like:

- Read, write, erase, programming, read flash information's

5.6.2 CFI definition

CFI (Common Flash Interface) is a JEDEC standard database that may be read from a Flash memory. It allows Flash driver software to query the installed device to determine the proper configuration.

From the CFI mode the user can access the following information:

Spansion's Flashes using the Common Flash Interface (CFI) specification which permit to interrogate the flash in order to read some information like:

1. A standard string of characters that any CFI compliant device will display so that Flash driver software can confirm that a CFI device is in use.
2. Information on device voltage and timing parameter.
3. Information on command protocol, device size, and sector sizes.
4. Information on vendor or device specific features that are supported.

5.6.3 OpenOCD configuration for Flash operations

OpenOCD also supports programming of external Common Flash Interface (CFI) flash memories. Some specific commands are done by OpenOCD and must be written at the end of the configuration file.

```
#-----------------------------------------------------------------------------
#flash configuration

flash bank cfi 0x10000000  0x00200000 2 2 0 # size 2 MB (0x00200000) NCS0
flash bank cfi 0x20000000  0x00200000 2 2 0 # size 2 MB (0x00200000) NCS1
flash bank cfi 0x30000000  0x00200000 2 2 0 # size 2 MB (0x00200000) NCS2
```

OpenOCD command: flash bank <driver> <base> <size> <chip_width> <bus_widht> <target>

Figure 13 : Flash configuration for OpenOCD

Calculation of the size: $2MB = 2 * 1024 * 1024 = 2097152_d = 0x200000_h$

To permit the access of theses Flashes the different chip select should be configured. These commands are written in the command file and are executed when we start OpenOCD.

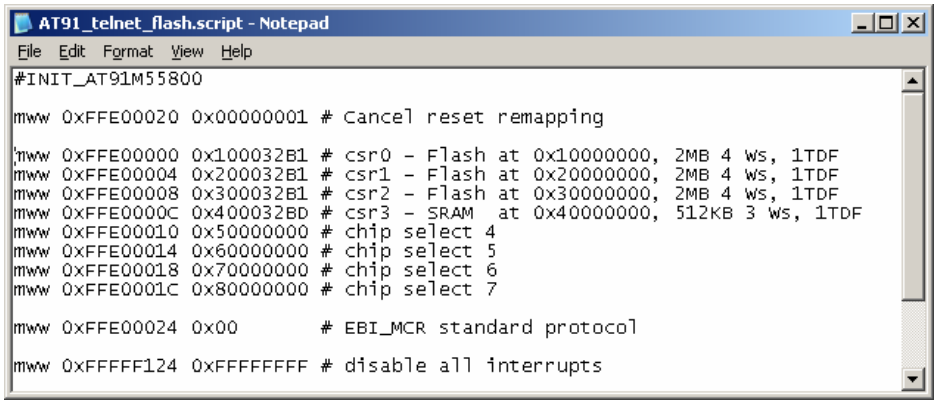


Figure 14 : OpenOCD script file

5.6.4 OpenOCD command for Flash

Once the connection with the board is established, we can start to test the communication with the three external Flashes connected to the board.

With the command "flash banks" OpenOCD show the three different banks we have configured in the configuration file. After that the command "flash probe" checks the communication with the flash with the parameters we put in the configuration file and return if the configured flash device is found.

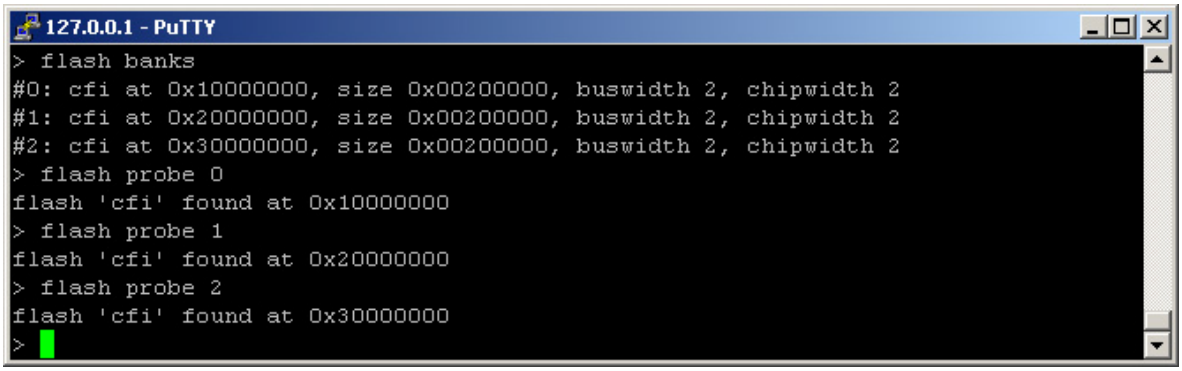


Figure 15 : OpenOCD flash probe command

The command "flash info" displays automatically the information containing in the Flash when entry in the CFI Query Mode. We can see the different sectors of the flash (34) and the status of each one (erased/not erased and protected/not protected).

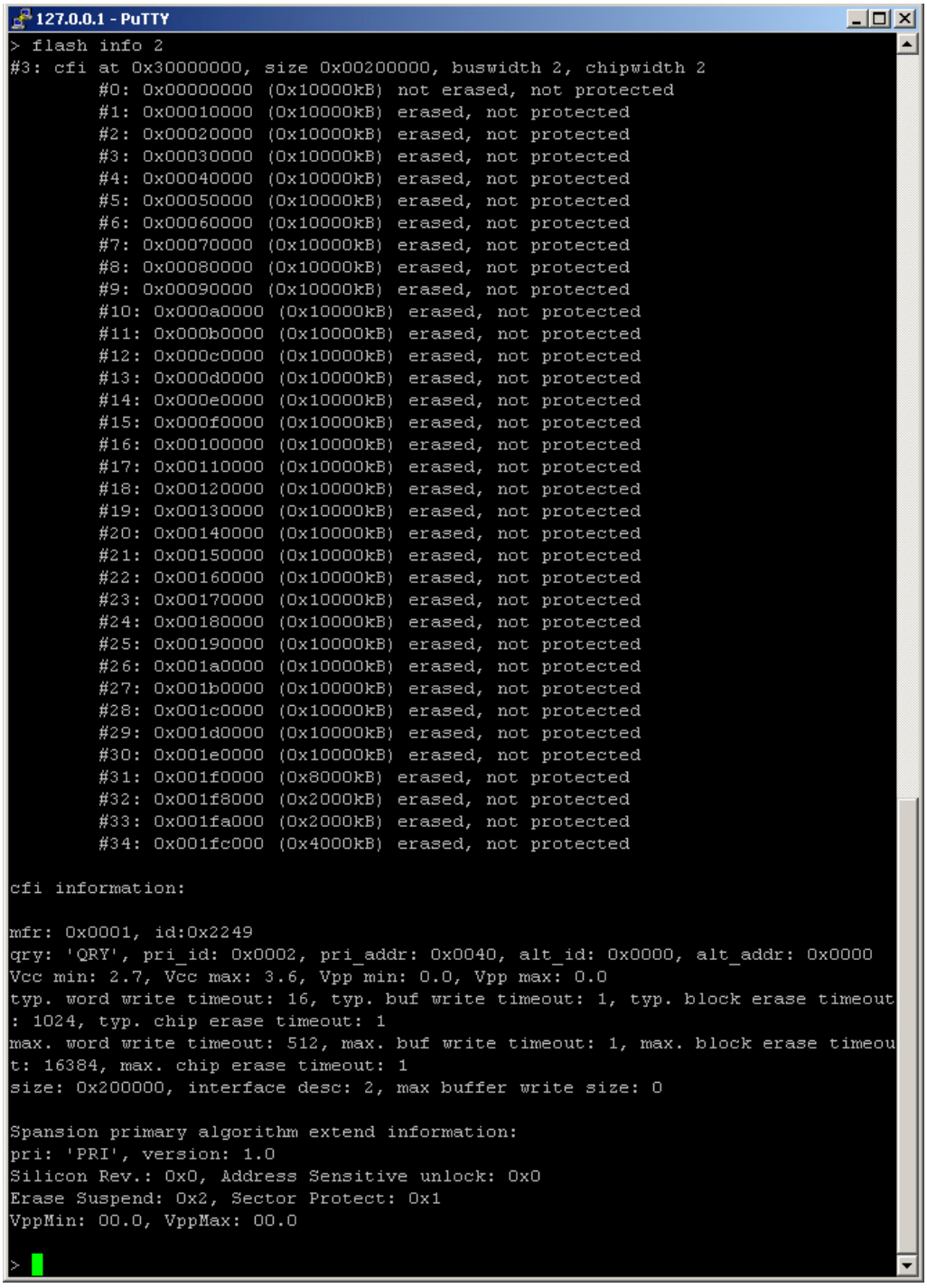


Figure 16 : OpenOCD flash info command

5.6.5 Entry in the CFI Query mode

It's possible to have also all these information manually. The device enters the CFI Query mode when the system writes the CFI Query command, 0x98, to address 0x55 in word mode, any time the device is ready to read array data.

When the device is note in the CFI Query mode, a read command shows the information of the flash. So here the flash is "erased" so we have data 0xFF

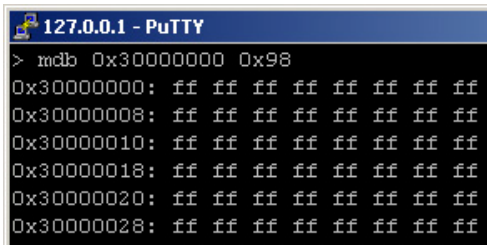
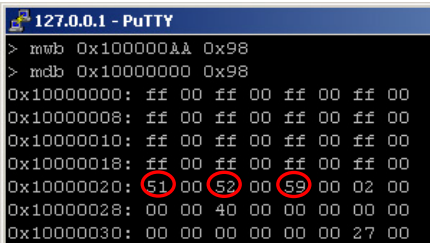


Figure 17 : Erased flash

Now when I put the command 0x98 to the address 0xAA and I read the information containing in the Flash, the device enters in the CFI Query mode and we can compare the data at different addresses with the tables containing in the datasheet of the flash Spansion.



Addresses (Byte Mode)	Data	Description
20h 22h 24h	0051h 0052h 0059h	Query Unique ASCII string "QRY"
26h 28h	0002h 0000h	Primary OEM Command Set
2Ah 2Ch	0040h 0000h	Address for Primary Extended Table
2Eh 30h	0000h 0000h	Alternate OEM Command Set (00h = none exists)
32h 34h	0000h 0000h	Address for Alternate OEM Extended Table (00h = none exists)

Figure 18 : Entry in the CFI query mode

5.6.6 Programming the flash

A specific OpenOCD command allows the user to download a binary file in the Flash at a specific offset. So I've tried to download a file in order to check this functionality.

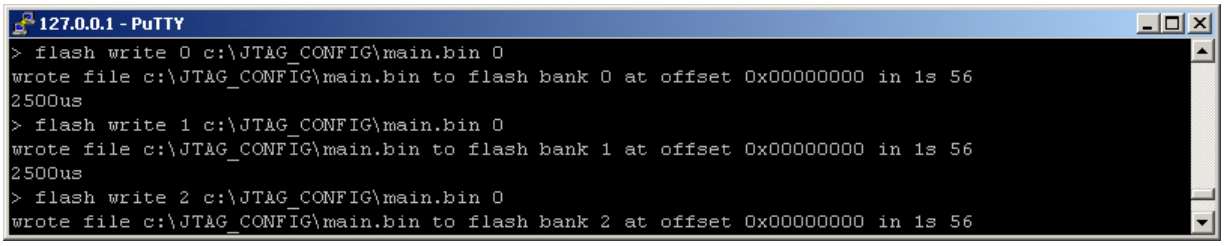


Figure 19 : Flash programming

The writing of the file on the three Flashes has been effectuated. But we don't have the confirmation that the data are the same as the source file. A writing error could have occurred. So it's important to check this point and to find a method to check that. The easier method is to dump the downloaded file in the Flash and to store these data in a file on the computer with another specific OpenOCD command. Then, we can compare the source and the destination file, to see whether they are or are not the same. This comparison task can be realized with a specific program which allows taking a file of any length and returning a fixed-size value, which is called the hash value and that's unique for each file. The MD5 algorithm returns a fixed-size value of 128 bits. A lot of free MD5 utility exists on the web.

On the following pictures, the MD5 signature of the source and the destination file are compared. The result shows that both files are the same. No data were corrupted during transfer to and from the Flash.

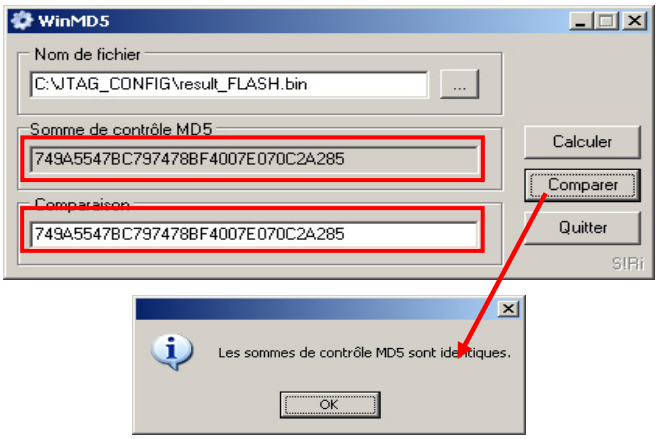


Figure 20 : MD5 test

5.6.7 Results

The three Flashes were successfully tested with the OpenOCD commands and are accessible by the AT91M55800A microcontroller and can be programmed.

5.7 Sram access

5.7.1 Objective

The goal of this test is to check the communication with the external Sram connected to the NCS3. To simplify, these accesses will be managed by OpenOCD commands, in order to make access like:

- Read, write, load and dump a binary file

5.7.2 Tests

Download a binary file into the Sram at the base address 0x40000000:

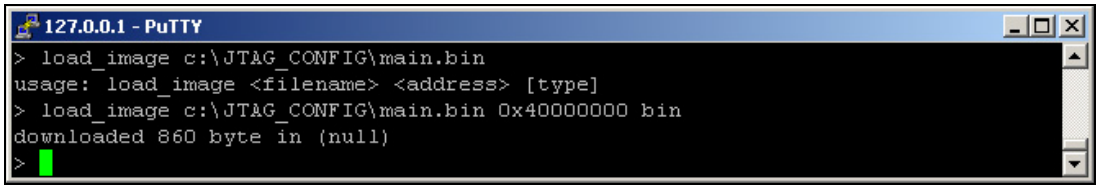


Figure 21 : Load of a binary file

Show a part of the downloaded data from the base address of the Sram:

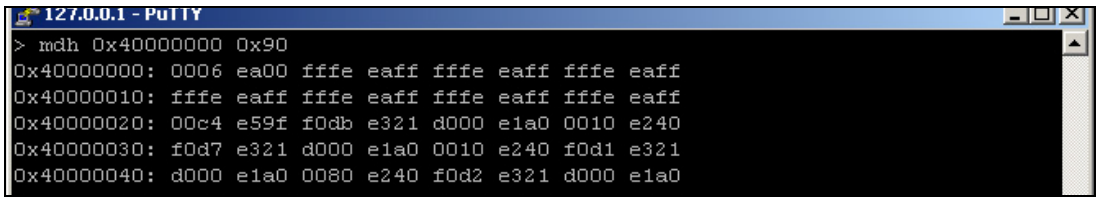


Figure 22 : Contain of the SRAM

Dump in a file on the computer, the data contains in the Sram:

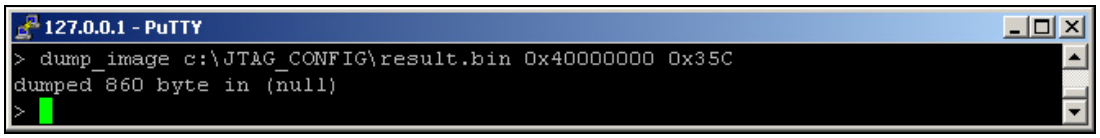


Figure 23 : Dump image from the Sram

Test of MD5 signature on the source and destination file:

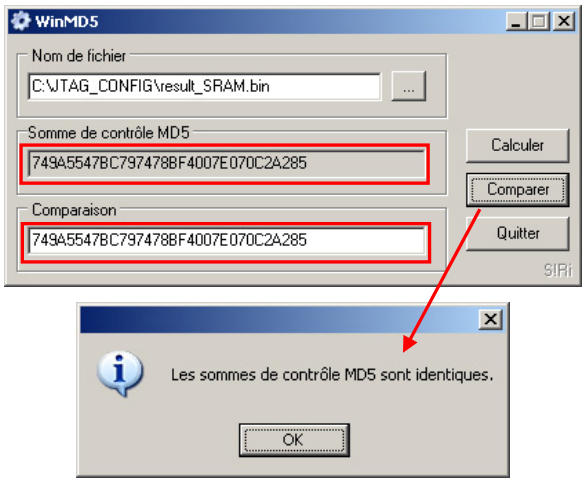


Figure 24 : MD5 test

5.7.3 Results

Different tests were executed on the Sram to prove that this external peripheral could be accessible by the microcontroller.

5.8 Shutdown – Wakeup feature

The AT91M55800A features an Advanced Power Management Controller (APMC) which optimizes both the power consumption of the device and the complete system.

With two specific pins, the system can controls the main power supply of the backup circuit of the CDMS.

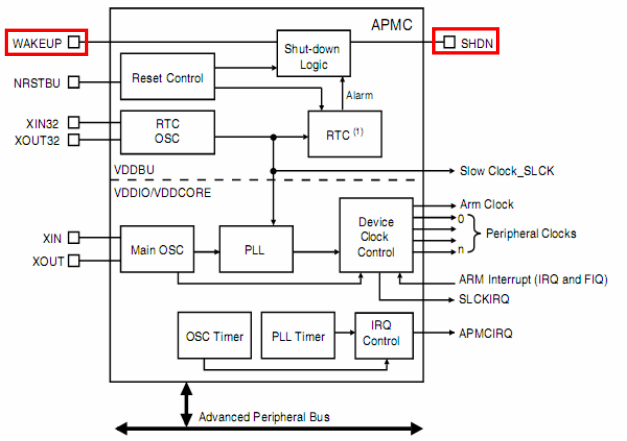


Figure 25 : APMC module

When the output **SHDN** drives at low level, it permits to power down the main power supply **VDD**, via the backup logic of the circuit. But the power supply **VDDBU** still stay on and supply the **VDDBU** part of the AT91M55800A where the RTC (Real-Time Clock) is always running and can give the time's base.

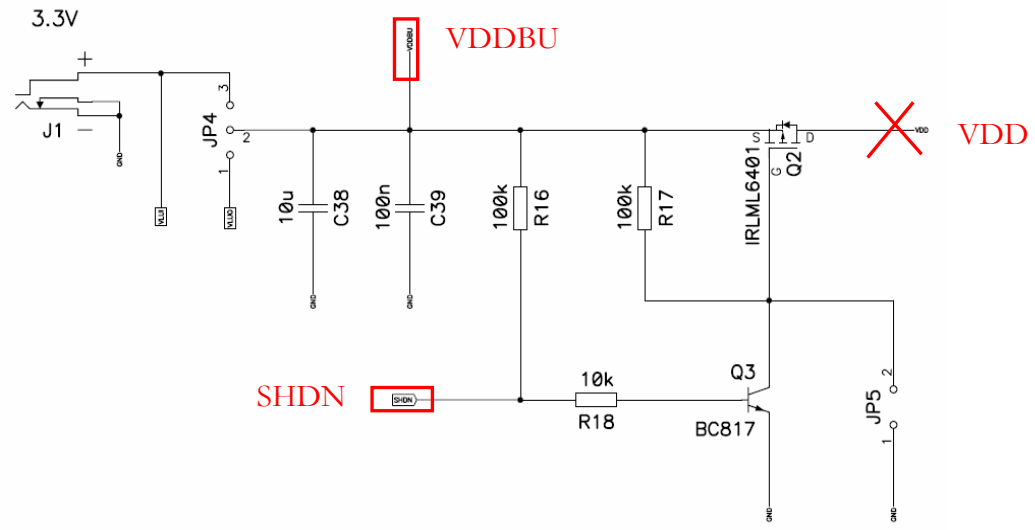


Figure 26 : Backup circuit of the CDMS

When the main power supply is powered down, a programmable edge on the input pin **WAKEUP** or an alarm on the RTC has the effect to drive the output pin **SHDN** at high level and increase the main power supply.

5.8.1 Objective

The goal of this test is to program the microcontroller to generate a low pulse on the Shut-down logic pin and to detect an edge on the wakeup pin. This test should demonstrate that the backup circuit is able to power down the main power supply until an edge is detected on the wakeup signal.

5.8.2 Configuration of the registers

In order to test the functionality of the shutdown pin, some register of the APMC user interface of the AT91M55800A must be configured.

1. The first register to set is the Power Mode Register (APMC_PMR)
   at base address 0xFFFF402C
2. Secondly, the Power Control Register (APMC_PCR) must be set
   at base address 0xFFFF4028 in order to shutdown the main power supply VDD

For further information about the configuration of these registers see:

→APMC: Chapter 12 on datasheet "AT91M55800A.pdf "

5.8.3 OpenOCD sequence

The following OpenOCD commands, permit to power down the main power supply.

Figure 27 : Shutdown sequence

To see the state of the **SHDN** pin and the result of this on the **VDD** supply. I use an oscilloscope to show the result.

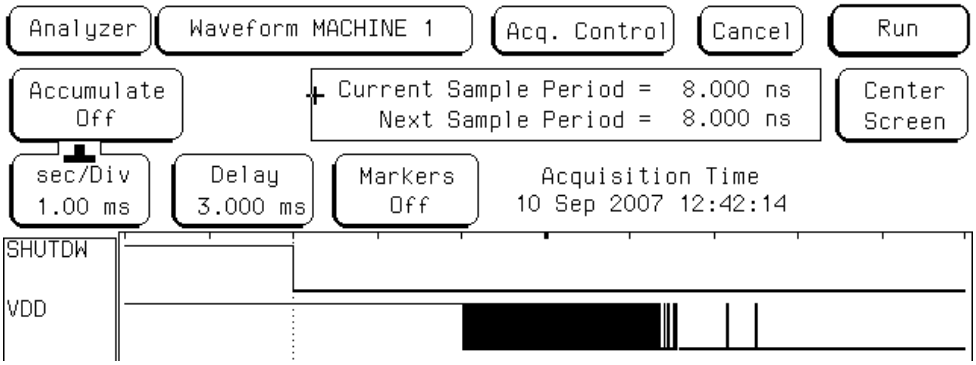


Figure 28 : Visualization of the shutdown

This capture shows the comportment of the **VDD** signal just after the falling edge of the **SHDN** pin. The functionality is properly working.

A switch is used on the CDMS board to simulate the signal coming from the EPS subsystem.

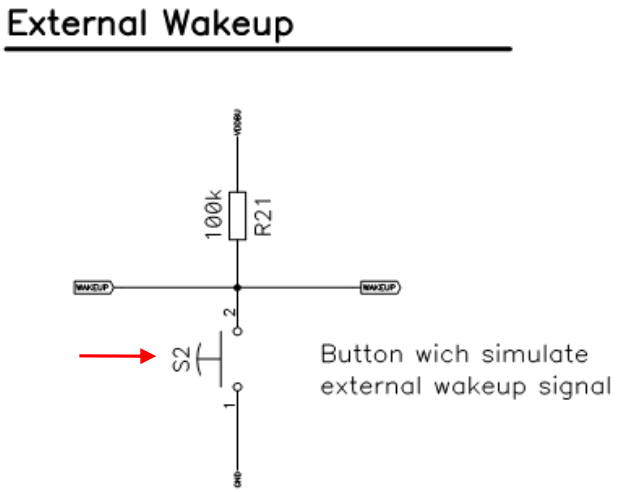


Figure 29 : Switch for the wakeup event

To see the state of the **WAKEUP** pin and the result of this on the **VDD** supply. I use an oscilloscope to show the result.

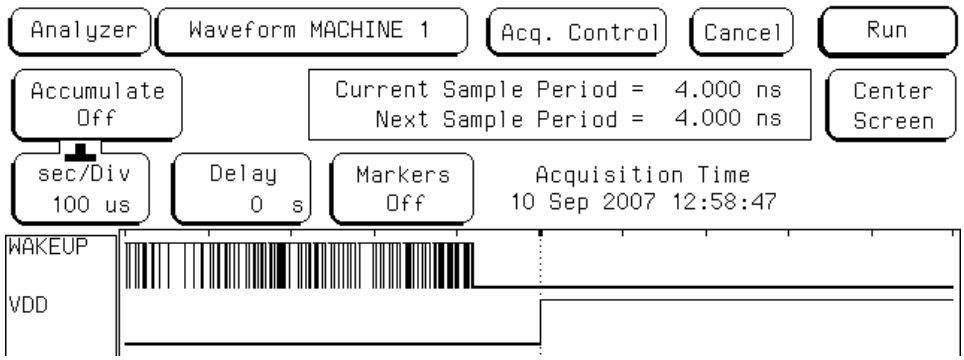


Figure 30 : Wakeup visualization

5.8.4 Results

As we can see on the precedent capture, the VDD supply rise just after the falling edge detection by the microcontroller. The functionality is properly working and satisfies the specifications. However when a wakeup occurs, a reset is performed on the microcontroller except for the part powered by VDDBU.

The datasheet of the AT91M55800A don't explain clearly the state of the microcontroller after a wakeup. The tests have clearly shown that a reset occurs and an initialization of the CDMS board is necessary to put the system back in this operating mode. The advantage of this mode is the very low power consumption, but we don't want to initialize the system each time we put the microcontroller in this mode.

Five operating modes are supported by the APMC module and offer different power consumption levels and event response latency times. With the last solution, the lower consumption mode was chosen, but it seems that it was not the most appropriate due to the problem of initialization that was found. One of the four other operating modes should be used in order to put the microcontroller in a stand-by mode.

Here are the five different operating modes of the AT91M55800A:

1. **Normal mode**

   The main power supply is switched on, the ARM Core Clock is enabled and the peripheral clocks are enabling according to the application requirements.

2. **Idle Mode**

   The main power supply is switched on, the ARM Core Clock is disabled and waiting for the next interrupt (or a main reset). The peripheral clocks are enabled according to the application requirement and the PDC transfers are still possible.

3. **Slow Clock Mode**

   Similar to normal mode, but the main oscillator and the PLL are switched off to save power. The device core and peripheral run in Slow Clock Mode. Note that Slow Clock Mode is the mode selected after the reset.

4. **Standby Mode**

   A combination of the Slow Clock Mode and the Idle Mode, which enables the processor to respond quickly to a wakeup event by keeping very low power consumption.

5. **Power-down Mode**

   The main power supply is turned off at the external power source until a programmable edge on the wake-up signal or a programmable RTC alarm occurs.

5.9 Watchdog feature

The AT91M55800A has an internal Watchdog Timer that can be used to prevent system lock-up if the software becomes trapped in a deadlock.

In normal operation, the user reloads the watchdog at regular intervals before the timer overflow occurs. If an overflow does occur, the watchdog timer generates one or a combination of signals, depending on the parameters in the Watchdog user interface.

5.9.1 Objective

The goal of this test is to generate a watchdog overflow on the **NWDOF** pin to check the following part of the circuit, to see if a **RESET** occurs.

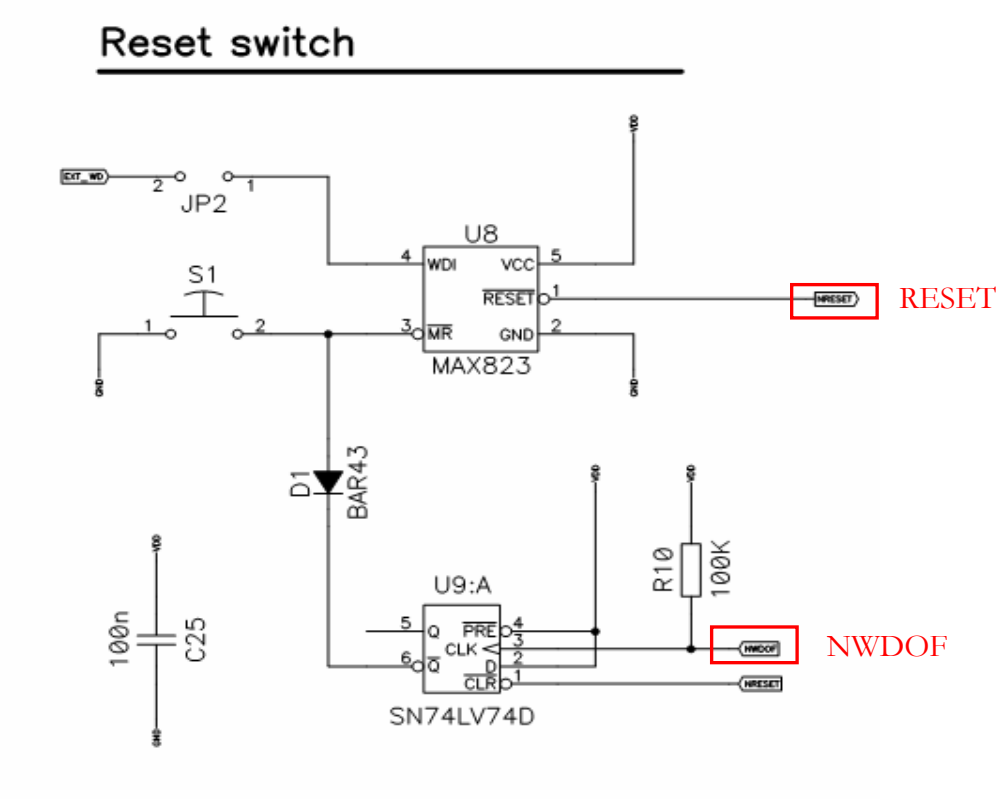


Figure 31 : Reset circuitry of the CDMS

5.9.2 Configuration and test of the Watchdog feature

Different register of the Watchdog user interface must be configured before to generate an overflow.

For further information about the configuration of these registers see:

→WD: Chapter 14 on datasheet "AT91M55800A.pdf "

The next figure shows the OpenOCD sequence in order to configure and start the Watchdog timer.

Figure 32 : OpenOCD watchdog sequence

After putting the value in the timer, the value is not reloaded and an overflow occurs. The following figure shows the comportment of the signals **NWDOF** generated by the Watchdog overflow, the input **MAX_MR** of the supervisor and finally the output of this supervisor where a reset by the pin **NRESET** is generated

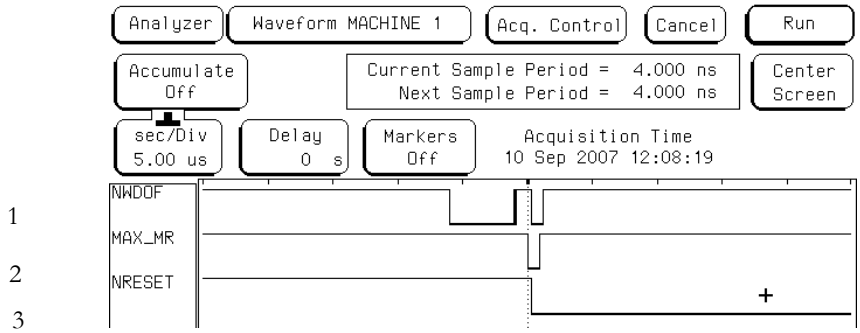


Figure 33 : Visualization of the reset

The precedent figure shows that:

1. A low pulse on the NWDOF pin is driven after the timer overflow.
2. A low pulse is detected on the input pin MR of the supervisor MAX823.
3. The supervisor MAX823 generates, at output, the reset signal for the ARM microcontroller.

5.9.3 Results

The test proves the functionality of the circuit to guarantee a reset of the CDMS board by the Watchdog of the AT91M55800A when a deadlock situation in the software happens.

5.10 Consumption test

5.10.1 Objective

For the moment we have no idea about the consumption of the CDMS Engineering Model board. The goal of this test is to determine this power consumption, to see if the values applied in the specification for the different operating mode is realistic with the reality.

5.10.2 Used material

To realize this test we use an Agilent handheld digital multimeter in serie between the power supply and the CDMS board.

Figure 34 : Material used

5.10.3 Results

The requirement for the power consumption done bye the specifications are:

- Consumption in OFF mode : no power
- Consumption in Stand By mode: 1 [mW]
- Consumption in Operational mode: average 150 [mW], peak 250 [mW]

The next table shows the power consumption of the actual CDMS Engineering Model board:

Tableau 1 : Consumption of the CDMS prototype board

	Courant	***tension***		
	[mA]	**3V**	**3.3V**	**3.6V**
		P [mW]		
Power ON after a reset (frequency 32768Hz)	17	51	56.1	61.2
Flash erase (freqeuncy 4MHz)	44	132	145.2	158.4
Stand-by (only RTC on, frequency 32768HZ)	0.428	1.284	1.4124	1.5408

These values of power consumption are calculated between the minimum and the maximum power supply and with a running frequency maximal for the AT91M55800A of 4 MHz. Naturally the power consumption depends a lot on the working frequency and increases with higher frequency.

The red values are higher as the values done by the requirement; these values were transmitted to the person who is responsible for the power budget to update them. Additional tests, more precise, will be performed on the next CDMS board.

5.11 Conclusion for the CDMS Engineering model

At this point, the prototype board of the CDMS is tested. Electrical and hardware tests were performed. The major parts of the design are working and satisfy the requirement. However, some modifications will be necessary, like the integration of the MSP430F1611 for the I2C communication, a new temperature sensor, and some others small corrections.

6 QUALIFICATION MODEL DEVELOPMENT OF THE CDMS

This phase consists in the modification of the actual design with the results of the tests done on the Engineering Model board and also with the modifications of the CDMS specifications. The goal is to deliver at the end of my diploma work, a new CDMS board, the Qualification Model. First the schematic must be reviewed in order to modify the design. Then the PCB will be build by a manufacturer and assembly by my self, with the new components of the board.

Once the board will be assembly, the same test procedures as those used for the Engineering Model board will be deployed in order to validate this new design. The Development Environment for both microcontrollers (ARM+MSP430) will be installed in order to make software to test the hardware. A PSA Analysis is planned too to check if all components present on the board never exceed fixed limits by the ECSS norm for different parameter.

6.1 New architecture of the CDMS Qualification Model board

The following schema represents the new architecture of the CDMS board for the Qualification Model. The various modifications on this architecture are described in detail in the next pages. The main modification his the integration of the MSP430F1611 for the I2C function and his SPI link with the AT91M55800A.

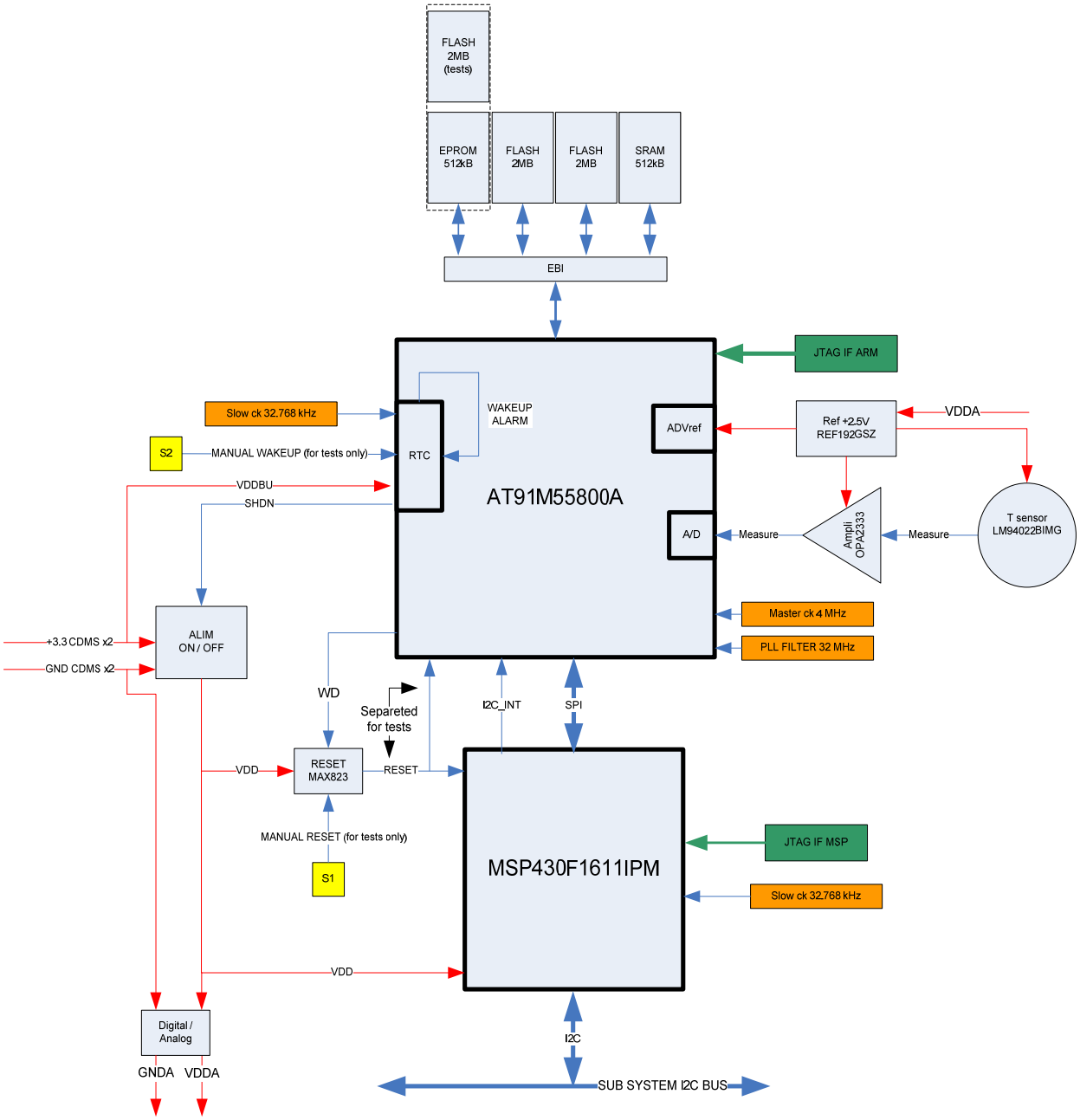


Figure 35 : Architecture of the new CDMS board

6.2 MSP430F1611

This is the main modification on the design for the Qualification Model of the CDMS board. In fact, during the test of the prototype board a problem of I2C communication was found with the component that was chosen to make the link for the I2C data between other subsystems and the AT91M55800A. Various solutions were proposed like that was explained on the first part of this report and finally the best solution seem to change the I2C Bridge by the MSP430F1611.

It's a very small microcontroller build by Texas Instrument and has the particularity to have very low power consumption. The role of this component on our board will be only to manage the I2C communication between other subsystems and the AT91M55800A. An SPI (Serial Protocol Interface) will be added for the communication between the AT91M55800A (master) and the MSP430F1611 (slave).

6.2.1 SPI interface on AT91M55800A

The AT91M55800A has one SPI module, which provides communication with external devices in master or slave mode. The SPI has four external chip selects which can be connected to up 15 devices.

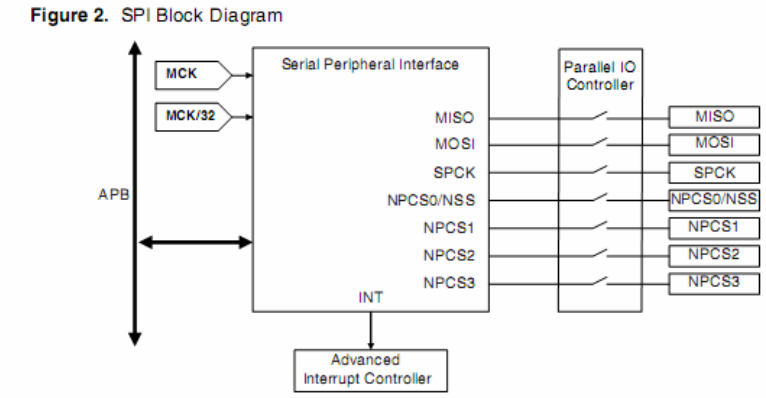


Figure 36 : SPI block diagram

The data length is programmable from 8 to 16 bit. The SPI features a full duplex (simultaneously transmit and receive) three wires synchronous transfer:

- MISO : Master In Slave Out
- MOSI : Master Out Slave In
- SPCK : SPI Clock

6.2.2 SPI interface on MSP430F1611

The MSP430F1611 have the universal synchronous/asynchronous receive/transmit (USART) peripheral interface supports two serial modes with one hardware module, in our case the SPI mode. In synchronous mode, the USART can connect the MSP430 to an external system with three or four pins. SIMO (MOSI), SOMI (MISO), UCLK (SPCK).

SPI features include:

- 7 or 8 bit data length
- 3 or 4 pin SPI operation
- Master or slave modes
- Independent transmit and receives shift registers
- Separate transmit and receive buffer registers
- Selectable UCLK polarity and phase control
- Programmable UCLK frequency in master mode
- Independent interrupt capability for receive and transmit

6.2.3 SPI connection between the AT91M55800A and the MSP430F1611

After having checked the datasheet of the MSP430 and the AT91M55800A, the conclusion is that these two microcontrollers are compatible to communicate together with an SPI interface. The ARM will be the master and the MSP430 the slave. In master mode, the SPI controls data transfers to and from the slave. To avoid the master to pool the slave every time to see if a data is coming from the I2C interface, an interrupt signal will by driven by the MSP430 to the AT91M55800A. With this solution, the AT91M55800A knows exactly when a data from the I2C interface on the MSP430 is arrived, and can initiate a transfer to get back the data.

The SPI 3 pin mode will be used to connect the two microcontrollers:

- MOSI (Master Out, Slave In)
- MISO (Master In, Slave Out)
- SPCK (Clock shared by the Master)

Normally, the master drives the chip select to the slave, but with only one slave, technically an external chip select isn't required.

6.2.4 MSP430F1611 schematic

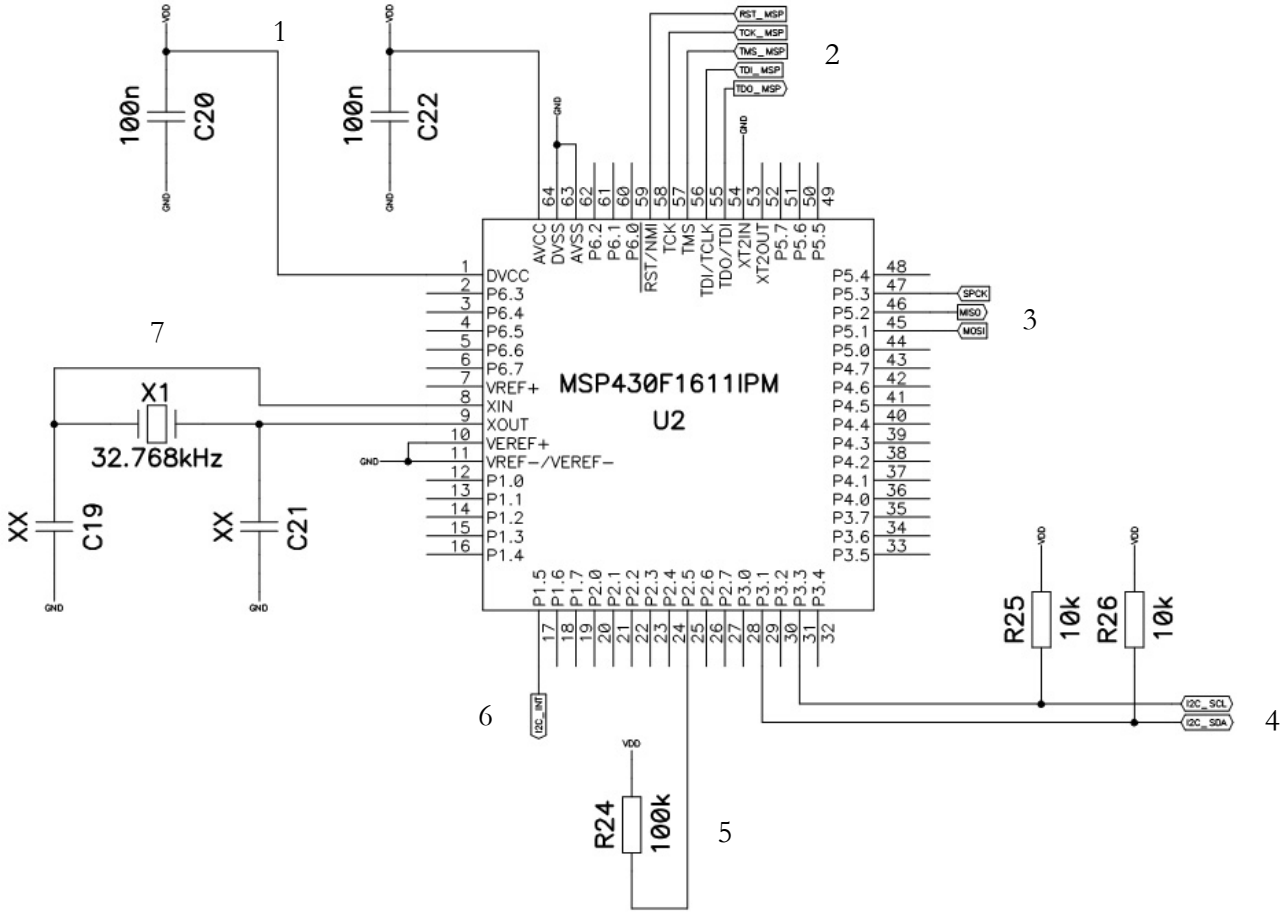


Figure 37 : MSP430F1611

1. Analog and digital power supply (VDD +3.3V)
2. JTAG signal for programming (TDI,TDO,TMS,TCK,RST)
3. SPI 3-pins mode signals (MISO, MOSI, SPCK)
4. I2C signals (SDA and SCL)
5. Rosc for the internal DCO (Digitally-Controlled Oscillator)
6. I/O use in output to generate the I2C interrupt on the ARM microcontroller
7. 32.768 kHz crystal

6.3 Voltage reference

The REF19x series is made up of micropower, low dropout voltage devices, providing stable output voltage from supplies as low as 100mV above the output voltage and consuming less than 45uA of supply current. Due to his very low temperature coefficient and his output voltage precision, this chip is used to supply all analogue parts of the CDMS circuit with a voltage of +2.5V.

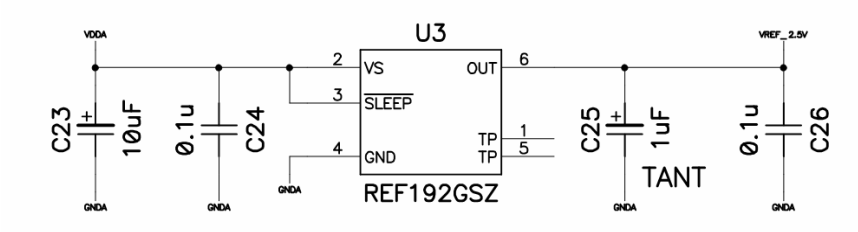


Figure 38 : REF192GSZ

6.4 Temperature sensor

The LM94022 is a precision analog output CMOS integrated circuit temperature sensor. While operating over the wide temperature range of -50°C to +150°C, it delivers an output voltage that is inversely proportional to measured temperature. The chip has a negative gain that can be chosen by putting two dedicated pins (GS0 and GS1) at GND or VCC. The chosen gain is 8.5mV/°C (GS0=0 and GS1=1). For the measuring range, the temperature accuracy is +-2.1°C. This circuit is already implemented on each others subsystems; it's why it was decided to use this one.

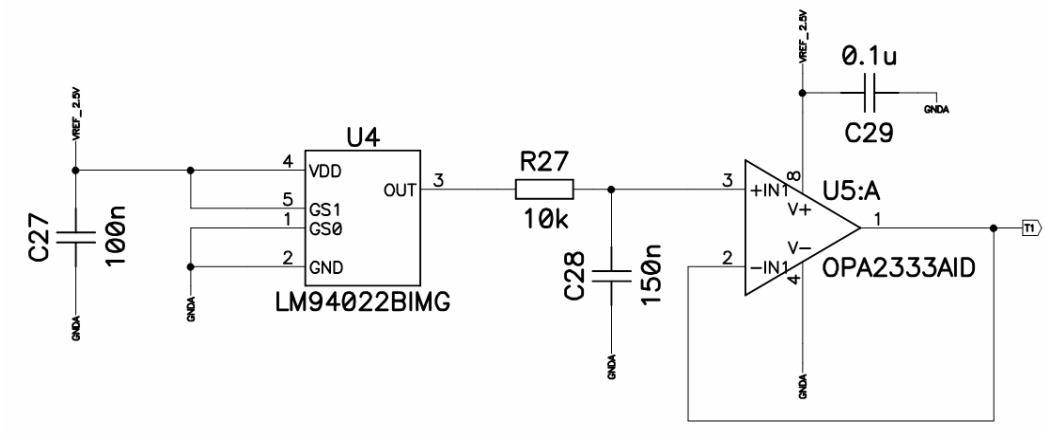


Figure 39 : LM94022BIMG + filter

Before sending the measure to the AT91M55800A microcontroller, we need to filter them due to the different disturbances of the working frequency (32768Hz to 32 MHz). A low pass filter was designed by the person working on the EPS subsystem, with a cut-off frequency of 100Hz, which is enough to cut all the disturbances described above.

The following table shows the maximum and the minimum input voltage on the A/D input of the AT91M55800A, over the wide range temperature.

$T_{measure\_min}$	-50°C	$U_{input\_ARM\_max}$	1955mV
$T_{measure\_max}$	+150°C	$U_{input\_ARM\_max}$	301mV

Tableau 2 : Voltage on A/D input

6.5 Reset logic

The MAX823 supervisor circuits combine reset output, watchdog, and manual reset input functions. It compares the supply of the board **VDD** with an internal reference and drive the reset pin low for 200ms, to ensure a correct state for the AT91M5580A microcontroller during power-up, power-down and brownout conditions. The MSP430 has an internal reset circuit that ensures a correct state, but a dedicated input pin "RST/NMI" can be also a trigger to generate a reset. So the reset signal is driven to the AT91M55800A and the MSP430F1611 microcontroller. As we can see on the schema, a switch "ON/OFF" is present in order to allow resetting independently on or the other microcontrollers. This function can be interesting during the debug phase.

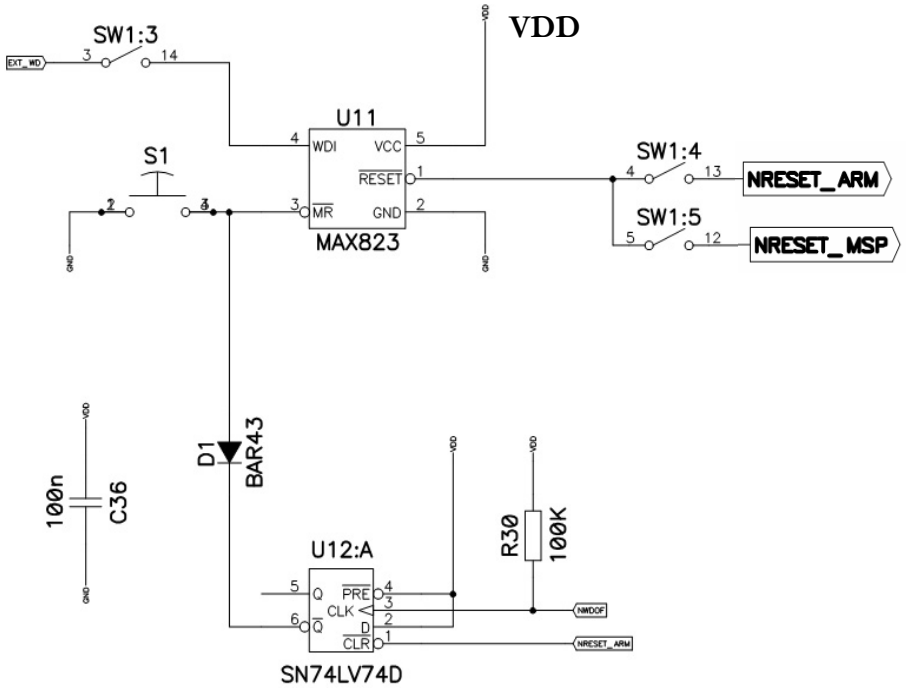


Figure 40 : Reset circuitry

These logic permits to assert the NTRST pin to initialize the TAP (Test Access Port) controller on the AT91M55800A even by the reset signal of the MAX823 or even by the reset done by the JTAG debug interface.

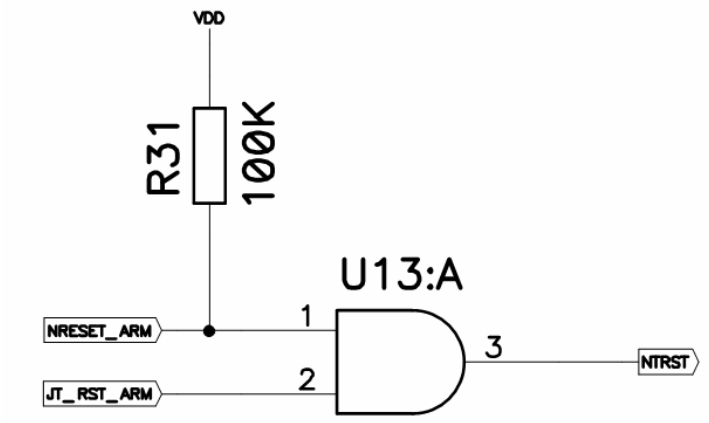


Figure 41 : Logic for ARM reset

These logic permits to assert the RST_MSP pin (MSP430 reset) even by the reset signal of the MAX823 or even by the reset done by the JTAG debug interface.

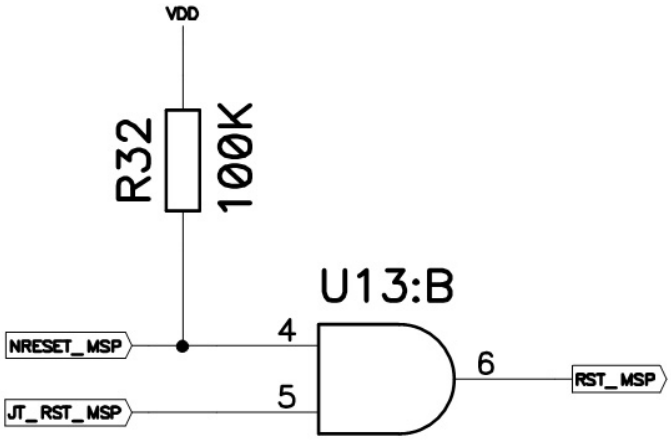


Figure 42 : Logic for MSP430 reset

6.6 Test buttons and switches

Some switches and buttons are necessary on the Qualification Model of the CDMS board. The maximal height over both sides of the PCB must not exceed 5 mm (*see figure 5, page 14*). It's why I've chosen a button and a switch model with a small height.

The CDMS Qualification Model has:

- 2 buttons
  - 1 for a manual RESET
  - 1 to simulate a manual WAKEUP event
- 1 switch with 8 ON/OFF position

The following table explains in detail the configuration of these switches with their functionalities:

NCS0 selection switch	
Selection of the FLASH	
SW_1	ON
SW_2	OFF
Selection of the EPROM	
SW_1	OFF
SW_2	ON
Caution !!	**never set SW_1 and SW_2 ON**

Reset selection during debug tests	
Reset ARM + MSP	
SW_4	ON
SW_5	ON
Reset ARM	
SW_4	ON
SW_5	OFF

External Watchdog interrupt	
Enable the Watchdog input of the MAX823	
SW_3	ON
Disable the Watchdog input of the MAX823	
SW_3	OFF

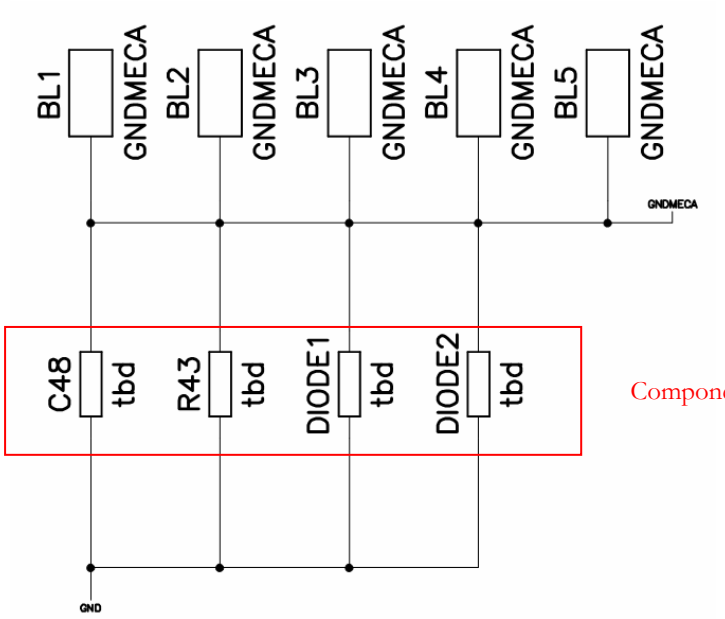
Shutdown feature	
Enable shutdown feature	
SW_6	OFF
Disable shutdown feature	
SW_6	ON

Boot select	
Simulate an external boot select signal	
SW_7	ON

Tableau 3 : Switches configuration

6.7 Filter for EMC problem

For the Qualification Model of the CDMS board, the electrical team has recommenced to ad on the layout, four footprints SMD1206. These footprints will be used to filter, if needed, the noise for each subsystem of the satellite. Some resistors, capacitors and diodes will be soldered if need. The calculation of these components will be made during the integration of the different sub-system in the satellite structure by the electrical team.



Components used for the filter

These four footprints must be the linked between the electrical ground of the satellite and the mechanical ground of the CDMS board.

The mechanical ground of the CDMS board must be realized with a plan of copper, which permit a good dissipation of the heat of the CDMS board, on the satellite structure.

This plan of copper is connected at the structure of the satellite by five hole presents on the CDMS board, where a part of copper is present and where the fixing screws will be put inside.

For more information see the mechanical design see:

→figure 5, page 5

6.8 Clocks source

6.8.1 ARM: Slow Clock

Remember, it was the problem with the prototype board. It was impossible to debug via the JTAG interface because instead of a crystal, an oscillator was present. So to correct that problem for the qualification board, a crystal was chosen.

The Slow Clock is the only clock considered permanent and is essential in the operations. **In any case, a 32768 Hz crystal must be connected to the XIN32 and XOUT32 pins** in order to ensure that the Slow Clock is present. The AT91M5580A has been especially designed to connect to a 6 pF typical load capacitance crystal and does not require any external capacitor, as it integrates the XIN32 and the XOUT32 capacitors to ground. For a higher typical load capacitance, two external capacitors must be wired as shown in the next figure.

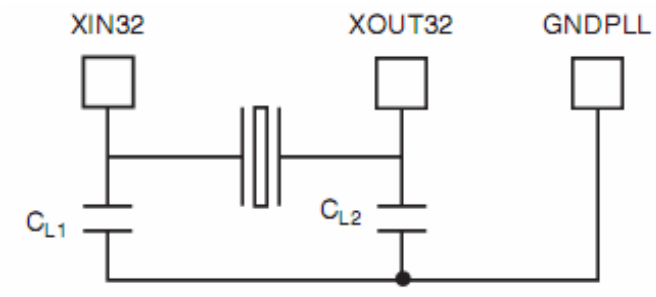


Figure 43 : Higher typical load capacitors

The model chosen is the crystal MC-146 manufactured by "Epson toyocom". It has the advantage to be delivered is a very small package, and to operate in the range of temperature of -40°C to +85°C. Is nominal frequency is 32768Hz with a load capacitance of 7pF. So in order to compensate the mismatch, two externals capacitances of 2pF must be wired as shown on the precedent pictures.

Figure 44 : MC-146 crystal

6.8.2 ARM: Main Oscillator

The Main Oscillator of the AT91M55800A is designed for a 3 to 20 MHz fundamental crystal. The Main oscillator can be bypassed, and in this case, any frequency up to the maximum specified in the electrical characteristics (>20 MHz) can be entered on the XIN pin.

In the prototype board for the Main Oscillator, an Oscillator of 4 MHz (and not a crystal) was used and connected on the XIN pin. As described on the datasheet, the Main Oscillator can be bypassed, but only with a frequency higher than 20 MHz. It was not the case with a frequency of 4 MHz. In order to respect the datasheet for the Qualification Model board, a crystal of 4 MHz was chosen and connected to the XIN and XOUT pin, like the following picture.

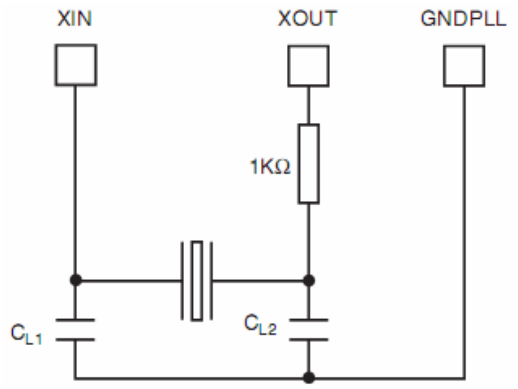


Figure 45 : Typical crystal connection of main oscillator

The 1 KΩ resistor is required for crystals with frequencies lower that 8 MHz. The oscillator contains 25pF capacitances on each XIN and XOUT pin. Consequently, CL1 and CL2 can be removed if a load capacitance of 12.5pF is used.

The model chosen is the crystal SMU4 manufactured by Jauch. It has the advantage to be built in a SMD package with a height of 4mm, because the maximum height authorized is 5 mm on each side of the PCB and a load capacitance of 7pF. So in order to compensate the mismatch, two externals capacitances of 1pF must be wired as shown on the precedent pictures.

Figure 46 : SMU-4 crystal

6.8.3 ARM: PLL

The Main Oscillator output signal feeds the phase lock loop, which aims at multiplying the frequency of its input signal by a number up to 64. So the operation on the AT91M55800A can be made up to a maximum of 33 MHz. The PLL is disabling at reset to minimize power consumption. The PLL has a dedicated PLLRC pin which must connect with an appropriate second order filter made up of one resistor and two capacitors. This filter was designed to work at a frequency of 32MHz. ATMEL provides an Excel file for calculation of the best components values for this filter.

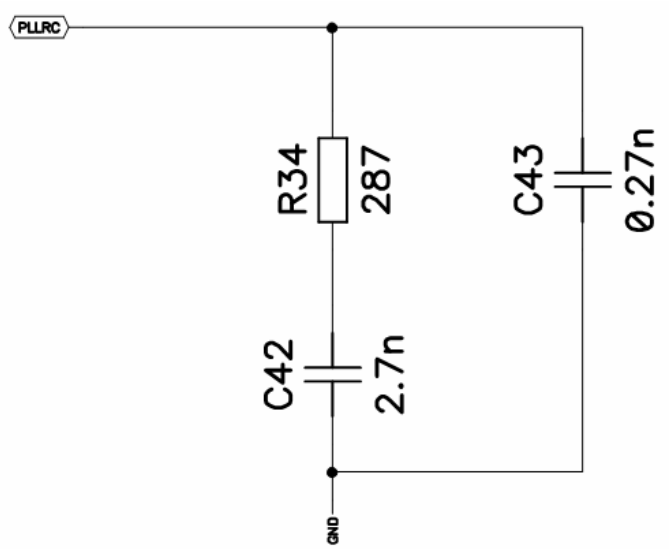


Figure 47 : PLL filter for a frequency of 32MHz

For further information about the calculation of this filter see:

→APPENDIX D

6.9 Externals connectors

For the Qualification Model board, three external connectors are present:

Power / Data connector	
Pin 1	+3.3V_CDMS
Pin 2	+3.3V_CDMS
Pin 3	GND_CDMS
Pin 4	I2C_SDA
Pin 5	I2C_SCL
Pin 6	GND_CDMS
Pin 7	NC
Pin 8	NC

JTAG programming connector for MSP	
Pin 1	GND_CDMS
Pin 2	TCK
Pin 3	TMS
Pin 4	RST/NMI
Pin 5	TDI
Pin 6	TDO

JTAG programming connector for ARM	
Pin 1	VDD
Pin 2	TMS
Pin 3	TCK
Pin 4	RST
Pin 5	TDO
Pin 6	NRST
Pin 7	TDI
Pin 8	GND

Tableau 4 : Connector of the CDMS Qualification Model

6.10 Divers

Some details were not presented in this chapter. The complete schematic is presented in:

→APPENDIX B

6.11 Bill of material

The following list represents all the necessary components to build five boards. I have had some contact with distributors to obtain some free samples. For the rest of the components, they were commanded by Farnell or Distrelec. Some capacitors, inductance and resistors were found at school.

Part name	Value	Distributor	Réf. Distributor	Price/unit	number / CDMS	QTY ordered	price/parts
74LVX08M	Gate AND	Farnell	1014093	0.918	1	5	4.59
BAR43	Diode	Farnell	9801367	0.408	1	5	2.04
BC817-25	Transistor NPN	Farnell	1081223	0.245	1	5	1.225
IRLML6401PBF	Mosfet canal P	Farnell	8660093	1.5	1	5	7.5
LM94022BIMG	T sensor	Farnell	1312600	2.25	1	5	11.25
MAX823	Supervisor	Farnell	9725784	4.8	1	5	24
REF192GSZ	Voltage REF 2.5	Farnell	9605207	6.55	1	5	32.75
SN74LV74D	D flip flop	Farnell	1085356	0.877	1	5	4.385
MC-146	Crystal 32.768kHz	Farnell	1278039	2.24	2	10	22.4
OPA2333AIDG4	Ampli	Farnell	1230455	7.95	1	5	39.75
S29AL016D90TFI02	FLASH 2MB	Farnell	12011527	6.2	3	10	62
CAPS0805	10u Tantale	Farnell	9227814	0.555	1	10	5.55
CAPS0603	150n	Farnell	3352006	0.112	1	5	0.56
CAPS0603	0.27n	Farnell	3019500	0.135	1	5	0.675
CAPS0603	2.7n	Farnell	3019688	0.112	1	5	0.56
RS0603	287	Farnell	1170732	0.067	1	5	0.335
SMU-4	Crystal 4MHz	Distrelec	644808	1.83	1	5	9.15
PUSHBUTT-KSC341	Button	Distrelec	200539	1.29	2	10	12.9
SW-NHDS-08T	Switch	Distrelec	210127	4.63	1	5	23.15
L0603	Inductance	Distrelec	351896	1.29	2	10	-
CAPS0603	100n	HEVS	-	-	31	155	-
TAJ-B	1u	HEVS	-	-	1	5	-
TAJ-C	10u	HEVS	-	-	1	5	-
RS0603	10k	HEVS	-	-	4	20	-
RS0603	100k	HEVS	-	-	33	165	-
RS0603	0R	HEVS	-	-	2	10	-
RS0603	1k	HEVS	-	-	1	5	-
RS0603	20	HEVS	-	-	1	5	-
RS0603	200k	HEVS	-	-	1	5	-
RS0603	400k (430k)	HEVS	-	-	1	5	-
Free sample							
OPA2333AIDG4	Ampli	TI	-	sample		5	
AT91M55800A-33AU	Microcontroller ARM7	Anatec	-	sample		5	
MSP430F1611IPM	Microcontroller TI	TI	-	sample		6	
R1LV0416CSB-7LI	SRAM 512kB	MSC	-	sample		5	
S29AL016D90TFI03	FLASH 2MB	Spoerle	-	sample		5	
						TOTAL	**264.77**

Tableau 5 : Command of material

6.12 PCB layout

The layout of the PCB has been made by a person from the school who his specialist in this domain. The name of the used program is P-CAD 2006 from Altium. The layout of a space product must be executed in conformity with the ECSS norms. The norm which should be respected is the ECSS-Q-70-11A. This standard defines the limits and/or the minimum requirement of parameters for the different existing kind of boards. The CDMS board is a "Rigid Multilayer Printer Board". So the corresponding requirements for this type of board were transmitted to the specialist with the mechanical design (*see figure 5, page 14*).

The layout of the new design of the CDMS board was executed in about six days.

Here are described the main requirements that were respected during the routing of the PCB:

- Conductor width
  - Internal 120um minimum
  - External 200um minimum

- Conductor spacing
  - Internal 150um minimum
  - External 300um minimum

For detailed information about the limits of approval and characteristics of finished rigid multilayer printed board see the following document:

→ ECSS-Q-70-11A page 37 to 39

To see the in detail the six layer of the CDMS Qualification Model:

→ APPENDIX C

6.13 PCB Fabrication

The fabrication of the PCB was made by the firm Euro Circuit in Hungary. There are online services on the web, which permit the command directly the PCB with the Gerber file generated by P-CAD 2006.

Here are the offers to build the PCB of the CDMS Qualification Model

Description du circuit			
Service	On Demand	**Notre réf.**	OFR012957
Votre réf.	TSTDE Cretaz D	**Nom du C.I.**	CDMS_2.0
Panneau	Non	**Dimensions du C.I.**	87.00 x 93.50
Couches	6	**Fraisage**	>= 2.0 mm
Fraisage en profondeur	NO	**Based On Order**	
Matériau de base			
Matériau de base	FR4	**Material Thick(mm)**	1.55 mm
Out StartCu(um)	17.50	**Inn StartCu(um)**	35.00
TG	Tg (130-140 °)	**Build-Up**	Standard
Empilage spécifique	Non		
Cuivre jusqu'au bords du circuit?	Non	**Trous métallisés sur les bords?**	Non
Finition et Test Electrique			
V.E. Côté Comp.	Vert	**V.E. côté soudure**	Vert
Sérigraphie C.C.	Non	**Sérigraphie C.S.**	Non
Test électrique	Oui	**Finition métallique**	Or chimique
Via Fill	Non	**Masque pelable**	Non
Contacts carbone	Non	**Logo UL**	Non
Edge Connector	Non	**Epaisseur Au connecteur PCI**	1.00
Chanfrein connecteur PCI	Non		
Traitement			
Cotés CMS	2-Côté	**SMD Pitch**	>= 0.50
Largeur de pistes/Distance d'isolation	>= 0.10 mm	**BGA**	Non
IAR	>= 0.15	**Anneau couches externes**	>= 0.13 mm
Plus petit diamètre final	>= 0.15 mm	**Nombre de trous**	< 500 per dm2
Drill	D	**Classification**	7

Délai de livraison (jours ouvrables)	Quantité	Nbr. de Panneaux	Prix unitaire (€)	Prix par panneau (€)	Offer Price (€)	Transport Cost (€)
7	5	-	215.39	-	1076.95	15.81
10	5	-	129.23	-	646.17	15.81
7	10	-	130.19	-	1301.85	16.63
10	10	-	78.11	-	781.11	16.63

Tableau 6 : Euro circuit offer

The price is very expansive due to the ECSS requirements on the values for the conductor width and spacing. These values are not supported in a standard fabrication and are only effectuated on demand by Euro-circuit.

6.14 Presentation of the CDMS Qualification Model board

On the following picture, we can see the new version of the CDMS board after I had assembled all the components. This board has a dimension of 87 x 93.5 x 1.55 [mm]. The components were chosen in order to respect the requirement for a height of maximum 5 [mm] over both sides.

On the upper face of the PCB, we find the main components like microcontrollers, memories, programming and power/data connector, also the switch and buttons used only for test.

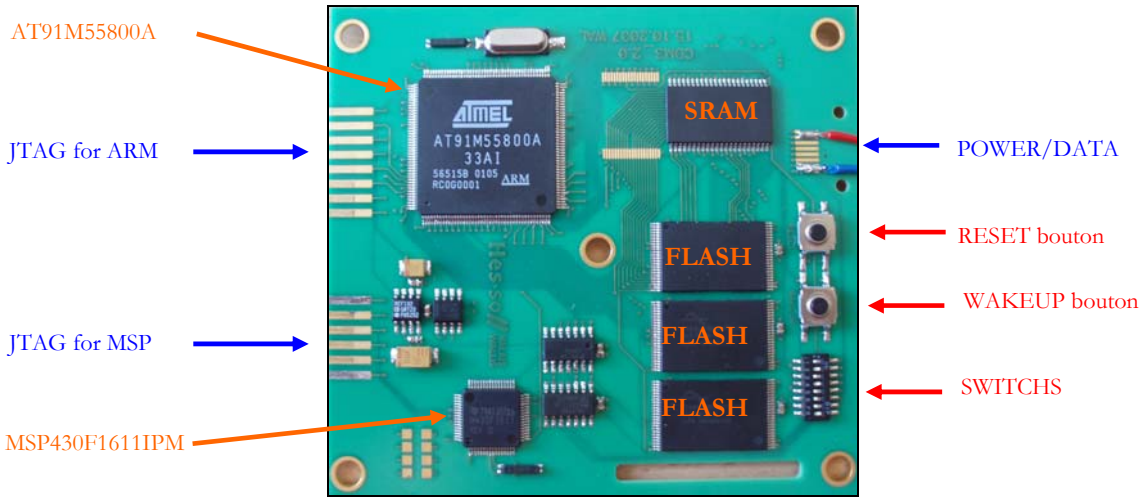


Figure 48 : Top of the PCB

On the lower face of the PCB, we find all capacitors and resistors used for the different parts of the circuit and some others components.

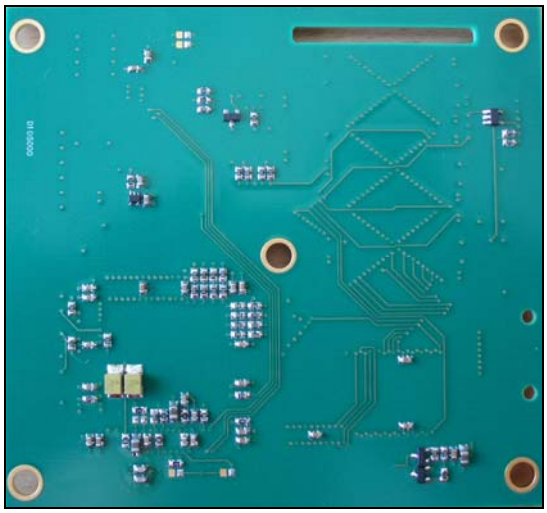


Figure 49 : Bottom of the PCB

7 SOFTWARE DEVELOPMENT

For the moment, the hardware tests were performed with OpenOCD with their different commands. To test the SPI interface between the ARM and the MSP as well as the temperature sensor, it's preferable and easier to make these tests by software. For that, it's necessary to install on the PC a cross development environment and to have an interface between the PC and the target. On the new CDMS board, we have two microcontrollers to debug via the JTAG interface, so it's necessary to have two development environments and two debug interfaces.

7.1 Development tool and environment for ARM

7.1.1 SDK4ARM

The most knows development kit for ARM microcontroller is called SDK4ARM and can be downloaded at the following address:

www.amontec.com

Amontec SDK4ARM is a complete Software Development Kit for ARM processors based on GNU tool (full sources).

Amontec SDK4ARM includes:

- **A IDE Viewer :** (Integrated development environment)
  Based on Eclipse platform/Embedded
  Zylin plug-ins
- **A Compiler :** GNU ARM GCC
- **A Debugger :** GNU ARM GDB
- **A JTAG server :** OpenOCD JTAG server

A great tutorial who explain in detail the installation and the using of this free Development kit are available in the zip file that can be found at the following address:

http://www.atmel.com/dyn/resources/prod_documents/atmel_tutorial_source.zip

7.1.2 Plug-in for Eclipse

Finally I used the Eclipse environment with a specific plug-in that was developed in our school which permits to debug an ARM microcontroller too. It has the advantage to be easier in usage and to generate automatically the makefile for the compilation of the different files of the project.

So we need the following program:

- Eclipse IDE with the specific plug-in (contain in the CDROM with the report)
- OpenOCD with the driver for the JTAGkey interface (www.yagarto.de)
- Yagarto GNU ARM toolchain (www.yagarto.de)

7.1.3 JTAG interface used

Figure 50 : Amontec JTAGkey

To connect the ARM microcontroller to the PC we used the Amontec JTAGkey through the USB port.

This interface can be bought on the website on Amontec at the following address:

www.amontec.com

The driver for this interface is included in OpenOCD

To create projects with Eclipse to debug the AT91M55800A see:

→APPENDIX F

7.2 Development tool and environment for MSP

7.2.1 Code Composer Essentials

Code Composer Essentials, is a fully integrated development environment, especially for the family of the ultra low power microcontroller of Texas Instruments MSP430. Code Composer Essentials is based on the Eclipse platform.

The free version of the TI Code Composer Essentials Evaluation v2.04 can be downloaded at the following address:

http://focus.ti.com/docs/toolsw/folders/print/msp-cce430.html

7.2.2 MSP-FET430PIF

The MSP-FET430PIF is a parallel port debug interface that is used to program and debug the MSP430F1611IPM through the JTAG interface. The entire drivers needed for this tool are include and automatically installed with Code Composer Essentials.

Figure 51 : MSP-FET430PIF

For further information about the use of the MSP-FET430PIF with Code Composer Essentials download these PDF at the following address:

http://www.ti.com/litv/pdf/slau157d

http://www.ti.com/lit/pdf/slau048

To create projects with Code Composer Essentials to debug the MSP430F1611 see:

→APPENDIX G

7.3 JTAG connectors

The programming JTAG connectors chosen by the electrical team is a special connector that is fixed on the PCB like shown on the following pictures.

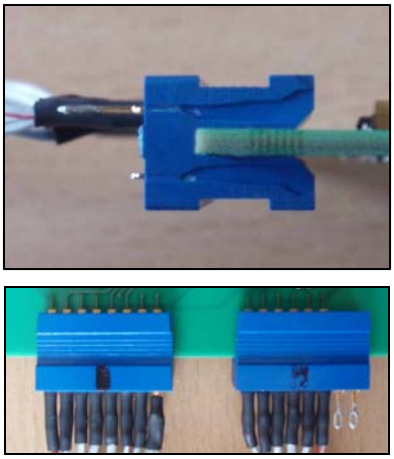


Figure 52 : View of the JTAG programming connectors

These connectors will be used for the programming of the CDMS board when the satellite will be entirely assembled. The access to the board will be easier. However this type of connector isn't the same that the standard connector delivered with the debug interface (JTAGkey + MSP-FET430PIF). In fact, the cable between these interfaces and the target had to be modified and soldered to the new connectors.

For further information about the soldering of these connectors see:

→APPENDIX E

8 FUNCTIONAL TEST OF THE CDMS QUALIFICATION MODEL

At this point, the PCB was assembled with the different components. Some electrical and software tests will be made in order to validate the new design of the CDMS board. These tests will prove that the CDMS board respects the last requirements in order to satisfy the goal of the mission. The most of these tests will be the same as those done on the prototype board, because the major design is the same.

The main task will be to implement software tests with a development environment, in order to check that the SPI interface between the ARM and the MSP430 is working including the temperature sensor.

All of these tests have been made in the way to respect a methodology which permits to check step by step the working as well as the problem of the board.

8.1 Shortcut test

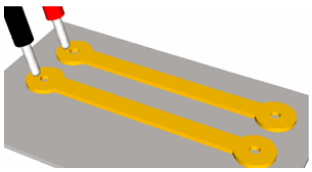
8.1.1 Objective

The goal of this test is to detect a possible shortcut problem between VDD and GND on the PCB wires, before applying power to the board.

A short cut is an accidental link between two connectors. This problem prevents the good circulation of the current.

8.1.2 Shortcut definition and used material

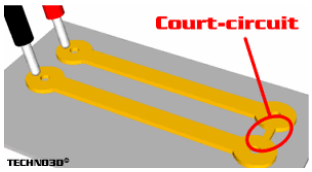
To realize this test we use an Agilent handheld digital multimeter in beep mode. When it has a contact between both probes "shortcut", the multimeter emits a beep sound.



Test OK, no shortcut

In this case there is no link between wire (VDD and GND).

The multimeter doesn't emit a sound.



Test not OK, shortcut

In this case there is a link between wires (VDD and GND).

The multimeter emits a sound.

8.1.3 Test of shortcut

To realize this test on the board and to check a potential shortcut between VDD and GND line, I take the VDD and GND reference on the input power connector. It's enough because we only have one VDD and GND reference on the PCB.

8.1.4 Results

The result of this test was positive. There is no shortcut problem on the PCB for the electrical power supply between VDD and GND.

8.2 Power on of the CDMS board

8.2.1 Objective

This second test consists to supply the board with a tension of +3.3V, then to check the power supply of all components present on the board to ensure they are properly working.

8.2.2 Material used

Figure 53 : laboratory power supply from TTi

8.2.3 Power supply connector

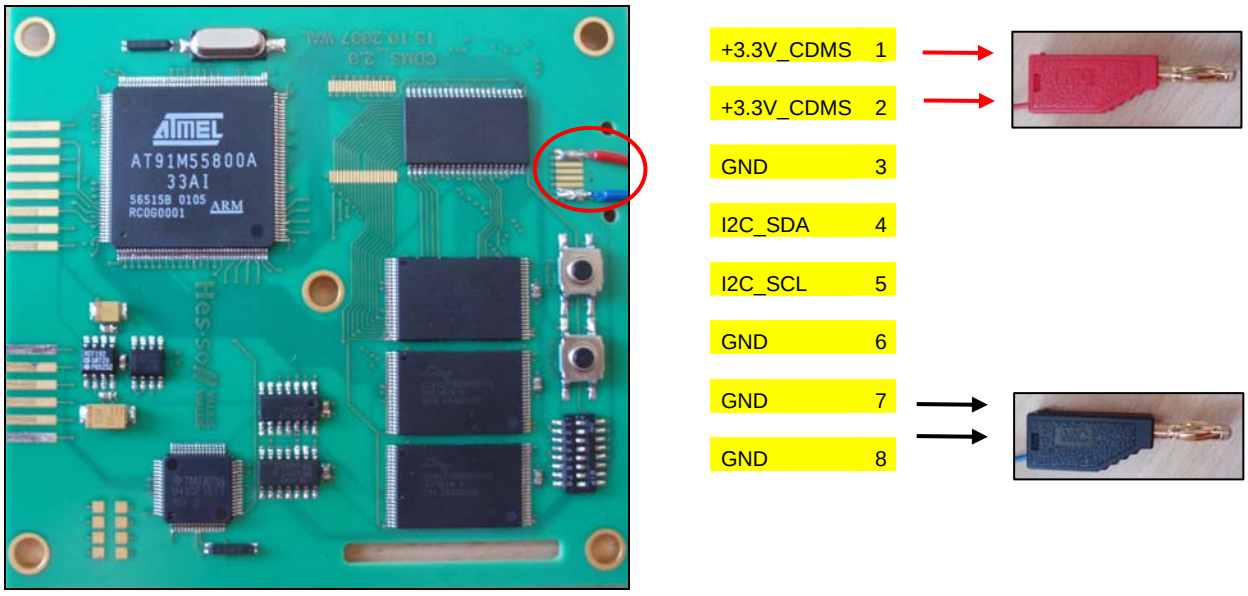


Figure 54 : CDMS board

To supply the board, it was necessary to solder two power connectors on the power/data pads.

8.2.4 Test of measure of tension

The value of the tension is measured with the Agilent multimeter. The tension of +3.3V is checked on the following points:

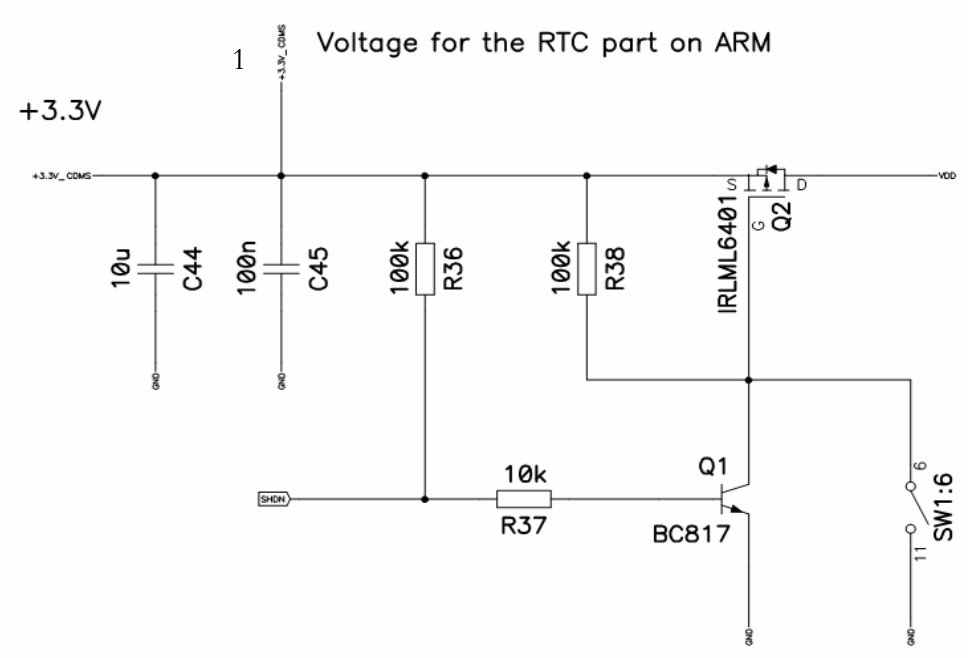


Figure 55 : Backup circuitry of the CDMS

1. +3.3V_CDMS is the alimentation of the APMC and RTC of the microcontroller AT91M55800A
2. VDD is the alimentation of the rest of the board

8.2.5 Results

A measure with the Agilent multimeter at both points VDDBU and VDD show that we have a tension of +3.3V at these circuit points.

8.3 Power check of all components

8.3.1 Objective

Having checked that the supply of the board is correct, now it's necessary to check the power supply of all components present on the board to assure their good functioning for the next tests.

8.3.2 Material used

Figure 56 : Agilent multimeter

8.3.3 AT91M55800A tension check

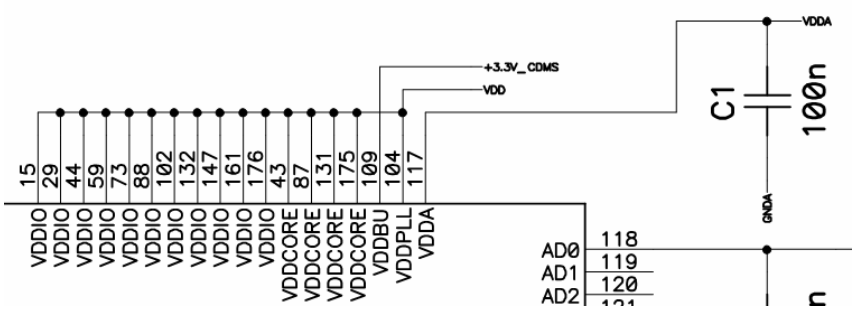


Figure 57 : Supply of the AT91M55800A

Symbol	Parameter	Value [V]	Result
VDDA	DC Supply Backup Battery	3.3	ok
VDDPLL	DC Supply Core	3.3	ok
VDDBU	DC Supply Oscillator and PLL	3.3	ok
VDDCORE	DC Supply Analog I/Os	3.3	ok
VDDIO	DC Supply Digital I/Os	3.3	ok
ADVREF	Voltage ref for AD part	2.5	ok
DAVREF	Voltage ref for DA part	2.5	ok

Tableau 7 : Measure of tension of the AT91M55800A

8.3.4 MSP430F1611 tension check

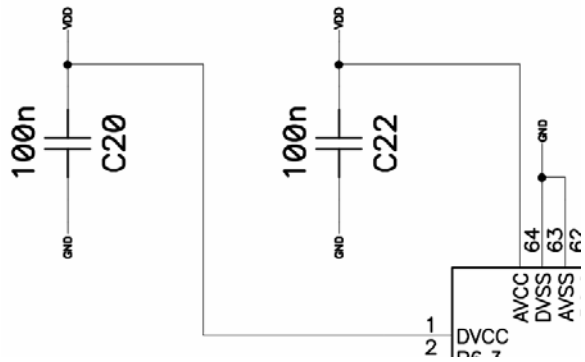


Tableau 8 : Supply of the MSP430F1611

Symbol	Parameter	Value [V]	Result
AVCC	DC Supply Analog	3.3	ok
DVCC	DC Supply Digital	3.3	ok

Tableau 9 : Measure of tension on the MSP430F1611

8.3.5 Memories tension check

Component ID	Component name	Symbol	Value [V]	Result
U6	Sram R1LV0416CSB	VDD	3.3	ok
U7	Flash S29AL016D	VDD	3.3	ok
U9	Flash S29AL016D	VDD	3.3	ok
U10	Flash S29AL016D	VDD	3.3	ok

Tableau 10 : Measure of tension on the memories

8.3.6 Reset switch tension check

Component ID	Component name	Symbol	Value [V]	Result
U11	MAX823	VDD	3.3	ok

Tableau 11 : Measure of tension on the supervisor

8.3.7 Dual-D Flip Flop tension check

Component ID	Component name	Symbol	Value [V]	Result
U12	SN74LV74D	VDD	3.3	ok

Tableau 12 : Measure of tension on the Dual-D Flip Flop

8.3.8 AND gates tension check

Component ID	Component name	Symbol	Value [V]	Result
U13	74LVX08M	VDD	3.3	ok

Tableau 13 : Measure of tension on the AND gate

8.3.9 Temperature sensor tension check

Component ID	Component name	Symbol	Value [V]	Result
U4	LM94022BIMG	VREF_2.5V	2.5	ok

Tableau 14 : Measure of tension on the temperature sensor

8.3.10 Operational amplifiers tension check

Component ID	Component name	Symbol	Value [V]	Result
U5	OPA2333AID	VREF_2.5	2.5	ok

Tableau 15 : Measure of tension on the operational amplifier

8.3.11 Reference Voltage

Component ID	Component name	Symbol	Value [V]	Result
U3	REF192GSZ	VDDA	3.3	ok

Tableau 16 : Measure of tension on the reference voltage

8.3.12 Results

All components present on the board are correctly supplied. There are no problems with the PCB or with the solder of the components

8.4 Clock signals check

8.4.1 Objective

The goal of this test is to check that the Slow Clock of 32768Hz is present. This clock is the most important for the AT91M55800A. When the microcontroller is powered on, or after a reset, this clock must be present in order to ensure correct operations. The Main Oscillator (4MHz) and the PLL (32MHz) must be activated by software after the reset. So these tests will be executed later.

8.4.2 Material used

Figure 58 : Agilent oscilloscope

8.4.3 Test of Slow Clock at 32.768 KHz on AT91M55800A

On the next picture we can show the signal at the input pin XIN32 of the microcontroller. On the red box we have the confirmation about the running frequency.

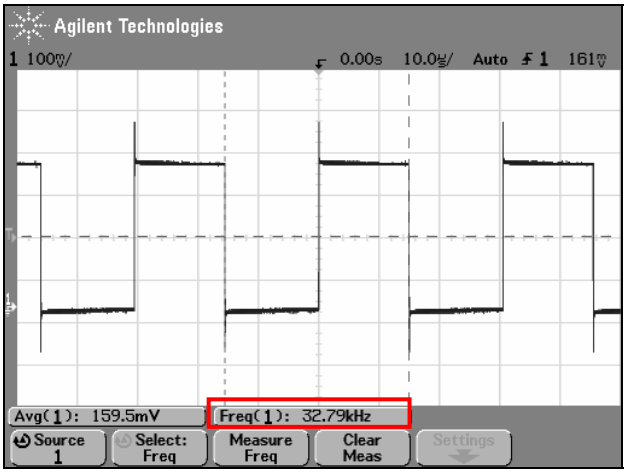


Figure 59 : Slow clock on the MCKO pin

8.5 JTAG communication test with AT91M55800A

After making some electrical tests to ensure that the CDMS board and their components are properly working, we can try to establish a JTAG communication through a telnet window in order to launch the OpenOCD server with the AT91M55800A microcontroller.

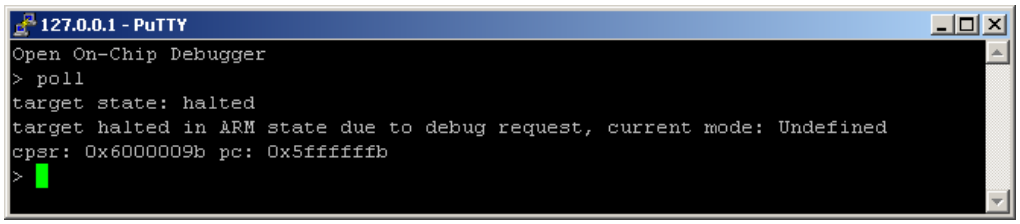


Figure 60 : JTAG communication test

The JTAG connection with the AT91M55800A is working.

8.6 Peripherals access and features tests

Like for the Engineering Model board of the CDMS, it's necessary to check the different access on the external peripheral (Flash, Sram) and features like reset, watchdog, shutdown and wakeup. These different test procedures are the same as those executed for the Engineering Model board and were already described in detail from the chapter 5.3 until the chapter 5.9.

Only the list and the results of these tests are described in the next table:

Sram	**Chapter 5.6 page 23**	
Read		ok
Write		ok
Load a binary file		ok
Dump a binary file		ok
MD5 signature test		ok
Flashes	**Chapter 5.7 page 28**	
Flash probe		ok
Flash info		ok
Flash erase		ok
Load a binary file		ok
Dump a binary file		ok
MD5 signature test		ok
Functions on the APMC	**Chapter 5.8 page 30**	
Shutdown		ok
Wakeup		ok
Watchdog Timer	**Chapter 5.9 page 35**	
Watchdog overflow		ok

Tableau 17 : List of the executed test on the new CDMS board

8.7 Test of the SPI interface between the ARM and the MSP

For testing the SPI interface between the ARM and the MSP, both microcontrollers must be programmed with their proper development environment with the C language.

8.7.1 Objective

The goal of this test is to make software code to test the SPI interface between the ARM and the MSP430, in order to assure that both microcontrollers could communicate, and also that the data coming from the I2C interface of the MSP can be transmitted by the SPI interface and also in the opposite direction.

8.7.2 SPI configuration

The ARM will be the master and the MSP the slave. The SPI 3 pin mode will be used.

- MOSI (Master Out Slave In)
- MISO (Master In Slave Out)
- SPCK (Clock signal driven by the master)

The chip select line normally driven by the master will not be used because we only have one slave device connected to the master.

The ARM has an interrupt line (IRQ0) driven by the MSP. This line is used by the MSP (Slave) to advise the ARM (Master) that a data must be recuperated, and also to avoid the master to poll every time the slave, to see if a data coming from the I2C interface, must be transferred to the ARM via the SPI interface.

The SPI bus is a synchronous link which operates in a full duplex mode (simultaneously transmit / receive). The master initiates always all transfers.

8.7.3 Test of the SPI interface

For detailed information about the SPI configuration see:

→See APPENDIX H for the ARM software code

→See APPENDIX I for the MSP software code

Two transfers of one byte were executed with the following sequence.

1. The MSP put in his SPI transmit buffer the value 0x11
2. The MSP generates an interruption for the ARM
3. The ARM detects the interruption and jumps to the IRQ routine to clear the corresponding interruption flag
4. The ARM initiates a SPI transfer with the value 0x77 in his transmit buffer
5. An interruption is generated when the data is received on the MSP, the received data is stored and a new value 0xAA is put in his TX buffer
6. The ARM waits the flag that indicates the end of the reception of the data in his RX buffer for the first transfer.
7. The ARM initiates a second transfer with the value 0x00
8. An interruption is generated when the data is received on the MSP, the received data is stored.

The following picture shows the first transfer initiated by the ARM and the value of the **MISO** and **MOSI** signals.

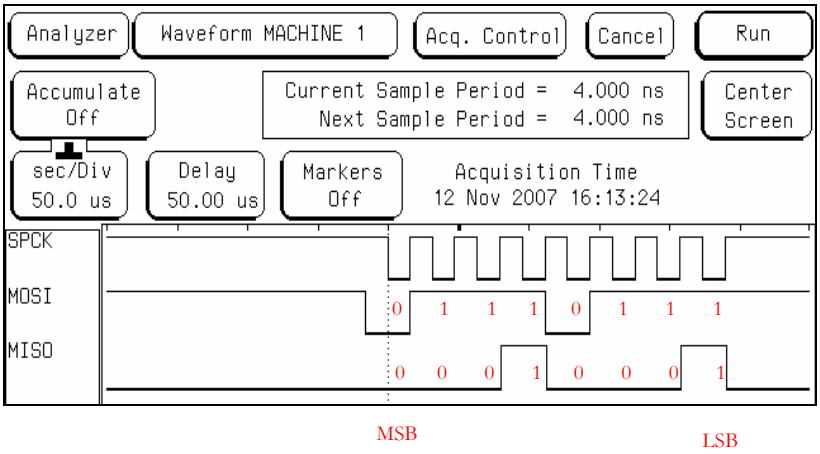


Figure 61 : First SPI transfer

The first bit to be transferred is the MSB then followed by the next bits until the LSB. The value 0x77 on the **MOSI** signal and the value 0x11 on the **MISO** signal have been transferred.

The following picture shows the second transfer initiated by the ARM and the value of the **MISO** and **MOSI** signals.

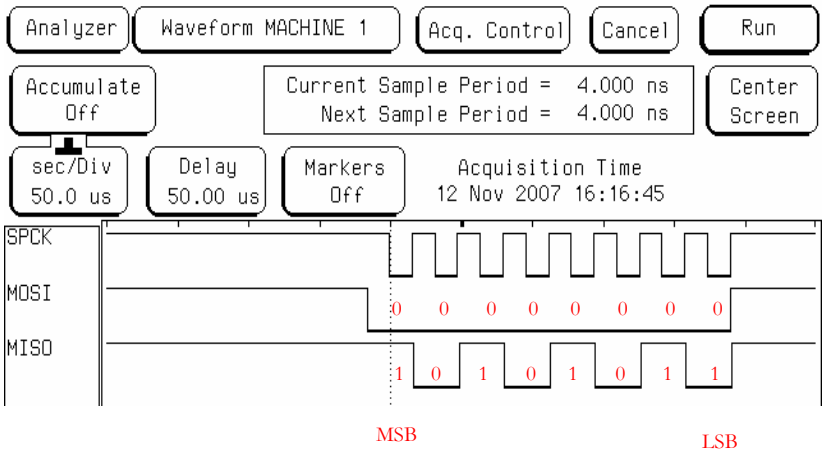


Figure 62 : Second SPI transfer

The first bit to be transferred is the MSB then followed by the next bits until the LSB. The value 0x00 on the **MOSI** signal and the value 0xAA on the **MISO** signal have been transferred.

8.7.4 Results

The SPI interface between the ARM and the MSP are working properly. When a data is coming from the I2C interface on the MSP430F1611, the following sequence is executed:

1. The MSP430 put the incoming data in the transmit buffer of his SPI module
2. The MSP430 generate an interrupt on the input IRQ0 of the ARM
3. The ARM know that a I2C data is waiting on the MSP430
4. He initiate a SPI transfer and receive the data

It's a very good solution and its work. So now the software team must write her proper protocol with these hardware capabilities.

8.8 Test of the I2C interface on the MSP430F1611

8.8.1 I2C module on the MSP430F1611

The I2C communication with the others subsystems will be managed by the MSP430. This mode is supported by the universal synchronous/asynchronous receive/transmit (USART) peripheral interface. The MSP430 has two USART modules (USART0 and USART1).The module who supports the I2C mode is the USART0.

The I2C module provides an interface between the MSP430 and other I2C compatible devices connected by way of the two wire serial bus (SDA and SCL).

The I2C module has the following features:

- Compliance to the Philips Semiconductor I2C specification v2.1
  - Byte / word format transfer
  - 7 or 10 bit device addressing modes
  - General call
  - START/STOP/RESTART
  - Multi-master transmitter / slave receiver mode
  - Multi-master receiver / slave transmitter mode
  - Combined master transmit/receive and receive/transmit mode
  - Standard mode up to 100kbps and fast mode up to 400kbps support
- Built-in FIFO for buffered read and write
- Programmable clock generation
- 16 bits wide data access to maximize bus throughput
- Automatic data byte counting
- Slave receiver START detection for auto-wake up from modes
- Interrupt capability

On all subsystems, the I2C protocol is realized with the MSP430, so there are no problems of compatibility.

8.8.2 I2C serial data

One clock pulse is generated by the master device for each data bit transferred. The I2C module operates with byte data. Data is transferred most significant bit first.

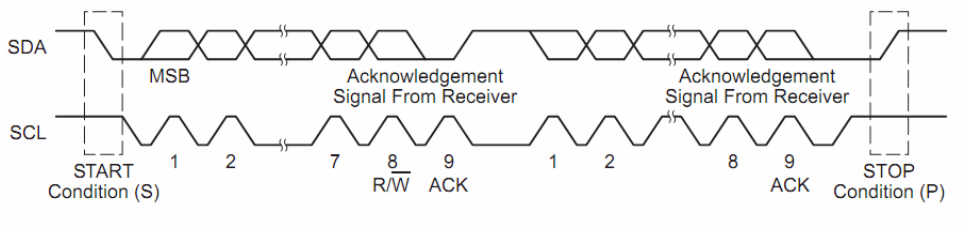


Figure 63 : Data transfer on the I2C bus

START and STOP conditions are generated by the master:

- START condition
  - High to low transition on the SDA line while SCL is high
- STOP condition
  - Low to high transition on the SDA line while SCL is high

The first byte after a START condition consists of a 7 bits slave address and the R/W bit.

- R/W =0
  - The master transmits data to a slave
- R/W=1
  - The master receives data from a slave. The ACK bit is sent from the receiver after each byte on the 9th SCL clock.

Data transfer with acknowledgement is obligatory. The acknowledge-related clock pulse is generated by the master. The acknowledge bit is sampled during the 9th clock pulse.

- ACK = 1
  - The receiver doesn't acknowledges
- ACK = 0
  - The receiver acknowledges

8.8.3 I2C test

For detailed information about the I2C configuration see:

→See APPENDIX I for the MSP software code

In order to check a transmission on the I2C interface, I have created a project with Code Composer Essential. This project just configures the I2C module of the MSP430 to be in master mode, send a frame with a START condition followed by the address of the slave, the acknowledgement of the slave and a STOP condition.

The see the bit of the transfer, the only possibility is to use an oscilloscope connected to the SCL and SDA pins.

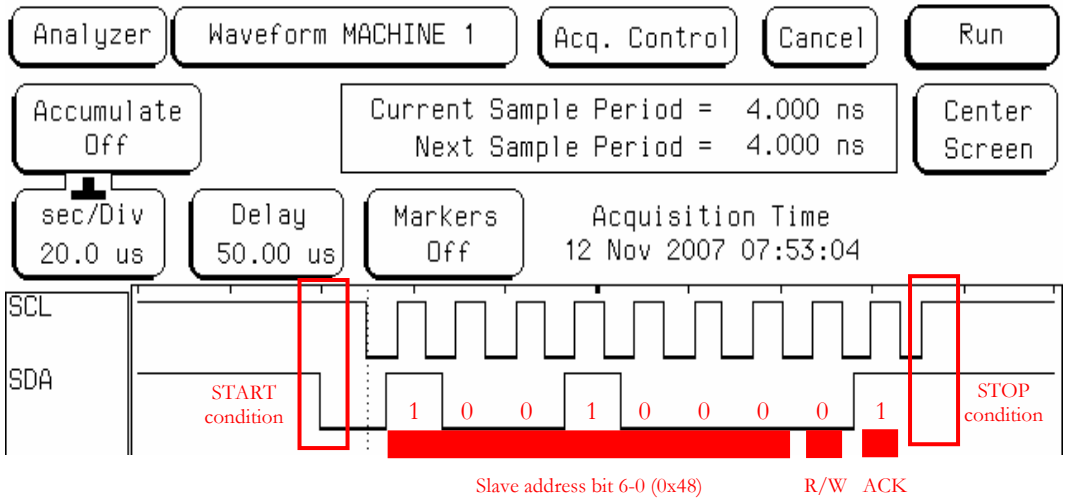


Figure 64 : I2C transfer from the MSP430F1611

The precedent capture shows the transmitted bits on the I2C interface:

- The START condition has been executed
- Then the bit 6-0 of the slave address 0x48 is transmitted
- The 8th bit is at 0, so the master transmits to the slave
- The 9th bit represents the acknowledgement of the slave. Normally if the slave responds, he drives a 0 and the value on the picture is 1. For this test it's normal because no slave devices are connected to the I2C interface. The SDA line his pulled-up, so it's normal to see this bit at level 1.
- When a slave doesn't acknowledge (like in our case, but its normal), the data line must be HIGH by the slave (in our case by the pull-up). The master can then generate a STOP condition to abort the transfer.

8.8.4 Results

The I2C interface of the CDMS board is working and was demonstrated by this little test. Now it's the job of the software team to implement all the functionality needed to ensure the communication between the different sub-systems of the satellite.

8.9 EPROM AT27BV4096 VSOP40 package

It's planned to write the final flight software on an EPROM. This solution was chosen because the spatial environment is very particular with their problem of radiation. The EPROM is a one-time programmable memory, for this reason it's quite impossible to modify the value of the stored data. For the moment, the EPROM is replaced by a Flash of 2MB.

Before solder to the PCB the EPROM, it's necessary to program it. To make this operation, a programming system with the corresponding type of socket is indispensable. At the HES-SO Sion we have a programming system (UniSite-xpi from DATA/IO) to program a lot of components package like PLCC, SOIC, TSOP, QFP, and PGA but no for the VSOP package.

An adapter for this type of package exists for the programming system we have, but his price is very expensive and the delay of delivery are longer.

I receive an offer for another new product from:

Computer Control SA www.ccontrols.com

Programming systeme	
Programmer +48	1800.-

VSOP adapter	
AS40-40-03TS-6YAN-G-S	435.-

TOTAL	2235.-

Tableau 18 : Programmer +48 offer

Here is the web link to the programmer +48:

http://www.data-io.com/products/default.asp?id=10

A lot a programming system exists on the market and one of these products should be choose in order to program the EPROM AT27BV4096. Another solution is to used the same EPROM model but in package PLCC 44 pin, because we have the adapter for the programming system UniSite-xpi at school. But the size and the height of the component are much higher and the type of pinout too. It's why for the moment it was not decided to change for this package.

8.10 Test of the temperature sensor

The CDMS should be able to deliver the temperature of the board. The temperature sensor used is the LM94022BIMG. This temperature sensor his connected to an A/D input of the microcontroller. The AT91M55800A has an internal sample and hold circuit that holds the sampled value during a complete conversion. The reference voltage pin ADVREF on the AT91M55800A allows the analog input conversion range to be set between 0 and ADVREF (in our case +2.5V). Analog inputs between these voltages convert to values based on a linear conversion.

8.10.1 Objective

The goal of this test is to configure by software the different registers used for the configuration of the Analog to Digital Converter and to take some measurements, at regular intervals, in order to see if the temperature sensor works, and if the converted values are realistic.

8.10.2 Gain of the temperature sensor

The following picture represents the value of the output voltage vs the temperature.

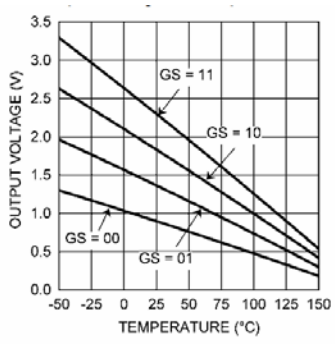


Figure 65 : Output voltage vs temperature

With the gain we choose (GS=01, -8.2 mV/°C, we have a value of tension between 1955 mV (-50°C) and 301 mV (+150°C).

The temperature in the laboratory is about 23°C. So the value of the tension at this temperature should be 1356 mV.

1955 mV – (50+23) * 8.2 mV/°C = 1356 mV

The measured value was about 1840 mV and does not represent the right value. After a moment, a realized that the chosen gain is not GS=01 but GS=10. I made an error on the schematic, and this error should be corrected for the flight model. The only problem with this gain is that all the values of tension above 2.5V will not be properly corrected because the analog input conversion range is set between 0 V and ADVREF (in our case +2.5V). But for this board it's not a problem.

With the new gain (GS=10, -10.9 mV/°C), we have a value of tension between 2616 mV (-50°C) and 420 mV (+150°C). With the temperature of the laboratory of about 23°C the value of the tension should be about 1820 mV, and this time it correspond with the measured value.

2616 mV – (50+23) * 10.9 mV/°C = 1820 mV

8.10.3 Test of the temperature sensor

The software code takes 20 values of tension with an interval of about 4 seconds. The voltage of the temperature is sampled and converted in a digital value of 10 bits. I made this test with different temperature conditions to check the comportment of the temperature sensor.

For detailed information about the configuration of the A/D converter see:

→APPENDIX H function "*initADC0()*"

The sampled value is converted into a digital value of 10 bits. So in order to convert this value in temperature degrees I used the following formula:

$$T[°C] = ((2.616 - (2.5/2^{10} * sampledvalue))/10.9^{-3}) - 50$$

Equation 1 : Digital to degrees conversion

At ambient temperature in the laboratory:

tempDeg	0x00001824
tempDeg[0]	22.237314
tempDeg[1]	22.90926
tempDeg[2]	23.133242
tempDeg[3]	22.90926
tempDeg[4]	23.581207
tempDeg[5]	22.90926
tempDeg[6]	22.90926
tempDeg[7]	22.90926
tempDeg[8]	23.357225
tempDeg[9]	22.90926
tempDeg[10]	22.90926
tempDeg[11]	22.90926
tempDeg[12]	24.029171
tempDeg[13]	23.357225
tempDeg[14]	22.90926
tempDeg[15]	22.90926
tempDeg[16]	23.133242
tempDeg[17]	22.90926
tempDeg[18]	23.357225
tempDeg[19]	23.357225

Tableau 19 : Temperature value of test 1

After having put the card outside for a few minutes:

tempDeg	0x00001874
tempDeg[0]	9.022363
tempDeg[1]	9.918291
tempDeg[2]	10.142274
tempDeg[3]	10.142274
tempDeg[4]	10.366256
tempDeg[5]	11.262185
tempDeg[6]	11.262185
tempDeg[7]	11.486167
tempDeg[8]	11.710149
tempDeg[9]	11.710149
tempDeg[10]	12.1581135
tempDeg[11]	12.1581135
tempDeg[12]	12.1581135
tempDeg[13]	12.606078
tempDeg[14]	12.606078
tempDeg[15]	12.83006
tempDeg[16]	13.054043
tempDeg[17]	13.949971
tempDeg[18]	13.502007
tempDeg[19]	14.173954

Tableau 20 : Temperature value of test 2

8.10.4 Results

- For the first test, at an ambient temperature of about 23°C, the 20 sampled values were between 22.23°C and 24.02°C. It represents a difference of 1.79°C between the minimal and the maximal temperature. The datasheet of the LM94022 specifies an error of +-1.8°C in the temperature condition between +0°C to +70°C with a gain GS=10. So these results are correct.
- For the second test, I took 20 measures, after having put the board outside for a few minutes. From the first to the last sampled value, the temperature has increased.

The different tests performed in different conditions of temperature, confirm that the temperature sensor is properly working. The only change for the next board is to choose the correct gain GS=01.

The capacitor C2 should be not present for the Flight Model of the CDMS board. A problem was detected during the first test of the temperature sensor. This capacitor modifies the input impedance during the sampling, and the sampled values are often different of about 10°C. Without this capacitor, the comportment of the temperature sensor is correct in any case.

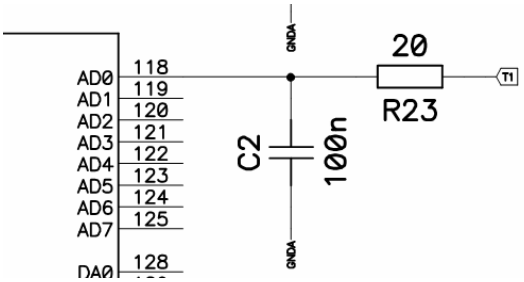


Figure 66 : Input impedance of the A/D converter

Test of the clock selection on the AT91M55800A

The AT91M55800A has three possible sources of clock. The Master Clock can be selected through the Slow Clock at 32768 Hz (present at reset), the output of the Main Oscillator at 4 MHz or the output of the PLL at 32 MHz. At reset, to save power, only the Slow Clock is present. After the reset, it's possible the enable or disable one of these three sources of clock by software.

8.10.5 Objective

The goal of this test is to show that the three clock sources are working and can be controlled by software. To make this test I used an oscilloscope on the pin MCKO of the AT91M55800A. This pin drives in output the current clock.

For detailed information about the configuration in order to switch these clocks see:

→APPENDIX H function *"lowLevelInit()"*

8.10.6 Clock switching

8.10.6.1 Slow Clock at 32768Hz

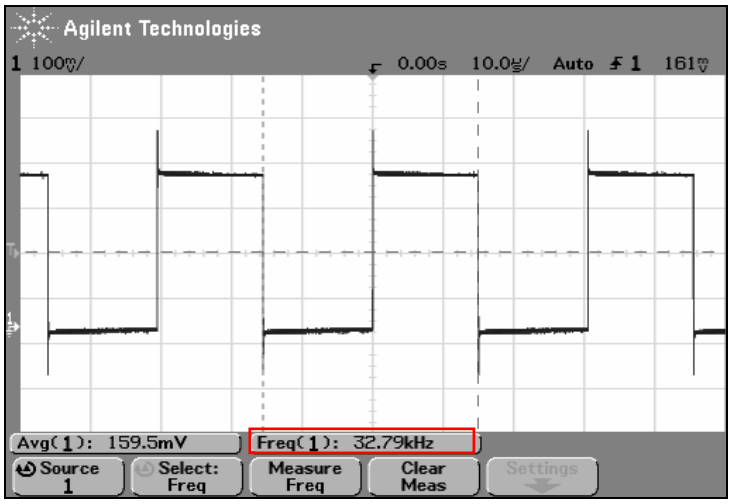


Figure 67 : Slow clock at 32768Hz

At reset the Slow Clock of 32768 Hz is present and assures correct operation of the microcontroller.

8.10.6.2 Output of the Main Oscillator at 4MHz

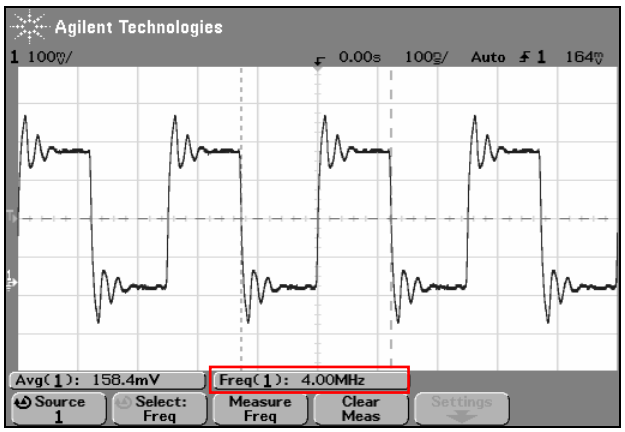


Figure 68 : Main Oscillator at 4MHz

By software, the output of the main oscillator at 4MHz is choosing.

8.10.6.3 Output of the PLL at 32 MHz

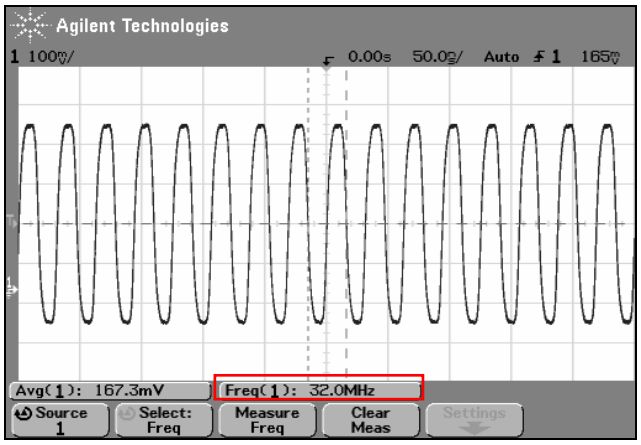


Figure 69 : PLL at 32MHz

The output of the Main Oscillator can be multiplied by a number between 0 (PLL disable) until 63. In our case the output of the Main Oscillator is 4MHz and a multiplication by eight is applied in order to have an operation frequency of 32MHz on the AT91M55800A.

8.10.7 Results

The three sources of clock working can be controlled by the software.

9 PSA ANALYSIS OF THE CDMS QUALIFICATION MODEL

To validate a space electronic board, different analysis should be executed in order to check if the board satisfies these different demands. These different analyses are very important in order to identify as soon as possible a potential problem in order to adapt the design in the way to limit the risk of failures or damages of a component or the board, during the mission.

Different analysis exists and their test procedures are done by the ECSS norms.

- Mechanical analysis
- Thermal analysis
- Radiation analysis
- Reliability analysis
- Worst Case Analysis (WCA)
- Failure Modes, Effect and Critical Analysis (FMECA)
- Part Stress Analysis (PSA)

The major parts of these analyses are or will be done by the EPFL, on the different boards of the satellite. However, I made an analysis on the CDMS Qualification Model board, the Part Stress Analysis.

9.1 Definition of the Part Stress Analysis

This analysis checks that the functional parameters of the components are respected with a safety margin given by the ECSS norm, for each category of component family. The goal of this analysis is to avoid stress, and detect all the components which could exceed the fixed limits.

If the analysis detects a potential problem of a component, a solution must be found like the replacement of the component or a modification of the design.

9.1.1 How to make the PSA Analysis

The different components present on the CDMS board should be first identified. Then the derating requirements are provided for each component family.

For each category and component type, the parameters to be derated are identified. The main parameters to be derated are:

- Junction or case temperature at maximum operating conditions
- Power (rating, dissipation)
- Voltage
- Current

These derating values must be applied for each parameter, to the maximum value given by the manufacturer and compared with the worst case that could appear on the circuit. The result of this comparison shows the percentage of stress for each parameter and if a parameter exceeds the fixed limits.

9.1.1.1 Example of PSA Analysis

The identified component C1 on the schematic of the CDMS is a ceramic capacitor. His family and group code is:

01	02	Capacitors	Ceramic Chip

Figure 70 : ECSS family and group code

With this family and code group, we could find the different parameters to be derated for this capacitor.

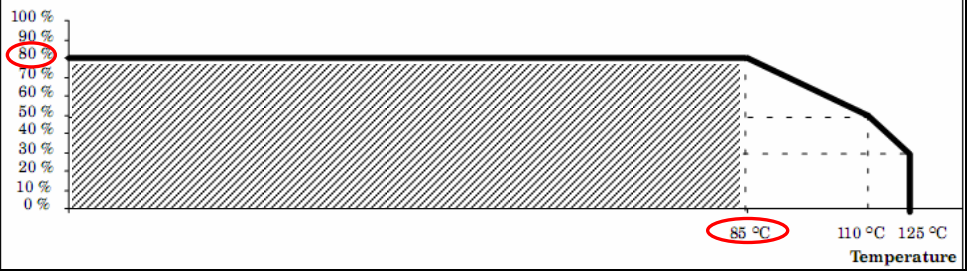
Parameters	Load ratio or limit
Voltage: rated voltage ≤ 500 V	

Figure 71 : Derating parameter for ceramic capacitor

Then all these information's can be put in an Excel table in order to simplify all the calculus.

CDMS qualif. Board	**C1**	CAPS0603 Ceramic 100nF 25V 10%	V_max	25 V	80%	20 V	3.6 V	18.0%	ok
			T_max	125 °C	100%	125 °C	60 °C	48.0%	ok
				1	2	3	4	5	6	7

1. Parameter
2. Value max given by the manufacturer
3. Derating value shown by the graphic
4. The allowed value
5. The maximal value that could happen in the circuit
6. The percent of stress
7. The status of the parameter

9.1.2 PSA Analysis on the CDMS board

The same procedure as it was demonstrated in the precedent example was followed in order to check all components present on the board.

→See APPENDIX J for the entire PSA Analysis report

9.1.3 Results

After having made the PSA Analysis, we can see that for some components, a stress above the fixed limits is reached for some parameters. For the capacitors, resistors there are no problems. All of these components with their proper parameter are below the maximal stress values. Some other components have one or more parameters that have reached or exceed the limit of stress.

The following table lists the different components where a problem was identified

Component	Parameter	Rated value	Derating	allowed value	Value	Stress	Status
AT91M55800A Microcontroller	Vddcore	3.6 V	90%	3.24 V	3.6 V	111.1%	Nok
	Vdda	3.6 V	90%	3.24 V	3.6 V	111.1%	Nok
	Vddpll	3.6 V	90%	3.24 V	3.6 V	111.1%	Nok
	Vddbu	3.9 V	90%	3.51 V	3.6 V	102.6%	Nok
	Vddio	5.5 V	90%	4.95 V	3.6 V	72.7%	ok
	Tj	85°C	Tjmax-40°C	45 °C	60 °C	133.3%	Nok
Microcontroller MSP430F1611IPM	Vcc	4.1 V	90%	3.69 V	3.6 V	97.6%	ok
	Tj	85°C	Tjmax-40°C	45 °C	60 °C	133.3%	Nok
R1LV0416CSB-7LI SRAM 256k x 16 bit	Vcc	3.6 V	90%	3.24 V	3.6 V	111.1%	Nok
	Tj	85 °C	Tmax-40°C	45 °C	60 °C	133.3%	Nok
S29AL016D90TFI02 FLASH 1M x 16 bit	Vcc	4 V	90%	3.6 V	4 V (max)	100.0%	ok
	Iout	200 mA	80%	160 mA			
	Tmax	85 °C	Tmax-40°C	45 °C	60 °C	133.3%	Nok
74LVX08M LOGIC AND GATE	Vcc	7 V	90%	6.3 V	3.6 V	57.1%	ok
	Tj	85 °C	Tjmax-40°C	45 °C	60 °C	133.3%	Nok

Tableau 21 : PSA analysis results

The main problem is with the temperature parameter (or junction temperature). The ECSS norm fixes that the derating value must be:

Tjmax-40°C, where *Tjmax* represents the junction temperature which if specified by the manufacturer or if not specified, the maximum operation temperature should be considered. In any case the problem is the same. With the respect of the ECSS norms, these parameters exceed the limits of stress.

The second problem that was identified is on the voltage parameter on both microcontrollers and the Sram.). The ECSS norm gives the following indication about the derating of this parameter:

90% of maximum rating (recommended by manufacturer)

So as we can understand, it's more a recommendation then a strict limit like it's the case for the others parameters analyzed.

9.1.4 Conclusion of the PSA Analysis

An important point to consider with the problems that were identified is that the Swisscube satellite is entirely build with commercial components and not with space qualified products. So it's normal that the exigencies for commercial products are lower that for a space qualified product.

For the Flash component S29AL016DTFI02, an extended version of this device exists with the operating range between -40°C to 125°C. So for the flight model of the CDMS board, this component will be simply change and the problem of stress on the temperature will be resolved.

For the others components, these results should be yet compared with the result of the PSA Analysis performed on the others subsystems. Each of these potential problems should be discussed in order to know if these know problems should be a potential risk of failures or damage for the mission and for the satellite.

10 BUDJET AND PERFORMANCE OF THE CDMS MODEL "QM"

Mechanical	
PCB Dimensions	93.5 x 87 x 1.55 [mm]
Maximal components height	5 [mm] over both sides of the PCB
Weight	34 [g]

Electrical	
Supply	+3.3 [V]
Consumption at 32768Hz	18.5 [mW]
Consumption at 4MHz	33.66 [mW]
Consumption at 32MHz	155.1 [mW]

Sub-system interfaces	
I2C	Standard mode (up to 100kbps) and fast mode (up to 400kbps)
Power suply	2 x +3.3 [V]
	4 x GND
Programming connectors	JTAG for the AT91M55800A
	JTAG for the MSP430F1611

Test interfaces	
Button S1	Manual RESET
Button S2	Manual WAKEUP event
Switch	8 ON/OFF positions

Memories	
Flash	3 x Spansion Flash S29AL016D91TFI02 2MB
Sram	1 x Renesas Sram R1LV0416CSB-7LI 512kB
ROM (only for flight model)	1 x ATMEL AT27BV4096 512kB

Microcontroller	
AT91M55800A	ARM7TDMI processor core, 32-bit RISC Architecture, 8KB Internal SRAM, maximal operating frequency 33MHz
MSP430F1611IPM	16-bit RISC Architecture, 48KB + 256B Flash Memory + 10KB Internal RAM, maximal operating frequency 8MHz

PCB	
Substrat	FR4, 6 layers

11 CONCLUSION

The goal of this diploma works is reached. A new CDMS board called the Qualification Model was developed, with the results of the different tests performed on the Engineering Model board. All the functionalities of this new board were tested, and the results show that all is properly working. The CDMS board respect all the requirements fixed by the specifications. A development environment for each microcontroller was used to debug simultaneously the ARM and the MSP430, to test the main new feature of the board, the SPI interface between these two microcontrollers as well the temperature sensor. The PSA Analysis has shown some potential problems, which should be discussed with the person working on the other subsystems, to know how to interpret these results.

The board is now ready to be delivered at the software team, for the software test integration.

12 FUTUR WORK

The first thing to do on the CDMS Qualification Model will be the software integration by the software team in St-Imier. Normally all subsystems should have developed a board and the CDMS will be integrated with all other subsystems of the Swisscube satellite. Some additional tests should be performed in different domains like vibration test, thermal test and radiation, to ensure that the CDMS board could operate in a space environment.

Finally a third board, the flight model of the CDMS will be build with the last modification and launch in space.

23 November 2007

Crettaz David

13 REFERENCES

[R1] The document of functional specification "S3-B-SE-2-0-CDMS_specification"

[R2] The PDF "doc1745.pdf" from ATMEL for the AT91M55800A

[R3] The PDF "doc1727.pdf" from ATMEL for the AT91M55800A Electrical Characteristics

[R4] The PDF "slau049.pdf" from Texas Instruments for the MSP430x1xx Family

[R5] The PDF "39340011.pdf" from Philips for the I2C bus specification, version 2.1

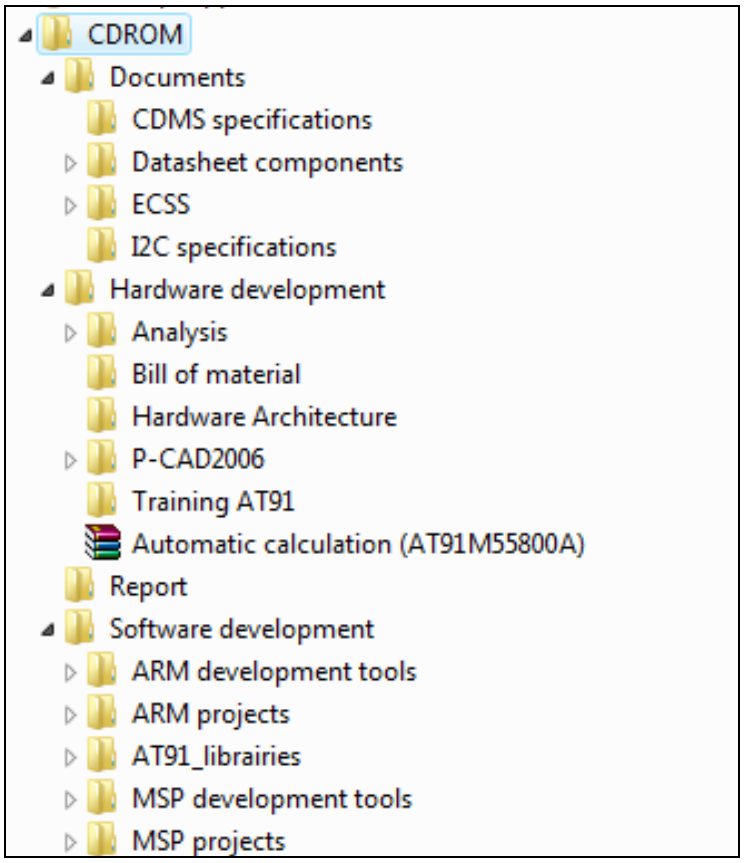
[R6] The Excel File "Automatic calculation (AT91M55800A).zip for PLL Filter Calculation

[R7] The document "Phase-B_EPS_Power_Management_Board_Stiener"

[R8] ECSS-Q-70-11A

14 ARCHIVE

The following figure shows the files contain in the CDROM annexed to my report.



APPENDIX A OPENOCD COMMANDS

OpenOCD Quick Reference Card

OpenOCD Homepage
`http://openocd.berlios.de`
Current OpenOCD revision: 128

Server

Configuration Commands

`telnet_port <port>` Listen for telnet connections on *port*.

`gdb_port <port>` Listen for GDB connections on *port*, *port+1*, ... (target 1, target 2, ...)

User Commands

`shutdown` Shut server down.

`exit` Exit telnet. Leaves server running.

`help` Print available commands.

Interpreter

The interpreter commands may be used to define variables used within other subsystems like JTAG.

`var <name> ['del'|([size1] [sizeN])]`
Allocate, display or delete variable. Allocation has to define the size for all num_fields elements.

`field <var> <field> [value| 'flip' ]`
Display or modify variable field.

`script <file>`
Execute commands from file.

Target

Configuration Commands

`target <type> <endianness> <reset_mode>`
Target *type* is arm7tdmi, arm9tdmi, arm720t, arm920t, arm926ejs, arm966e; *endianness* is either big or little; *reset_mode* is one of reset_halt, reset_run, reset_init, run_and_halt, run_and_init. **Do not use reset_halt or reset_init on LPC2 or STR7**.

`target arm7tdmi <endianness> <reset_mode> <jtag#> [variant]` *variant* is one of arm7tdmi_r4, arm7tdmi-s_r4, lpc2000.

`target arm9tdmi <endianness> <reset_mode> <jtag#> [variant]` *variant* is one of arm920t, arm922t and arm940t.

`daemon_startup <'reset'|'attach'>` Describes what to do with target on daemon startup.

`target_script <target#> <event> <scriptfile>` *event* is either post_halt or pre_resume.

`run_and_halt_time <target#> <time>` Delay in msec between reset and debug request.

`working_area <target#> <addr> <size> [backup|nobackup]`

User Commands

`targets [num]` Display list of configured targets, or make target *num* the current target.

`reg [#|name] [value|'force']` Display or modify registers.

`poll ['on'|'off']` Print information about the current target state. If the target is in debug mode, architecture specific information about the current state are printed. Enable or disable continuous polling with optional parameter.

`wait_halt` Wait for target halt

`halt` Halt target

`resume <address>` Resume the target at the current position or at *address*.

`step <address>` Single-step at the current position or at *address*.

`reset ['run'|'halt'|'init'|'run_and_halt'|'run_and_init']` Reset the target in a few variations.

`soft_reset_halt` Halt the target and do a soft reset.

`md[whb] <address> [count]` Display *count* words (32 bit), half-words (16 bit) or bytes at *address*. If *count* is omitted, one element is displayed.

`mw[whb] <address <value>` Write *value* at the word, half-word or byte location *address*.

`bp <address> <length> [hw]` Set a breakpoint of *length* bytes at *address*.

`rbp <address>` Remove breakpoint at address.

`wp <address> <length> <r|w|a> [value] [mask]` Set a watchpoint of *length* bytes at *address*.

`rwp <address>` Remove a watchpoint at *address*.

`load_binary <file> <address>` Load binary *file* into target memory (RAM) at *address*.

`dump_binary <file> <address> <size>` Dump target memory of *size* bytes at *address* into *file*.

ARM v4/5

`armv4_5 reg` Display all banked ARM core registers.

`armv4_5 core_mode ['arm'|'thumb']` Display the current core state, or switch between Arm and Thumb state.

`armv4_5 disassemble <address> <count> ['thumb']` Disassemble instructions

ARM720T

The commands are `cp15`, `virt2phys`, `mdw_phys`, `mdh_phys`, `mdb_phys`, `mww_phys`, `mwh_phys`, `mwb_phys`. See ARM7/9 for a short description.

ARM 7/9

`arm7_9 write_xpsr <value> <spsr>`
Write the program status register. spsr selects between the current program status register (0) and the saved program status register (1) of the current mode.

`arm7_9 write_xpsr_im8 <8bit immediate> <rotate> <not cpsr|spsr>`
Same as write_xpsr, but use the immediate operand opcode.

`arm7_9 write_core_reg <num> <mode> <value>`
Write core register num of mode with value.

`arm7_9 sw_bkpts <enable|disable>`
Enable or disable the use of software breakpoints.

`arm7_9 force_hw_bkpts <enable|disable>`
Force the use of hardware breakpoints.

`arm7_9 dbgrq <enable|disable>`
Use EmbeddedICE dbgrq instead of breakpoint for target halt requests (safe for all except ARM7TDMI-S).

`arm7_9 fast_memory_access <enable|disable>`
Use fast but potentially unsafe memory accesses instead of slow accesses. Assumes a sufficiently high clock speed.

`arm7_9 dcc_downloads <enable|disable>`
use DCC downloads for larger memory writes.

ARM920T

`arm920t cp15 <num> [value]` Display/modify CP15 register

`arm920t cp15i <opcode> [value] [address]`
Display/modify cp15 (interpreted access)

`arm920t cache_info` Display information about target caches.

`arm920t virt2phys <va>` Translate virtual to physical address.

`arm920t md[whb]_phys <physical addr> [count]` display memory word, half word, byte

`arm920t mw[whb]_phys <physical addr> <value>` write memory word, half word, byte

`arm920t read_cache <filename>` display I/D cache content

`arm920t read_mmu <filename>` display I/D mmu content

ARM926EJ-S
`arm926ejs ...` XXX To Do

ARM966E

`arm966e cp15 <num> [value]` Display/modify CP15 register

JTAG

Configuration Commands

`interface <name>` – *name* is one of parport, amt_jtagaccel, ft2232, ep93xx

`jtag_device <IR length> <IR capture> <IR mask> <IDCODE instruction>` – Describes the devices that form the JTAG daisy chain, with the first device being the one closest to TDO.

`jtag_nsrst_delay <ms>` – *ms* milliseconds delay between going nSRST inactive and following JTAG operations.

`jtag_ntrst_delay <ms>` – *ms* milliseconds delay between going nTRST inactive and following JTAG operations.

`reset_config <signals> [combination] [trst-type] [srst-type]` – *signals* is one of none, trst_only, srst_only or trst_and_srst. *combination* is one of srst_pulls_trst, trst_pulls_srst, combined, separate. *trst-type* is one of trst_open_drain, trst_push_pull. *srst-type* is one of srst_push_pull, srst_open_drain.

User & Config Commands

`jtag_speed <value>` Select JTAG Speed **ft2232**: 6 MHz / (*value*+1) **parport**: maximum speed / value **amt_jtagaccel**: 8 / 2**value

Note: Max. JTAG-Clock $\approx \frac{1}{6}$ × CPU-Clock!

Parport

`parport_port <port|num>` – Either I/O *port* address (e.g. 0x378) or the *number* of the /dev/parport device.

`parport_cable <name>` – *name* is one of wiggler, old_amt_wiggler, chameleon, dlc5 (Xilinx cable III), triton.

Amt_jtagaccel

`parport_port <port|number>` see previous description.

Ft2232

`ft2232_device_desc <description>` USB device *description*. Use `usbview` or similar tool to get this string.

`ft2232_layout <name>` Layout *name* is one of `jtagkey`, `usbjtag`, `signalyzer`, `olimex-jtag`, `m5960`, `evb_lm3s811` .

`ft2232_vid_pid <vid> <pid>` Vendor-ID *vid* and product-ID *pid* of the FTDI device.

User Commands

`scan_chain` Print scan chain configuration.

`endstate <tap_state>` Finish JTAG operations in *tap_state*.

`jtag_reset <trst> <srst>` Toggle reset lines.

`runtest <num_cycles>` Move to Run-Test/Idle and execute *num_cycles*.

`statemove <tap_state>` Move to current endstate or *tap_state*.

`irscan <device> <instr> [devN] [instrN]` Execute IR scan.

`drscan <device> <var> [devN] [varN]` Execute DR scan.

Flash

`flash banks` Display list of configured flash banks

`flash info <num>` Display information and list of blocks of flash bank *num*.

`flash probe <num>` Probe flash bank *num* if it matches the configured bank.

`flash erase_check <num>` Check erase state of flash sectors in bank *num*.

`flash protect_check <num>` Check protect state of flash sectors in bank *num*.

`flash erase <num> <first> <last>` Erase sectors at bank *num*, starting at sector *first* up to and including *last*. Sector numbering starts at 0.

`flash write <num> <binfile> <offset>` Write *binfile* (in **binary** format!) to bank *num* at *offset* bytes from start of bank.

`flash protect <num> <first> <last> <'on'|'off'>` Enable (on) or disable (off) protection of flash sectors *first* to *last* of flash bank *num*.

`flash bank <driver> <base> <size> <chip_width> <bus_width> [driver_options...]`
Configure a flash bank at address *base* of *size* bytes with a bus of *bus_width* bits formed by chips of *chip_width* bits size using *driver* lpc2000, cfi, at91sam7, str7x, str9x.

`flash bank lpc2000 <base> <size> 0 0 <lpc_variant> <target#> <cclk> ['calc_checksum']`
The internal flash of LPC2000 devices doesn't require chip- and buswidth to be defined. The *lpc_variant* specifies the supported IAP commands of the device (**lpc2000_v1**:2104—5—6, 2114—9, 2124—9, 2194, 2212—4, 2292—4; **lpc2000_v2**: 213x, 214x, 2101—2—3). The flash bank is part of *target#* which runs at *cclk* kHz. *Calc_checksum* inserts a valid checksum into the exception vector when this area is flashed.

For LPC2138 and LPC2148 devices, the *size* argument has to reflect the user-accessible size, i.e. 0x7d000 (500kB), not 0x80000 (512kB).

`flash bank at91sam7 0 0 0 0 <target#>`

`flash bank str7x <base> <size> 0 0 <variant> <target#>` – *variant* is one of STR71x, STR73x or STR75x.

`flash bank str9x <base> <size> 0 0 0`
The str9 needs the flash controller to be configured prior to Flash programming:
`str9x flash_config b0size b1size b0start b1start`
Example: `str9x flash_config 4 2 0 0x80000`
This will setup the BBSR, NBBSR, BBADR and NBBADR registers respectively.

`flash bank cfi <base> <size> <chip_width> <bus_width> <target#>`

`at91sam7 gpnvm <num> <bit> set|clear` set or clear at91sam7 gpnvm bit

`lpc2000 part_id <num>` print part id of lpc2000 flash bank num

XSVF

`xsvf <devnum> <file>` Program Xilinx Coolrunner CPLD

PLD

XXX To Do

Commandline Options

```
$ openocd  --help
Open On-Chip Debugger
(c) 2005 by Dominic Rath

--help       | -h  display this help
--file       | -f  use configuration file <name>
--debug      | -d  set debug level <0-3>
--log_output | -l  redirect log output to file <name>
--interface  | -i  use jtag interface driver <name>
```

Sample Configuration

[see $(SRCDIR)/doc/configs/*.cfg]

Some GDB commands

`(gdb) target remote localhost:3333`

`(gdb) monitor arm7_9 force_hw_bkpts enable`
force hardware breakpoints

More information

- Get current version with Subversion
`svn co svn://svn.berlios.de/openocd/trunk`

- OpenOCD Forum at Spark Fun Electronics
`http://www.sparkfun.com/cgi-bin/phpbb/viewforum.php?f=18`

- OpenFacts
`http://openfacts.berlios.de/index-en.phtml?title=Open_On-Chip_Debugger`

- Mailing List "openocd-development"
`http://developer.berlios.de/mail/?group_id=4148`

- Yagarto ARM toolchain (Windows)
`http://www.yagarto.de`

- Amontec JTAGkey[-Tiny], sdk4arm
`http://www.amontec.com`

- Olimex ARM-USB-TINY, ARM-USB-OCD
`http://www.olimex.com`

QuickRef written by Hubert.Hoegl@fh-augsburg.de
Date: 2007-02-03

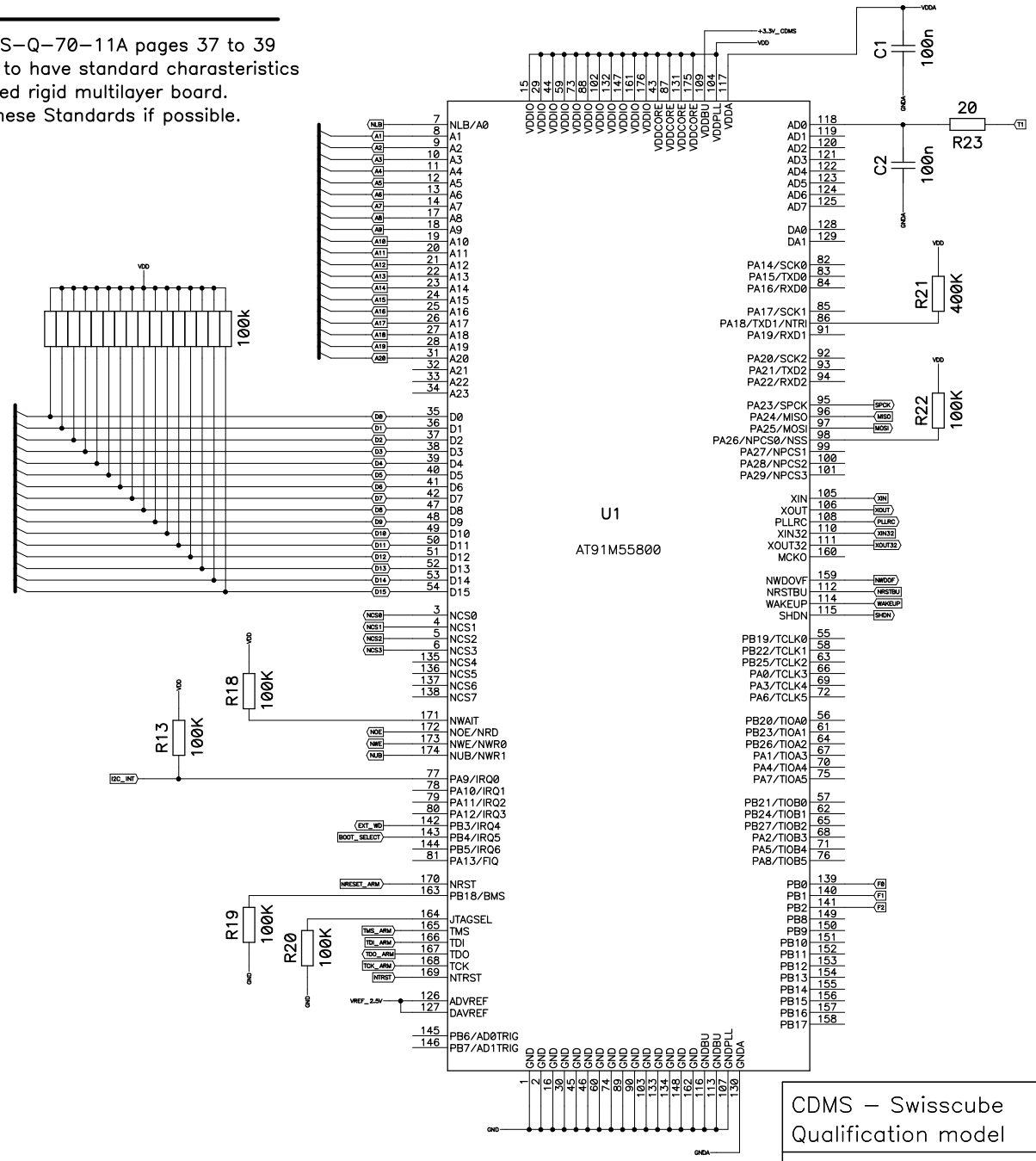
Download this QuickRef from

`http://www.fh-augsburg.de/~hhoegl/proj/openocd/oocd-quickref.pdf`

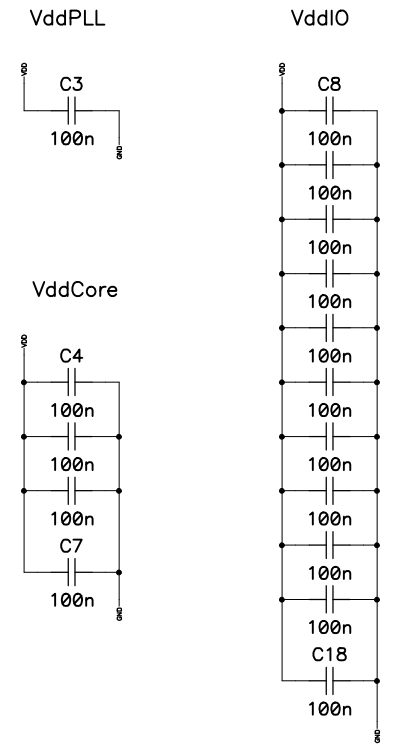
APPENDIX B SCHEMATIC OF THE CDMS MODEL "QM"

CPU

See ECSS-Q-70-11A pages 37 to 39
in order to have standard charasteristics
of finished rigid multilayer board.
Follow these Standards if possible.



Decoupling



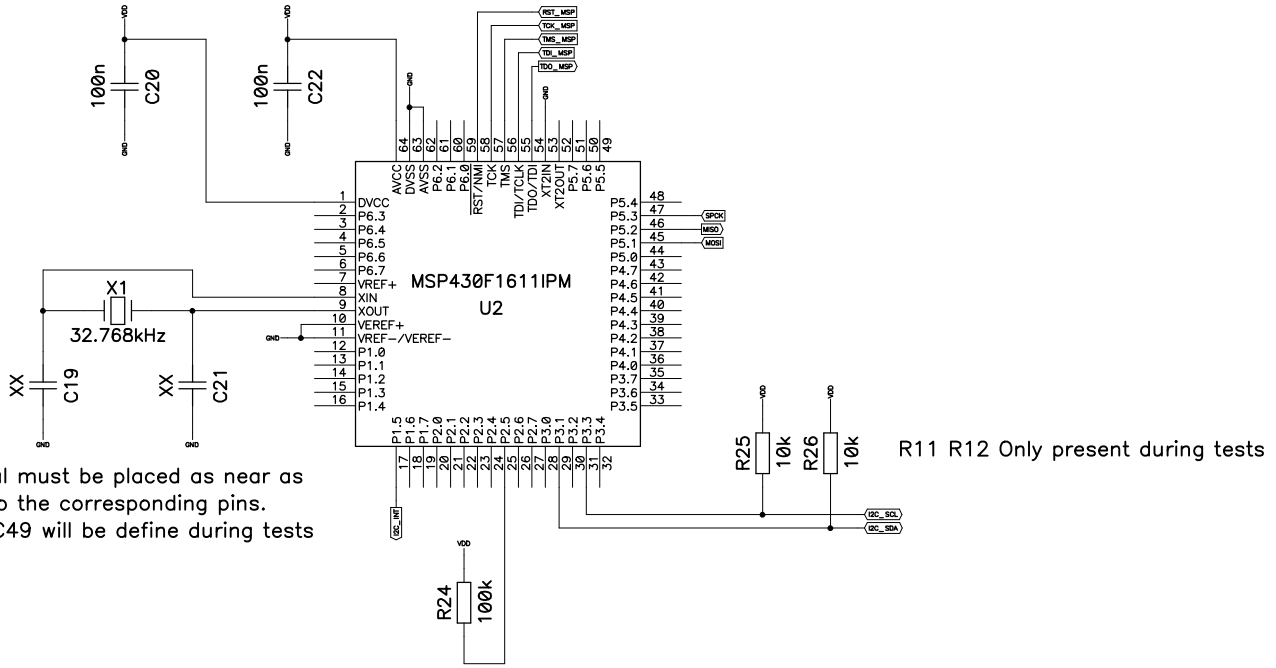
These decoupling capacitor must
be placed as near as possible
to corresponding pin.

Vddbu decoupling capacitor (C39)
is presented on page 7.

CDMS – Swisscube		DES	25.09.2007 Crettaz David
Qualification model	ARM	REV	V1.0
HAUTE ECOLE VALAISANNE		1/10	{Path} CDMS_2.0.sch

MSP430F1611

These decoupling capacitor must be placed as near as possible to corresponding pin.



R11 R12 Only present during tests

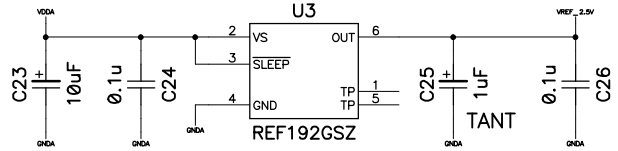
The crystal must be placed as near as possible to the corresponding pins.
C44 and C49 will be define during tests

Rosc for internal DCO

Metal film resistor, type 0257
0.6 watt with 1% tolerance and Tk = +/- 50ppm/C

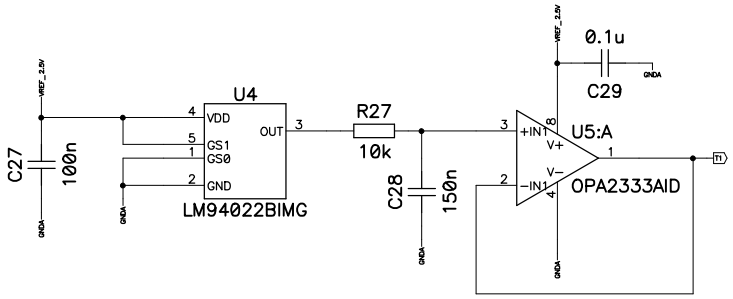
CDMS – Swisscube		DES	25.09.2007 Crettaz David
Qualification model	MSP	REV	V1.0
HAUTE ECOLE VALAISANNE		2/10	{Path} CDMS_2.0.sch

Voltage reference +2.5V (max 30 mA)

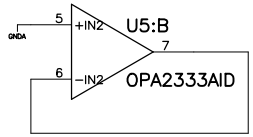


This voltage reference provide the supply for all analogic part on the CDMS board

LM94022 + Filter

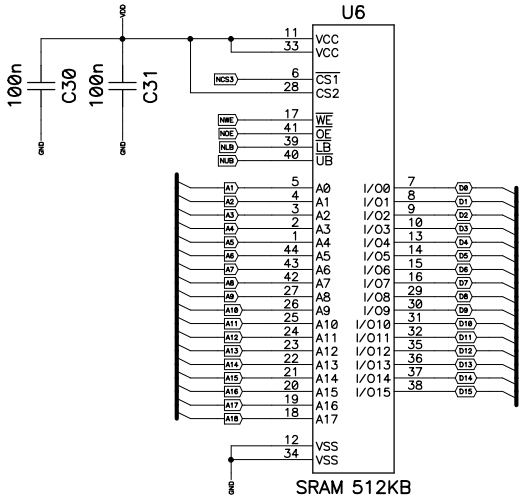


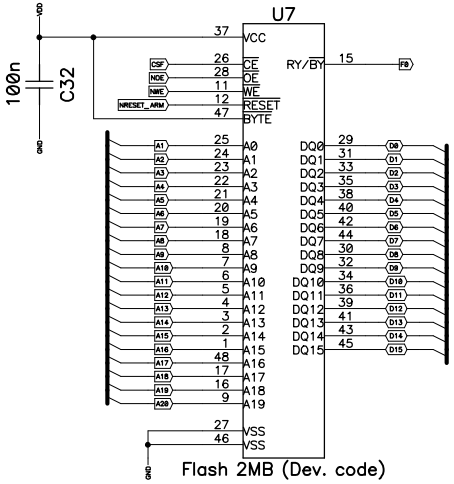
The temperature sensor is follow by a filter before to send the measure to the ARM

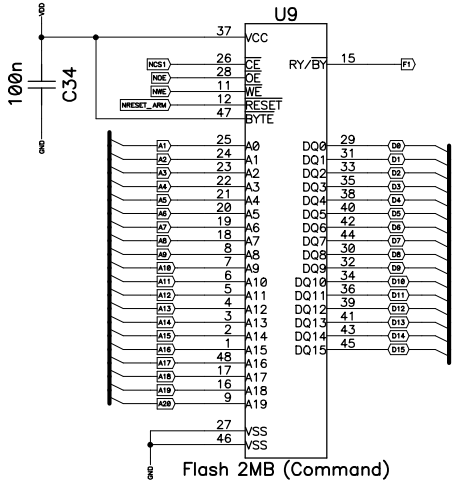


CDMS – Swisscube Qualification model LM94022+FILTER		DES	25.09.2007 Crettaz David
		REV	V1.0
HAUTE ECOLE VALAISANNE		3/10	{Path} CDMS_2.0.sch

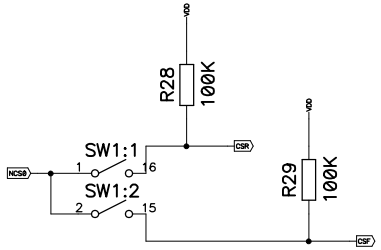
Memories



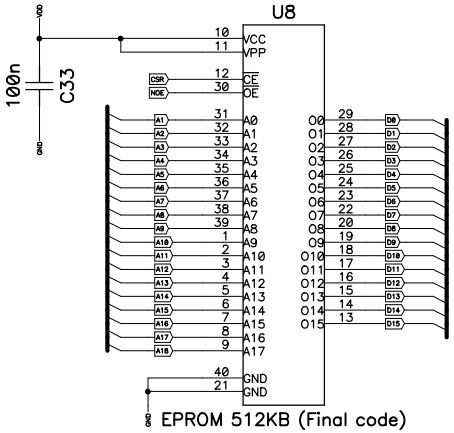


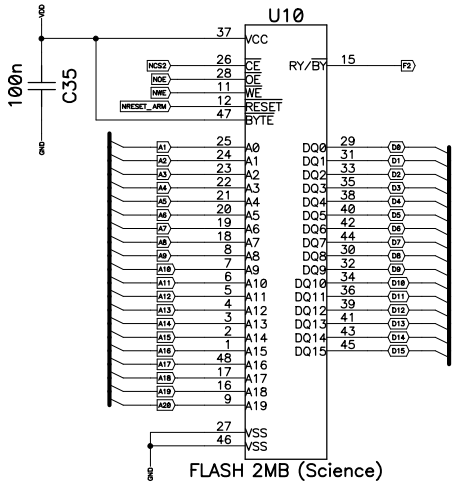


NCS0 – selection switch



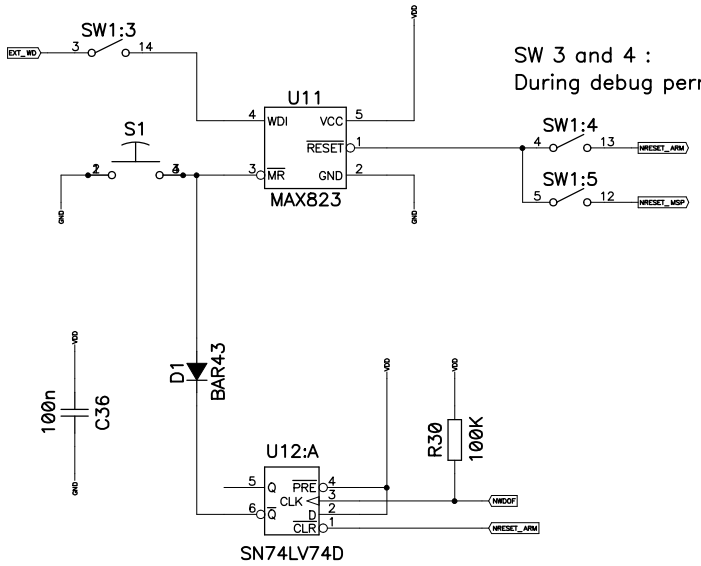
R6 and R7 are Pull–up resistor for respectively CSR and CSF signals.





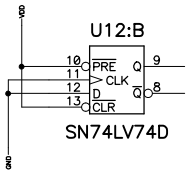
CDMS – Swisscube Qualification model	Memory	DES	25.09.2007 Crettaz David
		REV	V1.0
HAUTE ECOLE VALAISANNE		4/10	{Path} CDMS_2.0.sch

Reset part

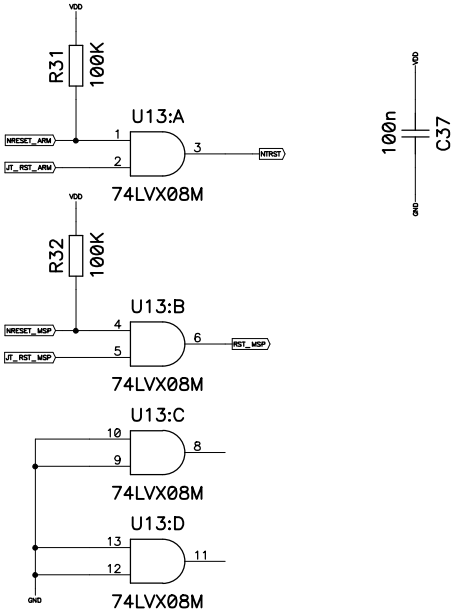


SW 3 and 4 :
During debug permit to reset independly the ARM or the MSP

The IC MAX823 is not decoupled by a capacitor to detect small power fault.
74LV74 is decoupled.

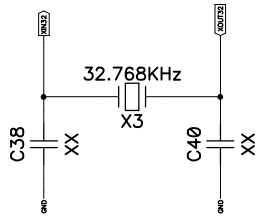


Logic reset for ARM and / or MSP



CDMS – Swisscube Qualification model	Reset	DES	25.09.2007 Crettaz David
		REV	V1.0
HAUTE ECOLE VALAISANNE		5/10	{Path} CDMS_2.0.sch

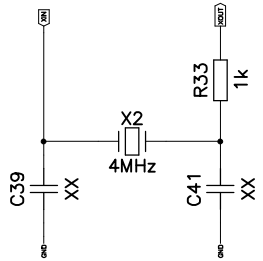
ARM : Crystal at 32.768kHz



As close as possible to
the XIN32 and XOUT32 pins

C45 and C46 will be define during tests

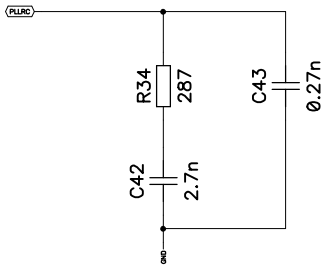
ARM : Crystal at 4 MHz



As close as possible to
the XIN and XOUTpins

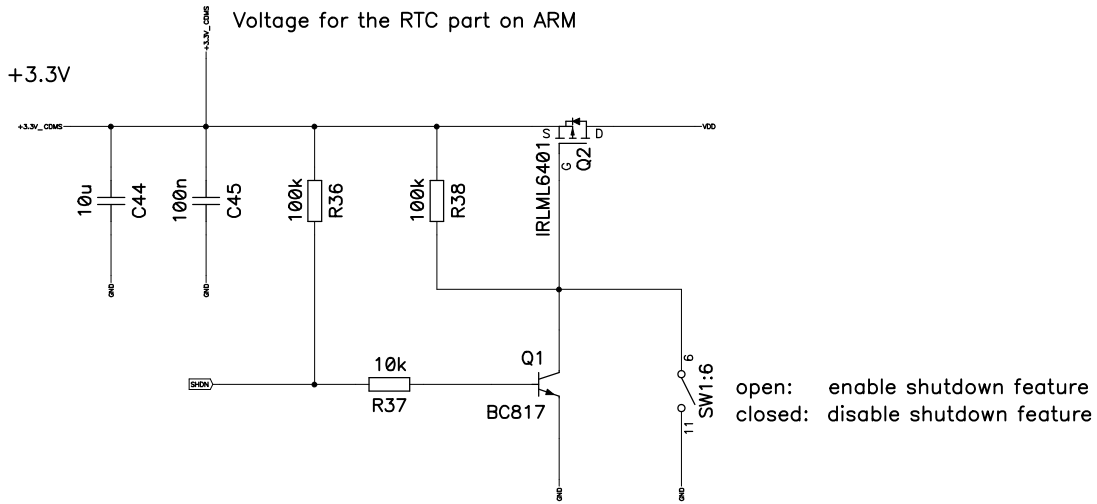
C47 and C48 will be define during tests

ARM : PLL filter for use at 32 MHz



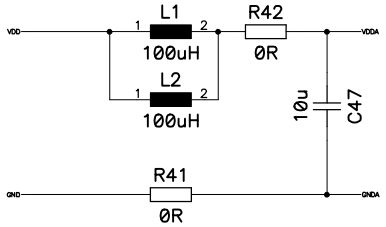
CDMS – Swisscube Qualification model — Clock	DES	25.09.2007 Crettaz David
	REV	V1.0
HAUTE ECOLE VALAISANNE	6/10	{Path} CDMS_2.0.sch

Back up alimentation



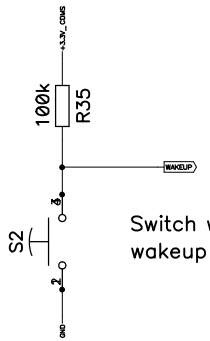
open: enable shutdown feature
closed: disable shutdown feature

Link between Analog and Digital



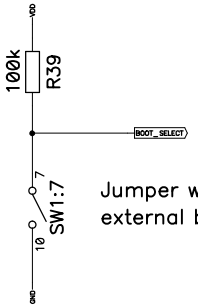
GNDA and VDDA should be connected
with one point to respectively GND and Vdd

Manual Wakeup



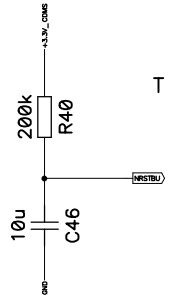
Switch which simulate
wakeup signal

Boot selection jumper



Jumper which simulate
external boot_ select signal

300ms min. reset pulse

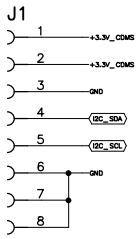


T = RxC (63.8%) = 2s

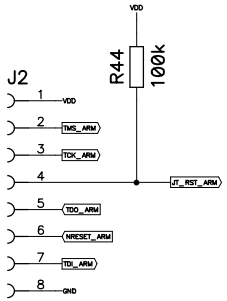
CDMS – Swisscube Qualification model	Backup	DES	25.09.2007 Crettaz David
		REV	V1.0
HAUTE ECOLE VALAISANNE		7/10	{Path} CDMS_2.0.sch

External connectors

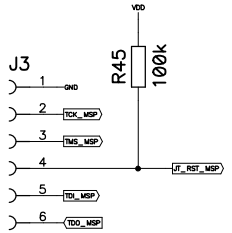
Power / Data



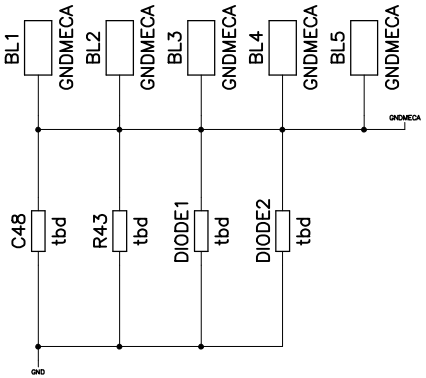
ARM : JTAG connecteur



MSP : JTAG connecteur



Circuit to reduce EMC problems



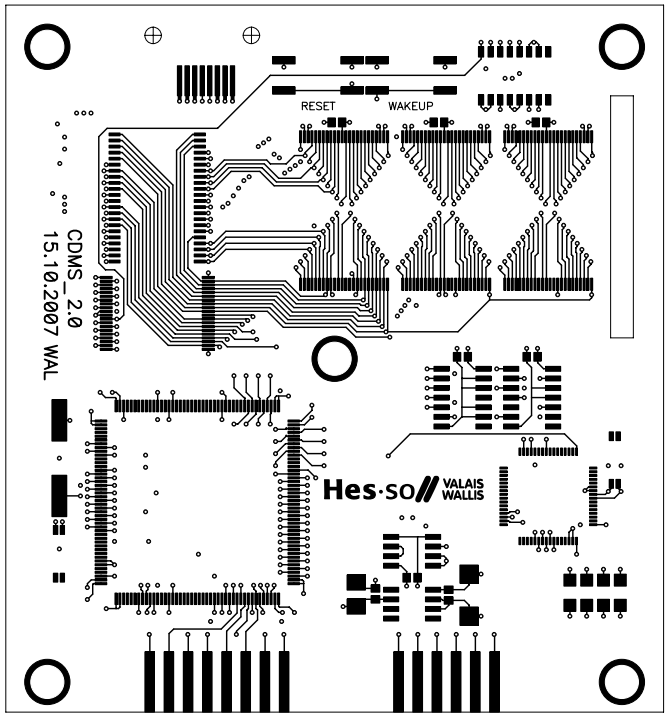
GNDMECA must be connected to the satellite's structure
via the 4 holding screw on the board (mecanical ground)

For the moment it's only necessary to design 4 footprints SMD1206
the value and the components will be calculated during tests

CDMS – Swisscube Qualification model Connectors	DES	25.09.2007 Crettaz David
	REV	V1.0
HAUTE ECOLE VALAISANNE	8/10	{Path} CDMS_2.0.sch

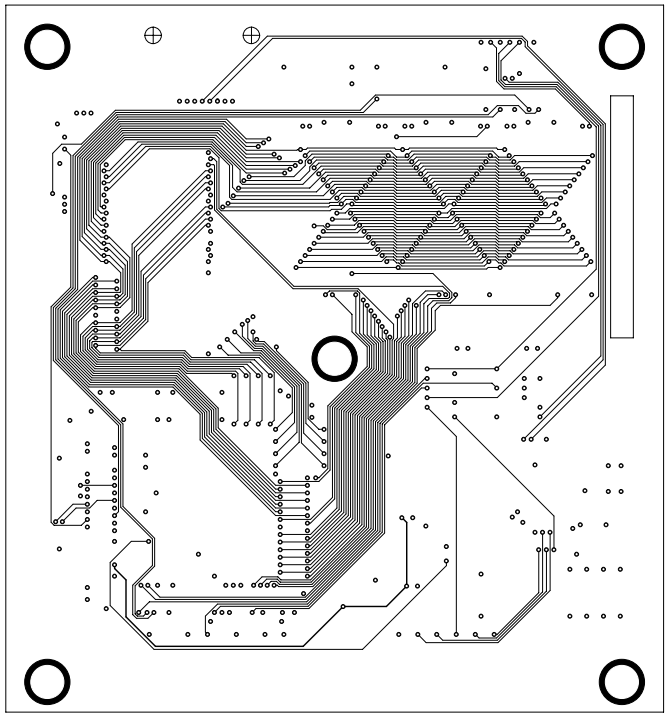
APPENDIX C PCB LAYOUT OF THE CDMS MODEL "QM"

Drill Table			
Hole Dia (mm)	Symbol	Quantity	Plated
0.150	+	478	Yes
2.200	X	2	Yes
4.400	Y	5	Yes



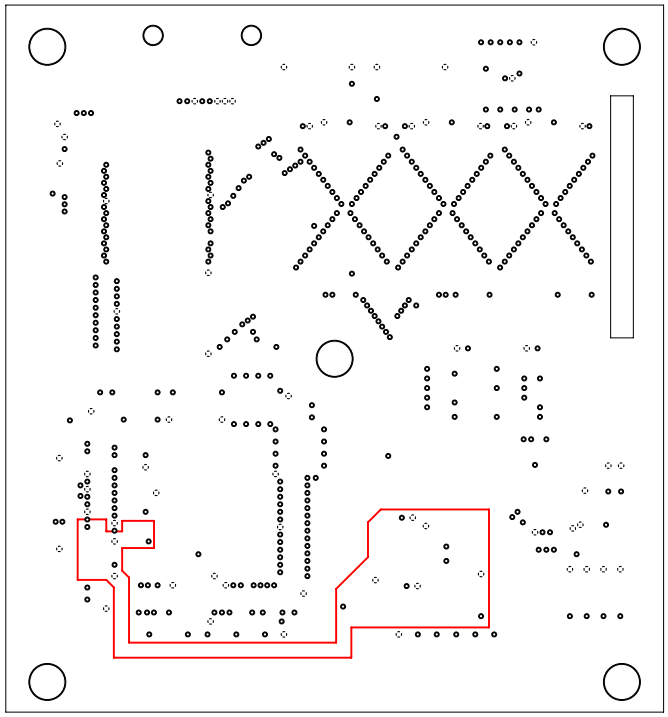
CDMS TOP view	DES	15.10.2007 WAL	
	REV	V2.0	
HAUTE ECOLE VALAISANNE	...\Student\TSTD\Crettaz David CDMS_2.0.pcb		

Drill Table			
Hole Dia (mm)	Symbol	Quantity	Plated
0.150	+	478	Yes
2.200	X	2	Yes
4.400	Y	5	Yes



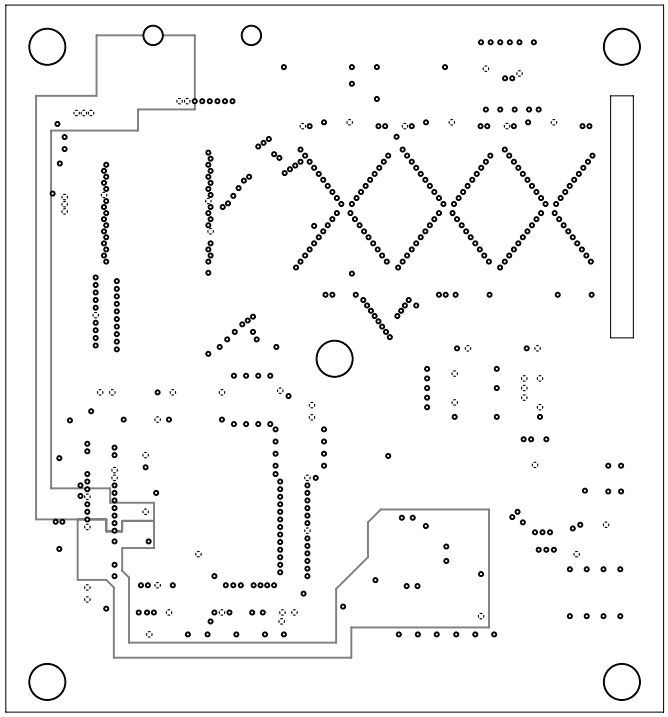
CDMS SIGNAL1 view	DES	15.10.2007 WAL
	REV	V2.0
HAUTE ECOLE VALAISANNE	...\Student\TSTD\Crettaz David CDMS_ 2.0.pcb	

Drill Table			
Hole Dia (mm)	Symbol	Quantity	Plated
0.150	+	478	Yes
2.200	X	2	Yes
4.400	Y	5	Yes



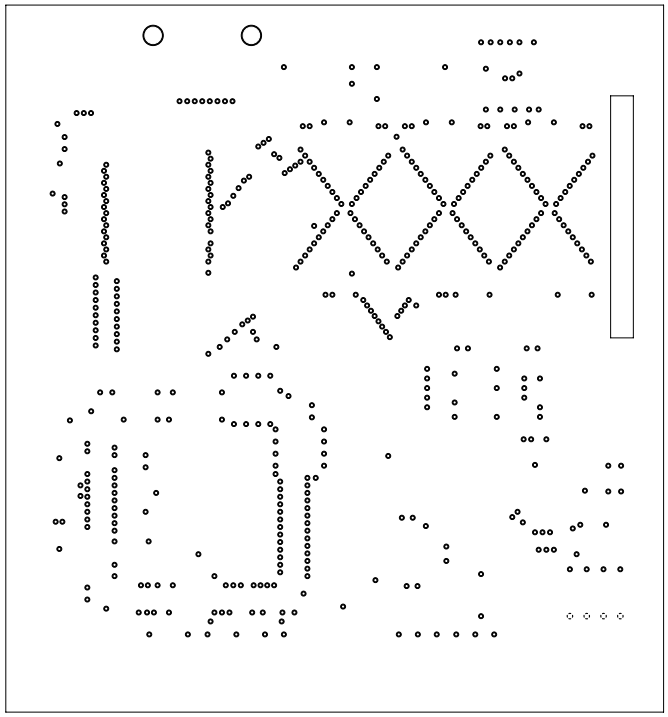
CDMS GND view	DES	15.10.2007 WAL
	REV	V2.0
HAUTE ECOLE VALAISANNE	...\Student\TSTD\Crettaz David CDMS_ 2.0.pcb	

Drill Table			
Hole Dia (mm)	Symbol	Quantity	Plated
0.150	+	478	Yes
2.200	X	2	Yes
4.400	Y	5	Yes



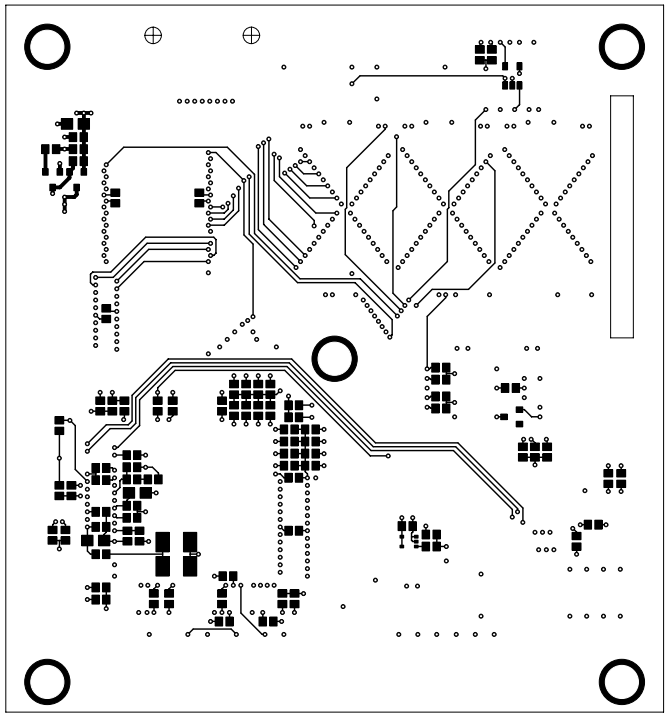
CDMS VCC view	DES	15.10.2007 WAL
	REV	V2.0
HAUTE ECOLE VALAISANNE	...\Student\TSTD\Crettaz David CDMS_ 2.0.pcb	

Drill Table			
Hole Dia (mm)	Symbol	Quantity	Plated
0.150	+	478	Yes
2.200	X	2	Yes
4.400	Y	5	Yes



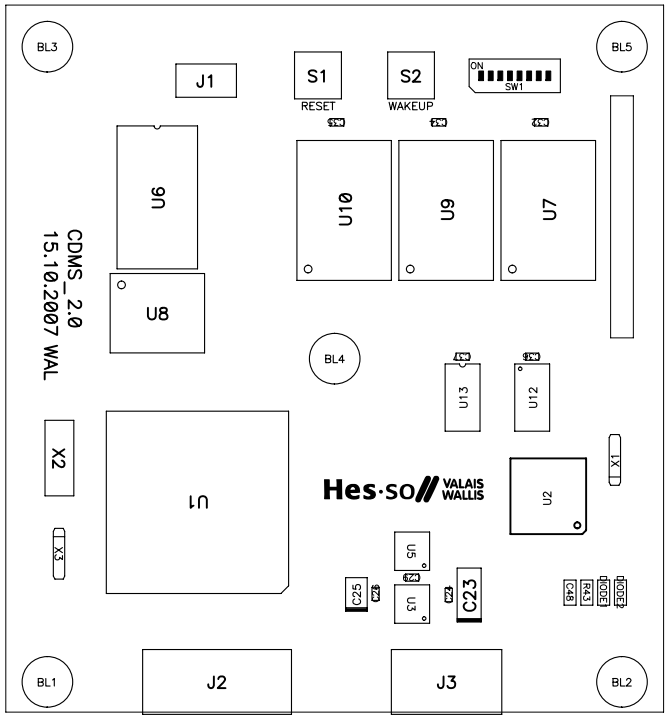
CDMS GNDMECCA view	DES	15.10.2007 WAL
	REV	V2.0
HAUTE ECOLE VALAISANNE	...\Student\TSTD\Crettaz David CDMS_ 2.0.pcb	

Drill Table			
Hole Dia (mm)	Symbol	Quantity	Plated
0.150	+	478	Yes
2.200	X	2	Yes
4.400	Y	5	Yes



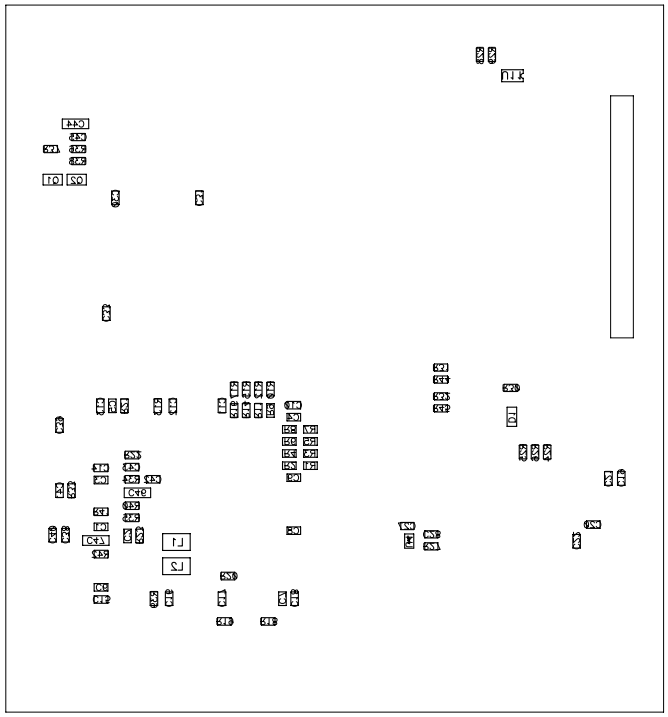
CDMS BOTTOM view	DES	15.10.2007 WAL
	REV	V2.0
HAUTE ECOLE VALAISANNE	...\Student\TSTD\Crettaz David CDMS_ 2.0.pcb	

Drill Table			
Hole Dia (mm)	Symbol	Quantity	Plated
0.150	+	478	Yes
2.200	X	2	Yes
4.400	Y	5	Yes



CDMS TOP ASSY view	DES	15.10.2007 WAL
	REV	V2.0
HAUTE ECOLE VALAISANNE	...\Student\TSTD\Crettaz David CDMS_2.0.pcb	

Drill Table			
Hole Dia (mm)	Symbol	Quantity	Plated
0.150	+	478	Yes
2.200	X	2	Yes
4.400	Y	5	Yes

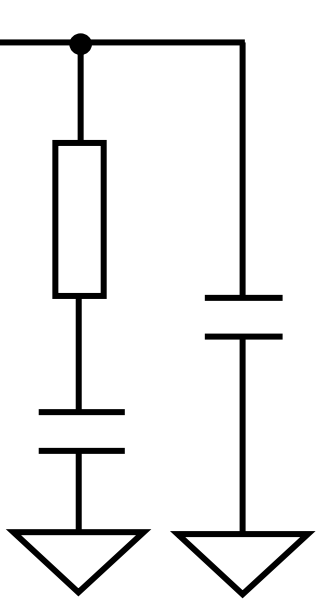


CDMS BOTTOM ASSY view	DES	15.10.2007 WAL
	REV	V2.0
HAUTE ECOLE VALAISANNE	...\Student\TSTD\Crettaz David CDMS_ 2.0.pcb	

APPENDIX D PLL AUTOMATIC FILTER CALCULATION

AT91M55800A PLL LOOP FILTER

PLLRC



Suplly Voltage	R1	Unit
At 2,4 V	1070	ohms
At 3,3 V	287	ohms
At 3,6 V	158	ohms

Spply Voltage	C1	Unit
At 2,4 V	0.68	nF
At 3,3 V	2.7	nF
At 3,6 V	4.7	nF

Spply Voltage	C2	Unit
At 2,4 V	0.068	nF
At 3,3 V	0.27	nF
At 3,6 V	0.47	nF

AT91M55800A PLL PARAMETERS

Note: Only orange colored fields can be modified. All other have been write-protected.

User parameters :

Capturing the three user fields

Parameters	Frequency value	Unit
Set the Low Clock Frequency	32768	Hz
Set the Main Clock Frequency	4	MHz
Set the Master Clock Frequency	32	MHz

Component values :

	At 2,4 V	At 3,3 V	At 3,6 V	Unit
R1 value	1070	287	158	Ohms
C1 value	0.68	2.7	4.7	nF
C2 value	0.068	0.27	0.47	nF

Register values :

The results must be programmed in the microcontroller registers.
With these timer values, the overshooting frequency stays lower than 10 %.

	At 2,4 V	At 3,3 V	At 3,6 V	Unit
Value (MUL) in APMC-CGMR register	7	7	7	decimal
Delay	55	27	19	µs
Timer value: PLLCOUNT user field	3	2	2	decimal

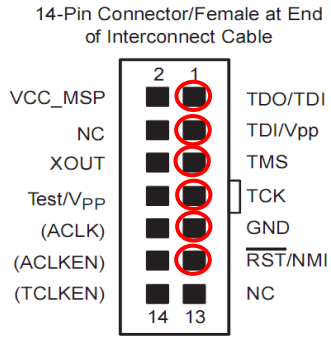
APPENDIX E SOLDERING OF THE JTAG CABLES

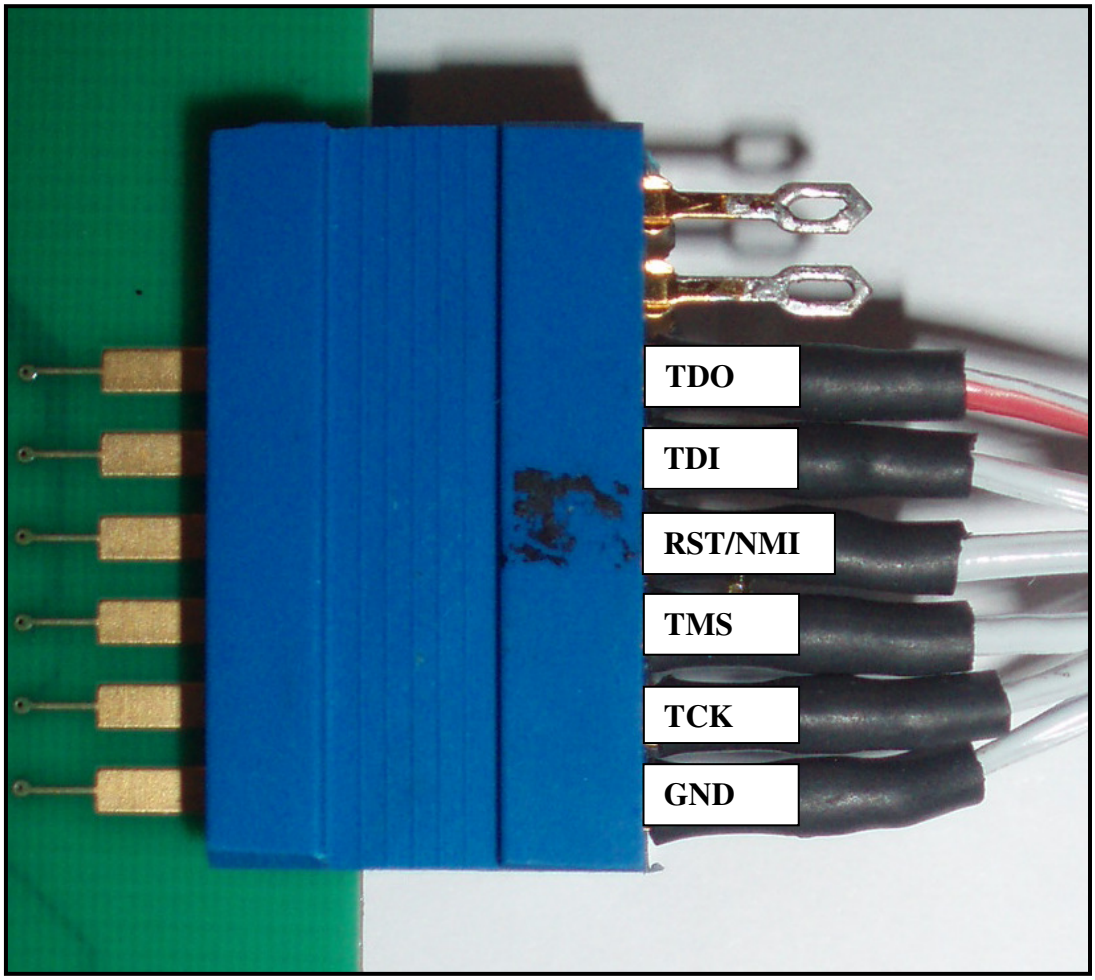
This APPENDIX described the soldering of the JTAG programming connector to their interface

- **For the MSP-FET430PIF interface**

To the parallel PORT (PC)

To the interconnect cable (TARGET)

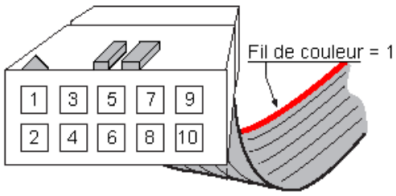




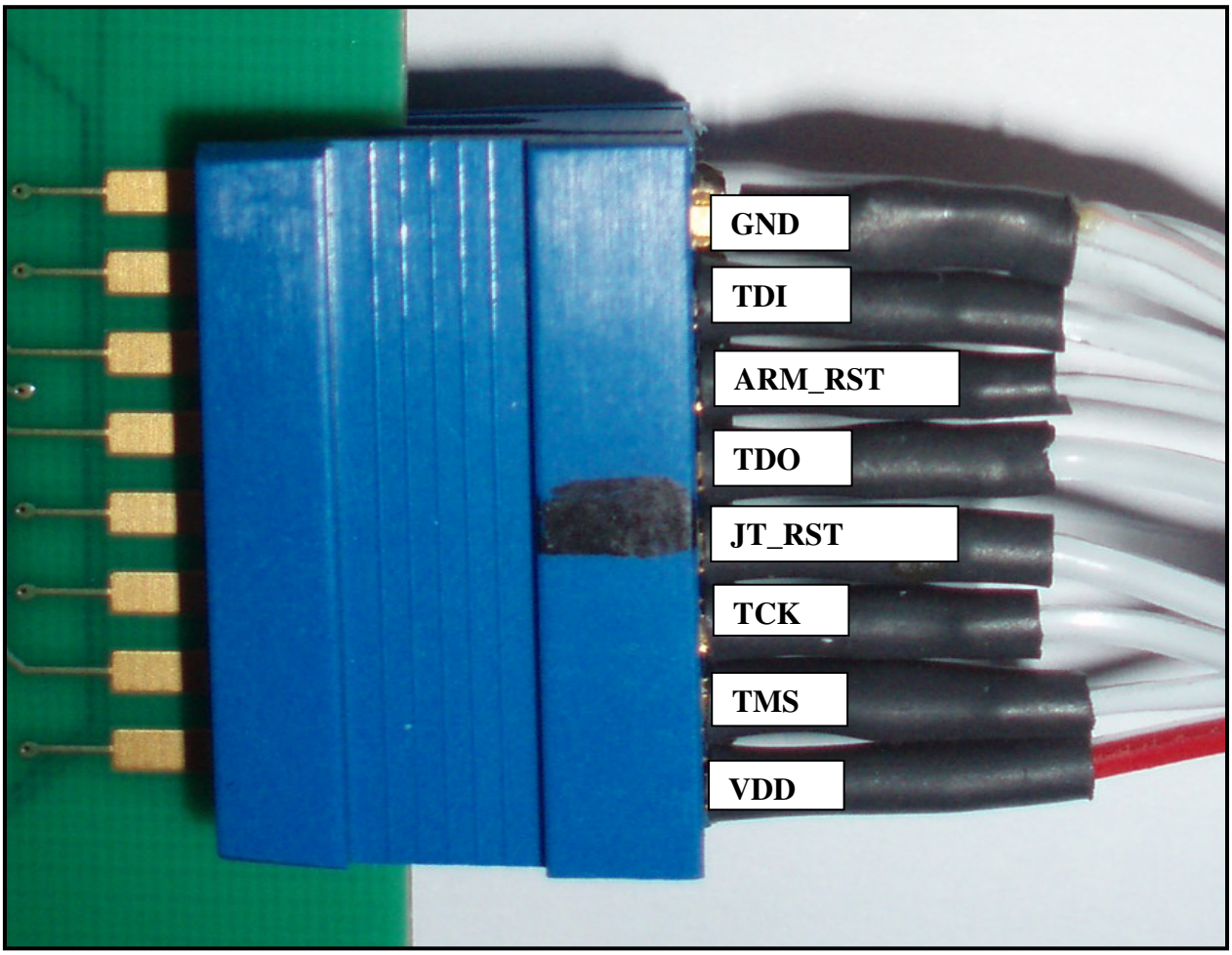
- **For the JTAGkey interface**

To the USB PORT (PC)

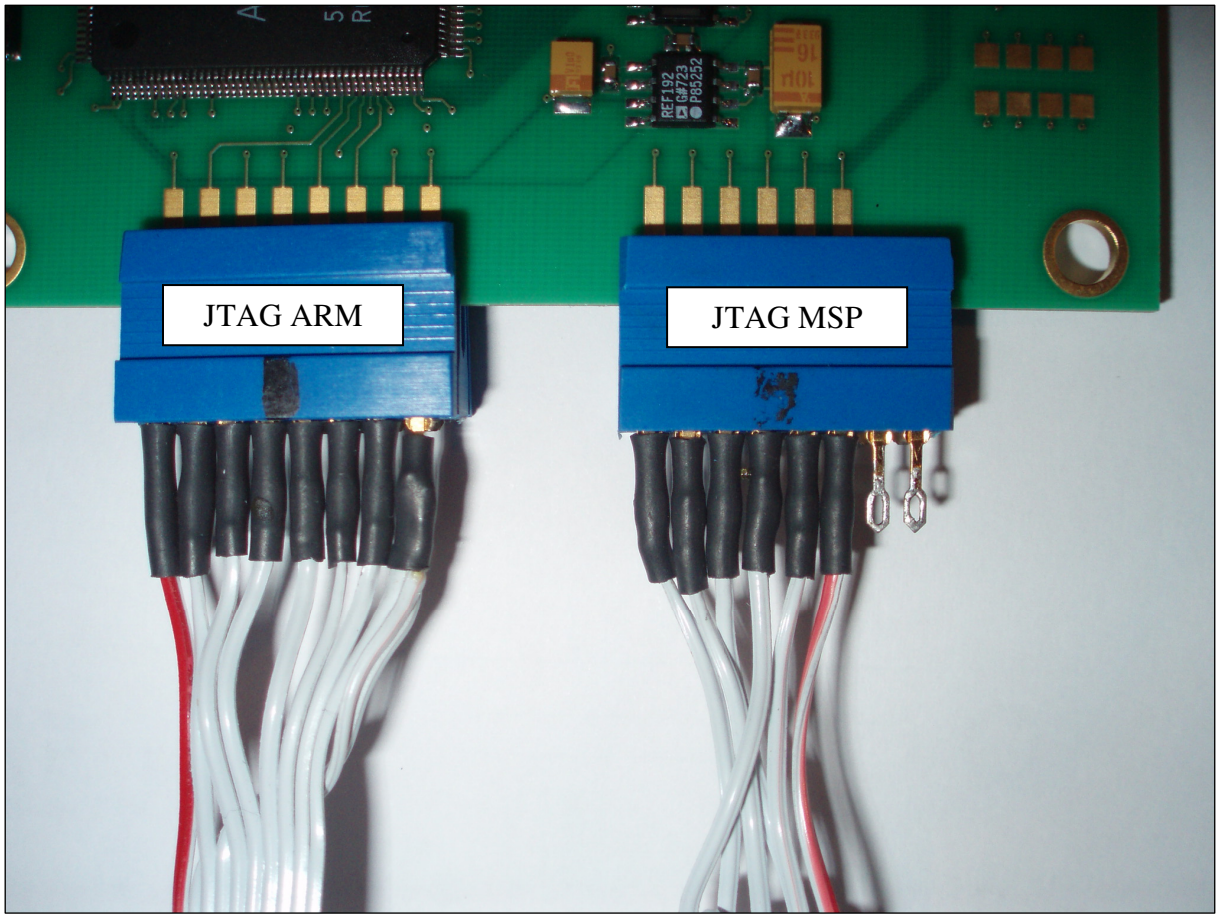
To the interconnect cable (PC)

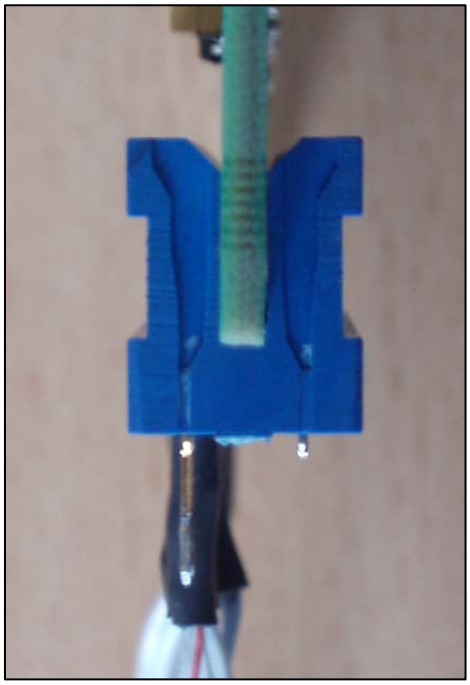


Pin_1	VDD
Pin_2	NC
Pin_3	TMS
Pin_4	NC
Pin_5	TCK
Pin_6	JT_RST
Pin_7	TDO
Pin_8	ARM_RST
Pin_9	TDI
Pin_10	GND



- **The following pictures shows how to connect these two cables on the CDMS board**





APPENDIX F HOW TO USE ECLIPSE FOR ARM PROJECTS

!!Before to start this tutorial is sure you have properly installing the following programs!!

OpenOCD

YAGARTO GNU ARM toolchain

These programs could be downloaded at the following address:

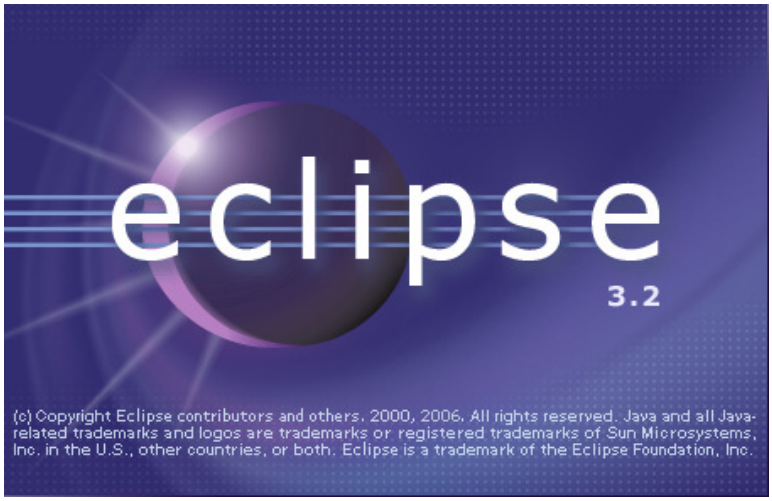
www.yagarto.de

JAVA Runtime Environment 1.5 or higher

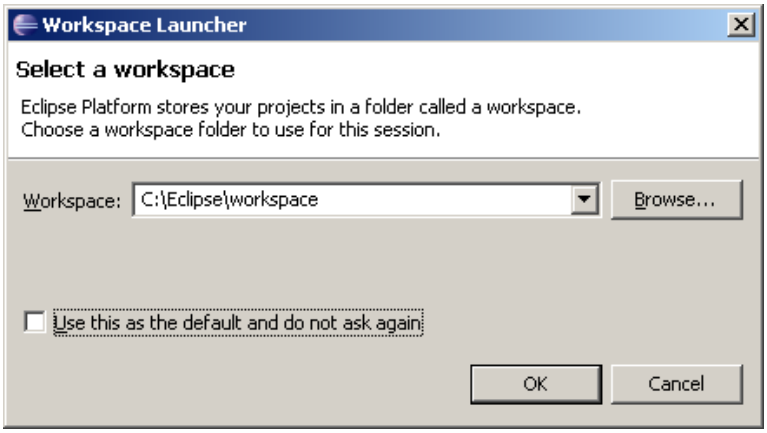
This application could be downloaded at the following address:

http://java.sun.com/javase/downloads/index.jsp

After the start of Eclipse the following splash screen is presented:

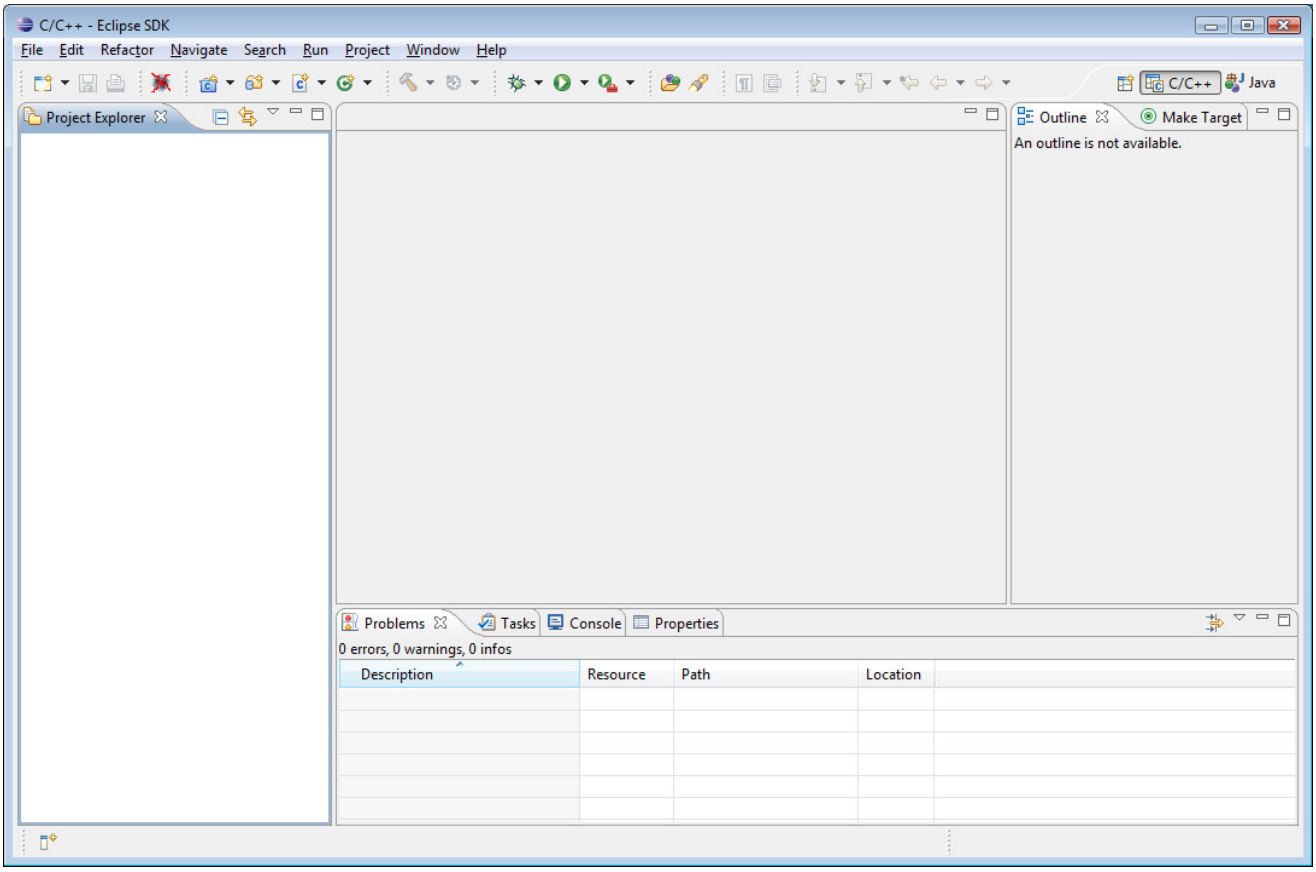


If not, you have a problem with the Java Runtime Environment. After a while Eclipse want to know where to store your projects:

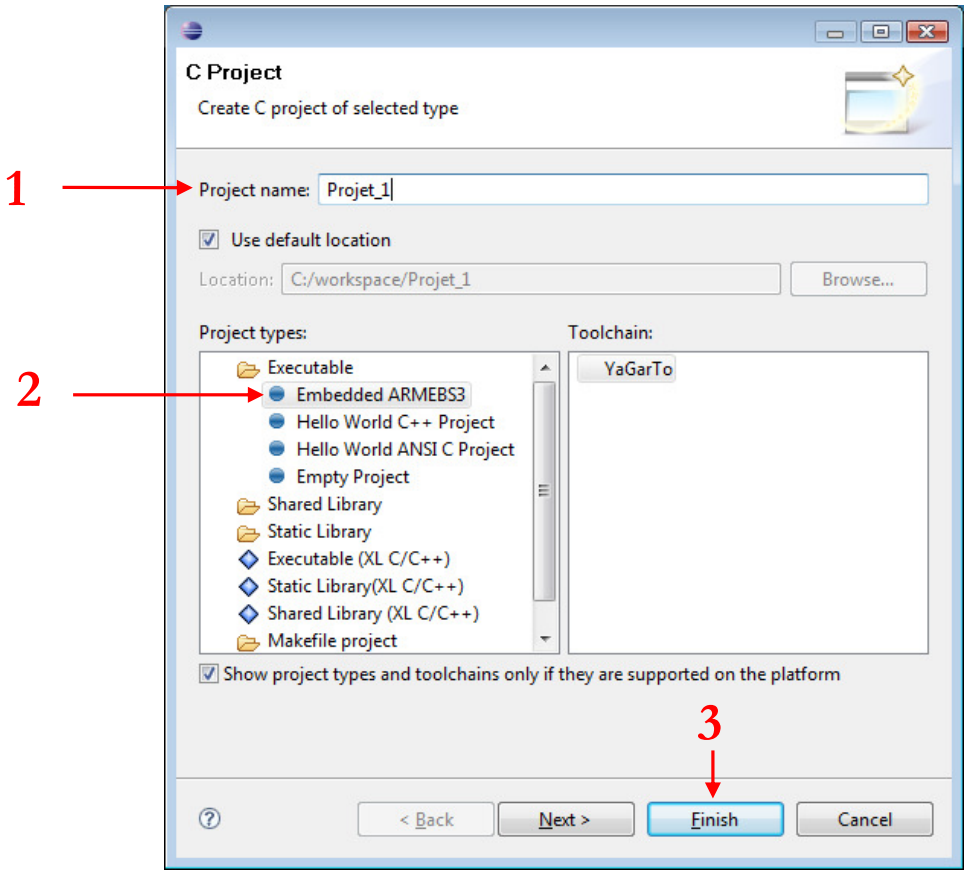


For the Eclipse workspace the folder C:\eclipse\workspace will be used. Enter the directory and press "OK". The following "Welcome" window will appear:

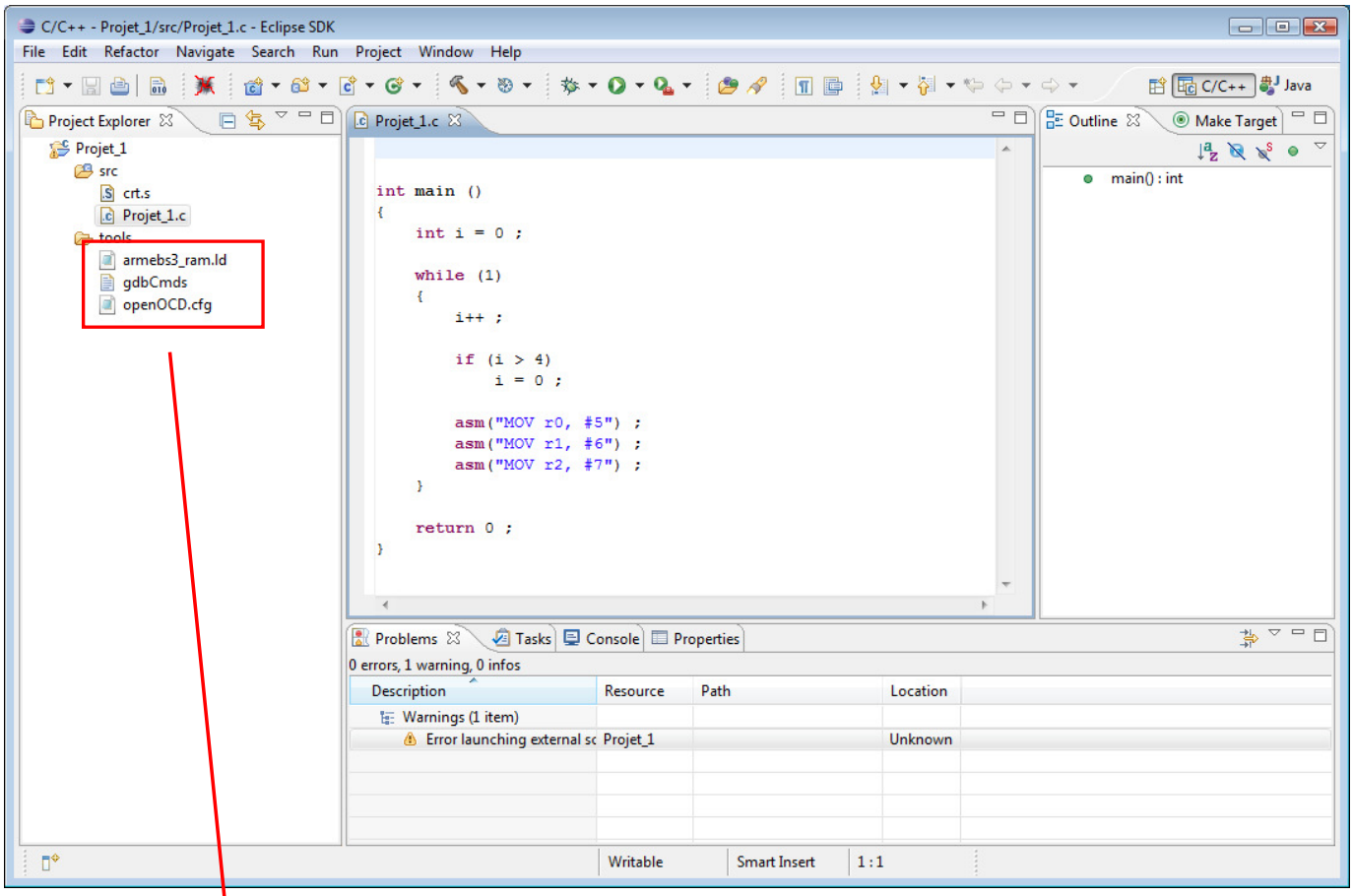
Now the Eclipse environment appears

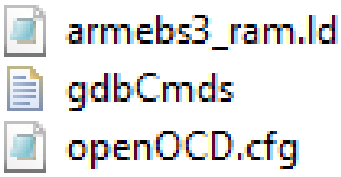


Click on "File-> New -> C Project" and the following screen appears



You have created your first project

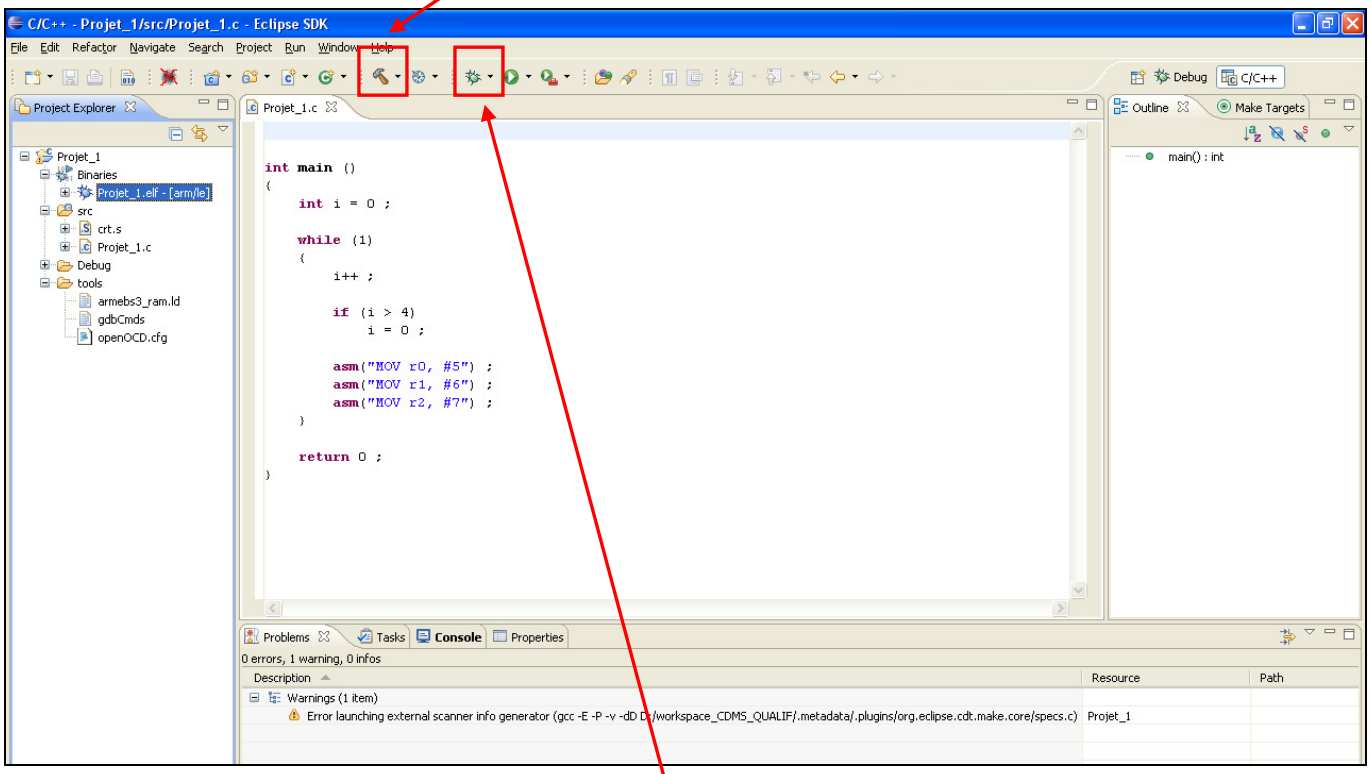




We just need to change these three files with the file contain in the CDROM in the folder…

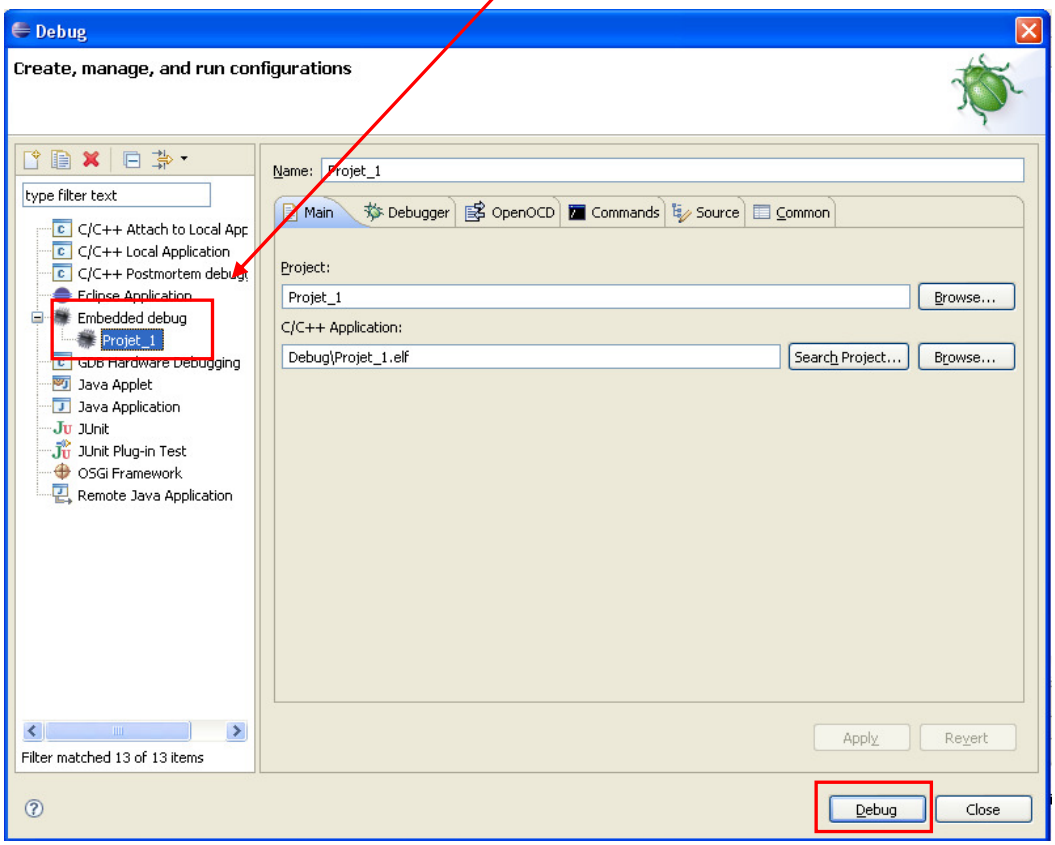
It's the entire configuration file needed for the ARM7 microcontroller

Now you can build the project to generate the ELF file



Click on the debug button

Right click on “Embedded debug->new”



You have made all steps!

Press the “Debug” button to start the debugger and debug your program

For further information about the use or the installation of the Eclipse Development environment:

- See at the following address :

  www.yagarto.de
- See the great PDF tutorial contains in the following zip file:

  http://www.atmel.com/dyn/resources/prod_documents/atmel_tutorial_source.zip

APPENDIX G HOW TO USE CODE COMPOSER ESSENTIALS

!!Before to start this tutorial is sure you have properly installing the following program!!

Code Composer Essentials

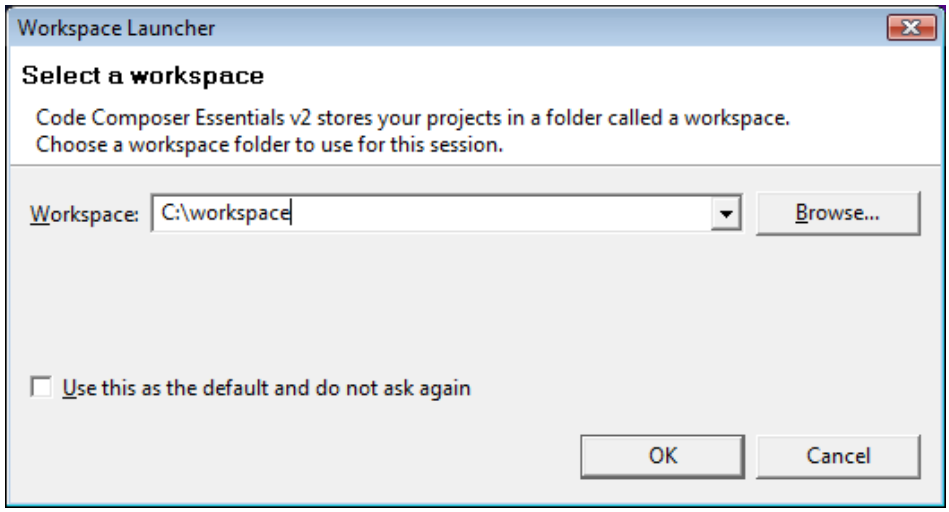
This program could be downloaded at the following address:

http://focus.ti.com/docs/toolsw/folders/print/msp-cce430.html

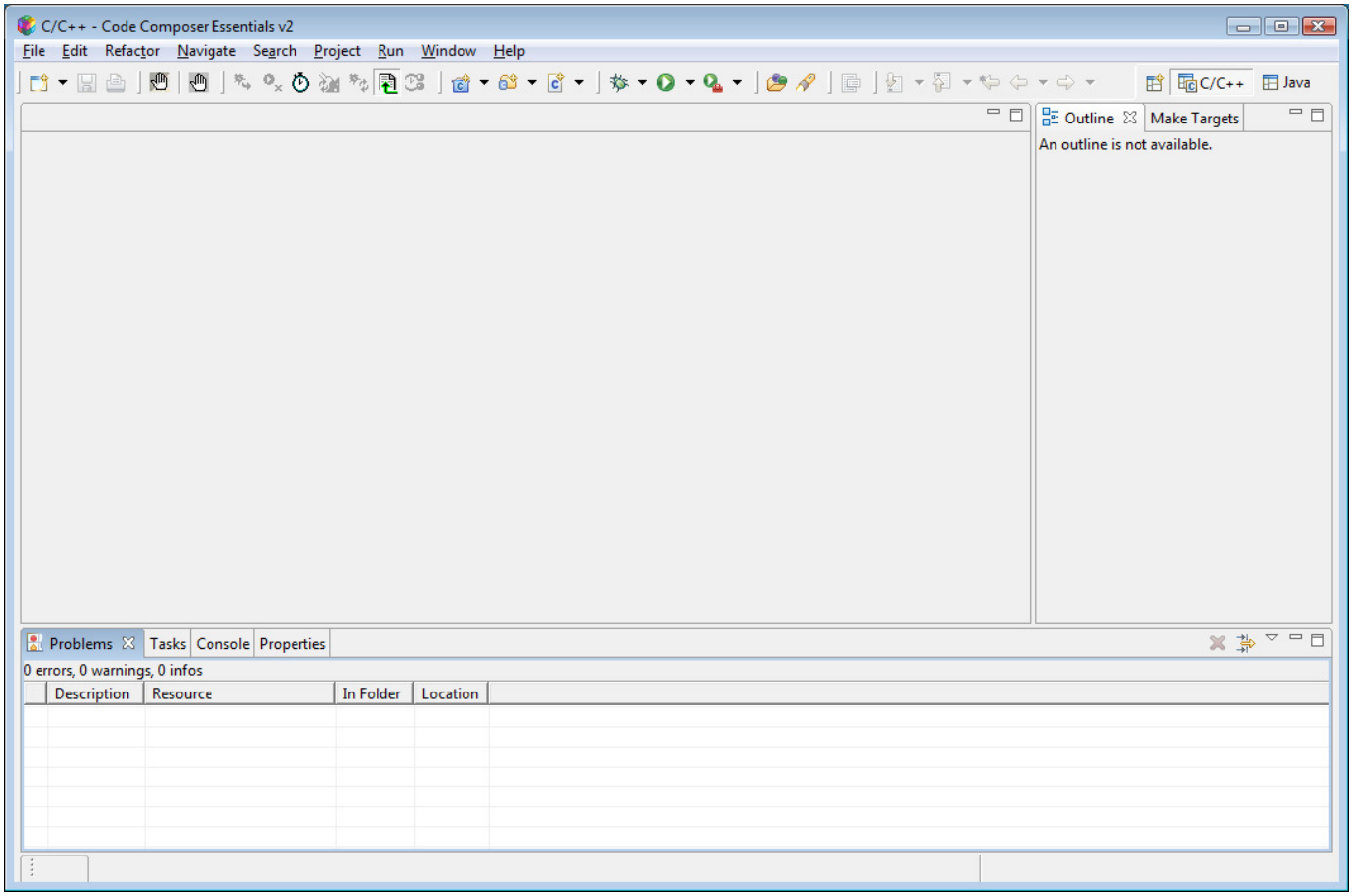
---

After the start of Code Composer Essentials the following screen appears

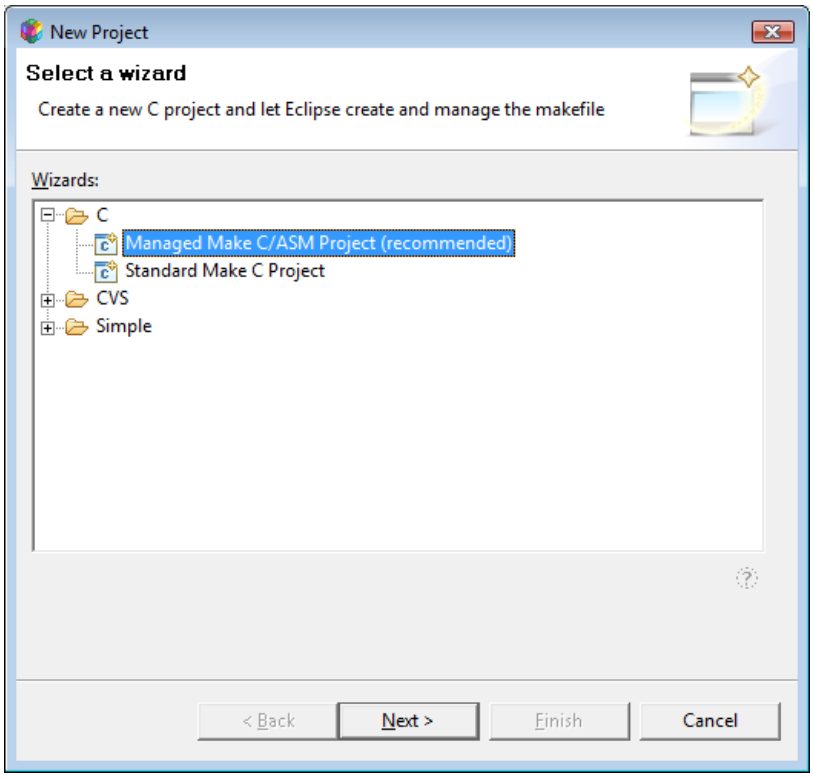
After a moment Code Composer Essential want to know where to store your projects



Now the Code Composer Essential environment appears

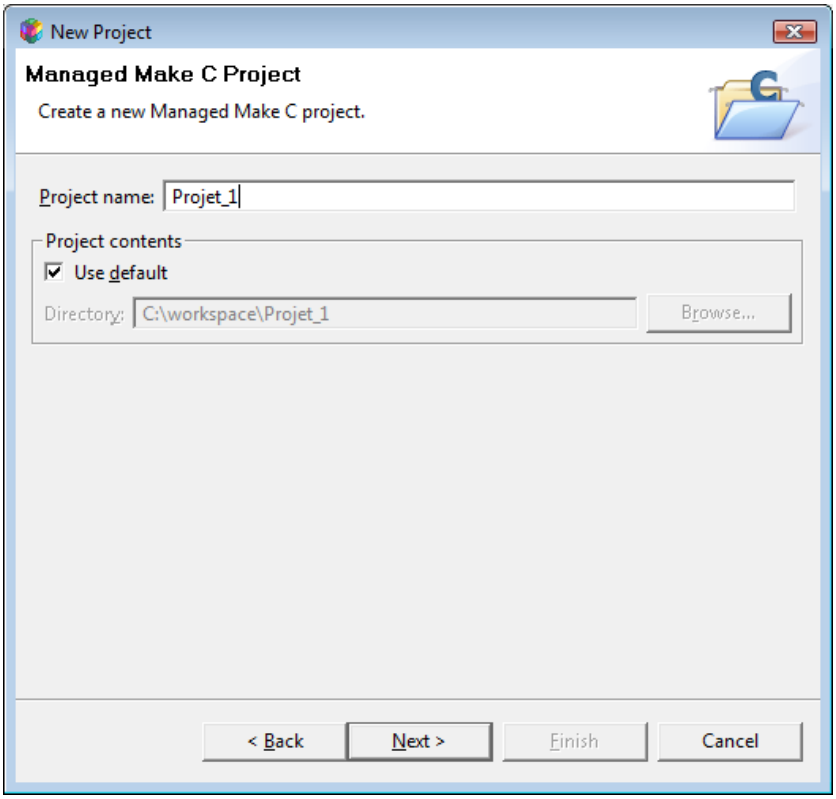


Click on "File-> New -> Project" and the following screen appears

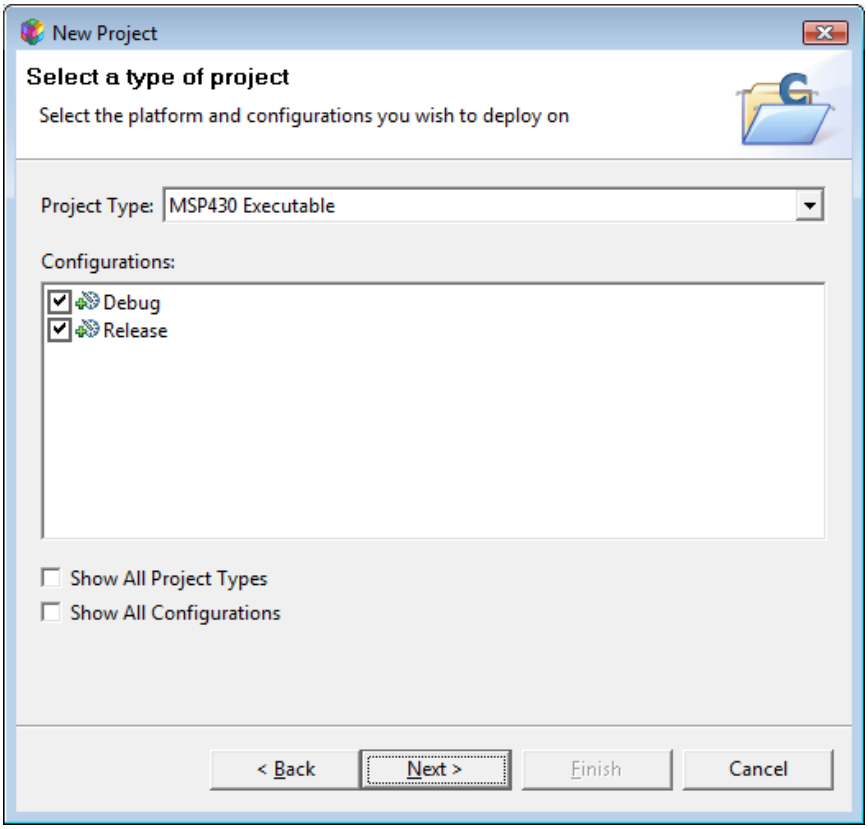


Select the highlighted field and click "Next"

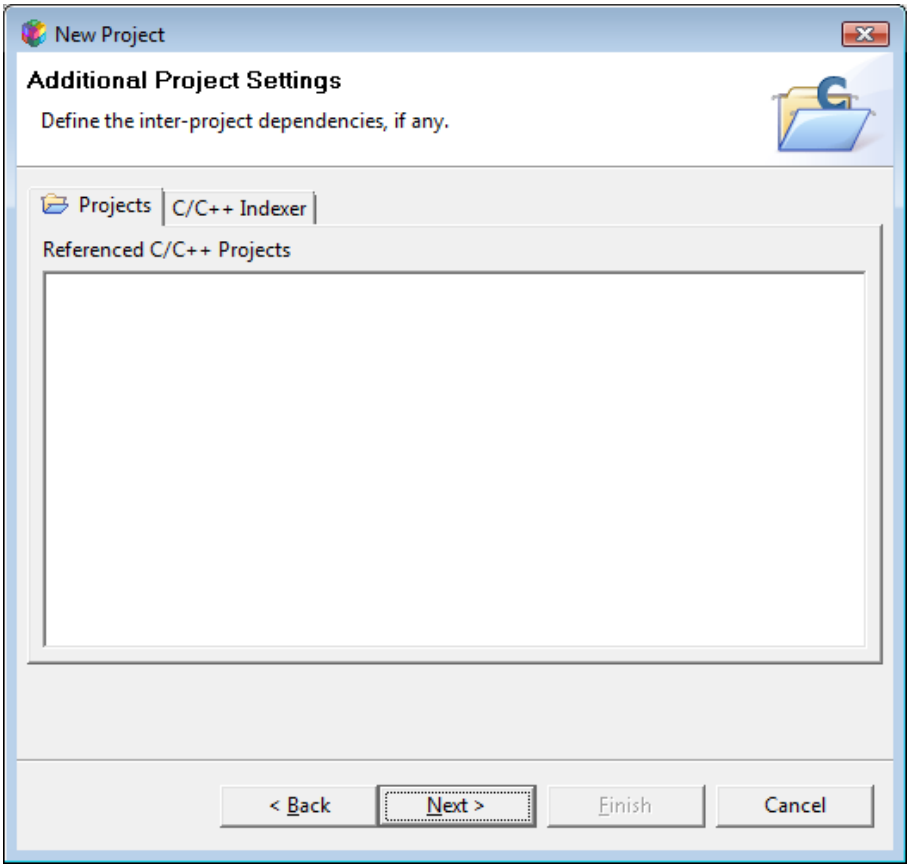
Put a name for your project and click “Next”



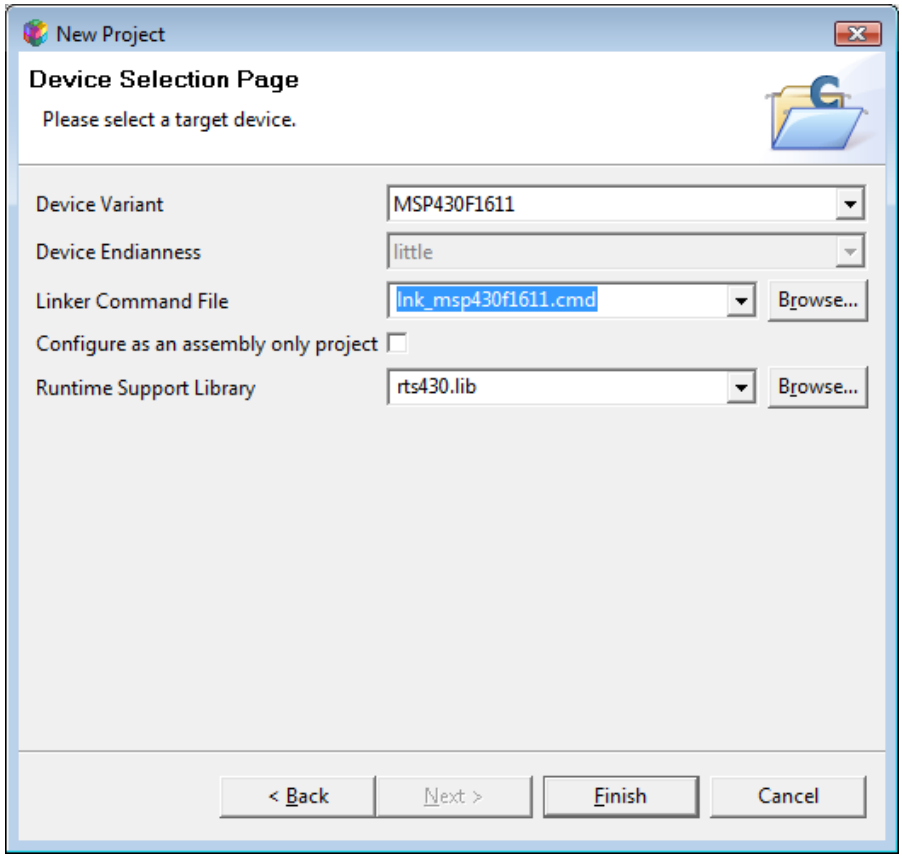
Select the same options and click “Next”



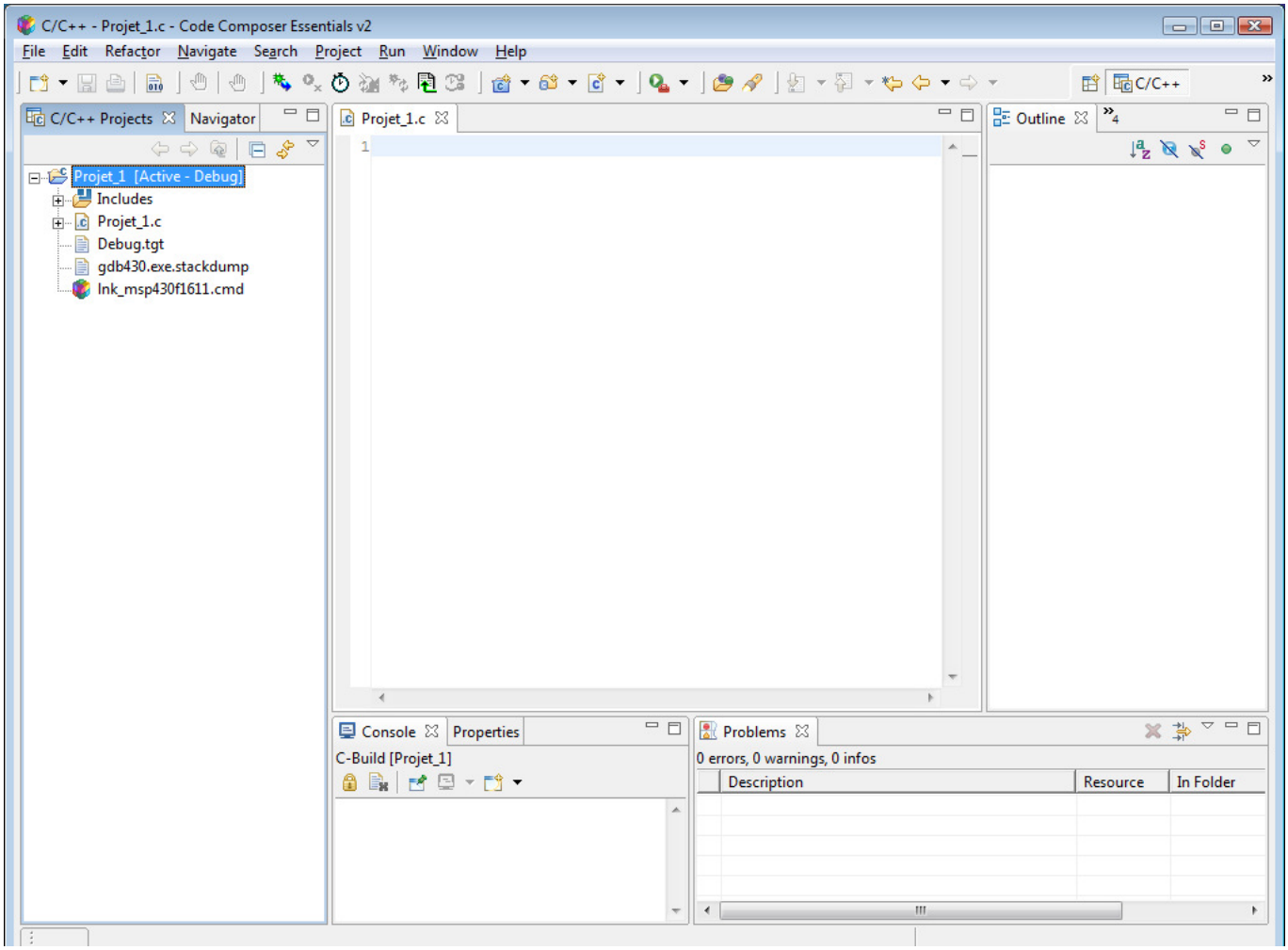
Just click “Next”



Select the same options and click “Finish”



You have created your first project



You can write your program and:

1. Click Project->Clean
2. Click Project->Build Project
3. Click Run->Debug Active Project

You can now debug you project

For further information about the use or the installation of the Code Composer Essentials:

- See the documentation included with the program

APPENDIX H SOURCE FILES FOR THE ARM

CDMS.h

```
//*-----------------------------------------------------------------------------*/
//* File Name               : CDMS.h                                            */
//* Object                  : CDMS Prototype Board Features Definition File.*/
//*                                                                             */
//* 1.0 17/10/06  TAP       : Creation from example eb55.h.                     */
//*-----------------------------------------------------------------------------*/

#ifndef CDMS_h
#define CDMS_h

#include "AT91M55800A.h"
#define __inline inline

/*-------------------------------------*/
/* CDMS External Memories Definition */
/*-------------------------------------*/

/* Flash Memory NCS0: S29AL016D 1M*16 */
#define FLASH0_BASE     (0x10000000)
#define FLASH_SIZE      (2*1024*1024)           /* byte */

/* Flash Memory NCS1: S29AL016D 1M*16 */
#define FLASH1_BASE     (0x20000000)
#define FLASH_SIZE      (2*1024*1024)           /* byte */

/* Flash Memory NCS2: S29AL016D 1M*16 */
#define FLASH2_BASE     (0x30000000)
#define FLASH_SIZE      (2*1024*1024)           /* byte */

/* SDRAM Memory NCS3: R1LV0416C 256k*16 */
#define EXT_SRAM_BASE   (0x40000000)
#define EXT_SRAM_SIZE   (2*256*1024)            /* byte */

/*-----------------------------------------------------------------------------*/
/* EBI Initialization Data                                                     */
/*-----------------------------------------------------------------------------*/
/* The EBI User Interface Image which is copied by the boot.                   */
/* 4 MHz master clock assumed.                                                 */
/* That's hardware! Details in the Electrical Datasheet of the AT91 device.    */
/* EBI Base Address is added at the end for commodity in copy code.            */
/*-----------------------------------------------------------------------------*/

/* 16bits data bus, 4 wait states, 1 TDF, only 16bit capable                   */
#define EBI_CSR_0       ((unsigned int)(FLASH0_BASE | 0x32B1 ))

/* 16bits data bus, 4 wait states, 1 TDF, only 16bit capable                   */
#define EBI_CSR_1       ((unsigned int)(FLASH1_BASE | 0x32B1))

/* 16bits data bus, 4 wait states, 1 TDF, only 16bit capable                   */
#define EBI_CSR_2       ((unsigned int)(FLASH2_BASE | 0x32B1))

/* 16bits data bus, 3 wait states, 1 TDF, only 16bit capable                   */
#define EBI_CSR_3       ((unsigned int)(EXT_SRAM_BASE | 0x32BD))

/* The unused Chip Select must be configured with a different base address */
#define EBI_CSR_4       ((unsigned int)0x50000000)              /* unused */
#define EBI_CSR_5       ((unsigned int)0x60000000)              /* unused */
#define EBI_CSR_6       ((unsigned int)0x70000000)              /* unused */
#define EBI_CSR_7       ((unsigned int)0x80000000)              /* unused */

#endif /* CDMS_h */
```

cdmsQualifv1.c

```c
#include "AT91M55800A.h"
#include "CDMS.h"

//Declaration of functions
void lowLevelInit(void);
void initSPI(void);
void entry_flash_cfi_qry(void);
void initADC0(void);
void initIRQ() ;
void IRQHandler (void)   __attribute__ ((interrupt("IRQ")));

//Declaration of variable
char receiveData[2]={0x00,0x00};

void lowLevelInit(void)
{
    //EBI configutation
    AT91PS_EBI     pEbi;
    pEbi = AT91C_BASE_EBI ;

    pEbi->EBI_CSR[0] = EBI_CSR_0 ;
    pEbi->EBI_CSR[1] = EBI_CSR_1 ;
    pEbi->EBI_CSR[2] = EBI_CSR_2 ;
    pEbi->EBI_CSR[3] = EBI_CSR_3 ;
    pEbi->EBI_CSR[4] = EBI_CSR_4 ;
    pEbi->EBI_CSR[5] = EBI_CSR_5 ;
    pEbi->EBI_CSR[6] = EBI_CSR_6 ;
    pEbi->EBI_CSR[7] = EBI_CSR_7 ;

    //APMC configuration
    AT91PS_APMC    pAPMC;
    pAPMC = AT91C_BASE_APMC;

    //Disable all peripheral clocks
    pAPMC->APMC_PCDR = 0xFFFFFFFF;

    //Enable Master Clock
    pAPMC->APMC_CGMR = AT91C_APMC_MOSCEN  //Enable main oscillator
                     | AT91C_APMC_OSCOUNT;//Setup the oscillator counter

    // Reading the APMC Status register to detect when the oscillator is stabilized
    while ( (pAPMC->APMC_SR & AT91C_APMC_MOSCS) != AT91C_APMC_MOSCS ) {} ;

    // Commuting from Slow Clock to Main Oscillator (4Mhz)
    pAPMC->APMC_CGMR |= (0x1<< 14);

    //Enable PLL
    pAPMC->APMC_CGMR |= (0x7<< 8)    //MUL=7 =>PLL=32Mhz
                       |(0x2<< 24); //PLLCOUNT=2

    //Read PLL status until he's stabilized
    while ( (pAPMC->APMC_SR & AT91C_APMC_LOCK) != AT91C_APMC_LOCK ) {} ;

    // Commuting from 4Mhz to PLL @ 32MHz
    pAPMC->APMC_CGMR  = (AT91C_APMC_CSS & (0x02 << 14)) | (pAPMC->APMC_CGMR &
 ~AT91C_APMC_CSS);

    // Now the Master clock is the output of PLL at 32MHz
}
```

```
void initSPI(void)
{
    //APMC base address
    AT91PS_APMC    pAPMC;
    pAPMC = AT91C_BASE_APMC;

    //PIOA base address
    AT91PS_PIO pPIO;
    pPIO = AT91C_BASE_PIOA;

    //SPI base address
    AT91PS_SPI pSPI;
    pSPI = AT91C_BASE_SPI;

    //Enable SPI clock
    pAPMC->APMC_PCER |=AT91C_APMC_SPI;

    //Disable PIO pin -> enable peripheral pin for SPI mode 3 pin
    pPIO->PIO_PDR |= (0x7<<23);

    //Define PA26/NCS0 as open-drain
    pPIO->PIO_MDER = (0x1<<26);

    //Disable interrupt
    pSPI->SPI_IDR = 0x3F;

    //Reset SPI
    pSPI->SPI_CR = (0x1<<7);

    //Enable interrupt RDRF and TDRE
    pSPI->SPI_IER = 0x3;

    //Configure the SPI mode register
    pSPI->SPI_MR = AT91C_SPI_MSTR     //SPI in Master Mode
                 | AT91C_SPI_PS_FIXED //Fixed Peripheral Select
                 | AT91C_SPI_DIV32    //SPI Master Clock = MCK / 32
                 | (0xE<<16);         //Select NPCS0 for Fixed Peripheral Select

    //Configure the Chip Select 0 (NCS0)
    pSPI->SPI_CSR[0] = AT91C_SPI_CPOL  //Clock polarity for inactive state of SPCK = 1
                     | (0x1<<1)        //Clock phase = 1
                     | AT91C_SPI_BITS_8//Bits per transfer
                     | (0x10<<8);      //Define the Serial Clock Baud Rate
                                       //(31.25KHz, with MCK=4MHz ->0x2)
                                       //(31.25KHz, with MCK=32MHz ->0x10)

    //Enable SPI Interrupt
    pSPI->SPI_IER = 0x3F;

    //Enable spi
    pSPI->SPI_CR = (0x1<<0);
}
```

```
void entry_flash_cfi_qry(void)
{
    unsigned short *cfiQryAd0, *qry;
    char q,r,y;

    //Write the command 0x98 to the address 0xAA in flash0
    cfiQryAd0 =(unsigned short *) (FLASH0_BASE | 0xAA);
    *cfiQryAd0=0x98;

    //CFI Query identification string
    //Check the string "QRY" on FLASH0
    qry=(unsigned short *) (FLASH0_BASE | 0x20);
    q=*qry;
    qry=(unsigned short *) (FLASH0_BASE | 0x22);
    r=*qry;
    qry=(unsigned short *) (FLASH0_BASE | 0x24);
    y=*qry;
}

void initADC0(void)
{
    //APMC base address
    AT91PS_APMC    pAPMC;
    pAPMC = AT91C_BASE_APMC;

    //ADC0 base address
    AT91PS_ADC pADC0;
    pADC0 = AT91C_BASE_ADC0;

    float tempDeg[20];
    int i,j,tempValue[20];
    i=j=0;

    //Initialize the tables
    for(i=0;i<20;i++)
    {
        tempDeg[i]=0;
        tempValue[i]=0;
    }

    //Reset ADC0
    pADC0->ADC_CR = AT91C_ADC_SWRST;

    //Setup ADC0
    pADC0->ADC_MR = AT91C_ADC_TRGEN_DIS         //Software trigger
                  | AT91C_ADC_RES_10_BIT        //10 bit resolution
                  | AT91C_ADC_SLEEP_NORMAL_MODE //Normal mode the ADC stay always ON
                  | (0x34<<8);                  //PRESCALER = 6 (0x06)
                                                //so the frequency is 300kHz
                                                //(with MCK=4MHz)
                                                //PRESCALER = 52 (0x34)
                                                //so the frequency is 300kHz
                                                //(with MCK=32MHz)

    //Enable ADC0 channel
    pADC0->ADC_CHER = AT91C_ADC_CH0;

    //Enable ADC0 peripheral clock
    pAPMC->APMC_PCER = AT91C_APMC_ADC0 | AT91C_APMC_ADC1;
```

```
        for(i=0;i<20;i++)
    {
        //Start an ADC conversion
        pADC0->ADC_CR = AT91C_ADC_START;

        //Check the end of conversion status bit
        while((pADC0->ADC_SR & AT91C_ADC_EOC0 ) == 0x0) {};

        //Simulate a trigger of about 5 sec
        while(j<6000000)
        {
            j++;
        }

        j=0;

        //Read the value
        tempValue[i] = pADC0->ADC_CDR[0];

        //Compute the value in degrees
        tempDeg[i]=((2.616-(2.5/1024*tempValue[i]))/10.9e-3)-50;
    }
}

void IRQHandler (void)
{
    //clear IRQ
    AT91C_BASE_AIC->AIC_EOICR = (1 << AT91C_ID_IRQ0) ;
    AT91C_BASE_AIC->AIC_ICCR = (1 << AT91C_ID_IRQ0) ;

    //Send a dummy byte on the SPI interface
    AT91C_BASE_SPI->SPI_TDR=0x77;

    //Check the SPI status register to see when the Data has been received
    while((AT91C_BASE_SPI->SPI_SR & 0x1) != 0x1){};

    //Store the incoming Data of the SPI RX Buffer
    receiveData[0]=(AT91C_BASE_SPI->SPI_RDR);

    //Send a  second dummy byte on the SPI interface
    AT91C_BASE_SPI->SPI_TDR=0x00;

    //Check the SPI status register to see when the Data has been received
    while((AT91C_BASE_SPI->SPI_SR & 0x1) != 0x1){};

    //Store the incoming Data of the SPI RX Buffer
    receiveData[1]=(AT91C_BASE_SPI->SPI_RDR);
}
```

```
void initIRQ ()
{
    //PIO enable peripheral
    AT91C_BASE_PIOA->PIO_PDR |= (0x1<<9);

    //Level sensitivity and priority of interrupt
    AT91C_BASE_AIC->AIC_SMR[AT91C_ID_IRQ0] = (0x1<<5) | 5 ;

    //give fonction to call when interrupt comes
    AT91C_BASE_AIC->AIC_SVR[AT91C_ID_IRQ0] = (unsigned int)IRQHandler ;

    //enable IRQ
    AT91C_BASE_AIC->AIC_IECR = (1 << AT91C_ID_IRQ0) ;
}


int main ()
{
    //*****************************************************************
    //Call functions for different tests
    //*****************************************************************

    lowLevelInit();
    initIRQ ();
    initSPI();
    entry_flash_cfi_qry();
    initADC0();

    int i=0;
    while (1)
    {
        i++ ;

        if (i > 4)
            i = 0 ;

        asm("MOV r0, #5") ;
        asm("MOV r1, #6") ;
        asm("MOV r2, #7") ;
    }
    return 0 ;
}
```

APPENDIX I SOURCE FILES FOR THE MSP

spi.c

```c
#include "msp430x16x.h"

//*-------------------------------------------------------------------*/
//* a little prog to test spi transaction with the ARM                */
//*---------------------------------------------------**********------*/

__interrupt void SPI_TX_ISR(void);
__interrupt void SPI_RX_ISR(void);

// buffer to store incomming byte
char bufRX[2]={0x00,0x00};
int i;

void main(void)
{
    WDTCTL = WDTPW + WDTHOLD;     //Stop watchdog timer

    P1DIR |= 0x20;                //Set P1.5 to output direction
    P1OUT  = 0x20;                //Set the output high

    UTCTL1 = STC;                 //External UCLK, 3-pin mode
    UCTL1 = CHAR+SYNC ;           //8-bit SPI Slave **SWRST**
    UBR01 = 0x50;                 //UCLK/80
    UBR11 = 0x00;                 //0
    UMCTL1 = 0x00;                //no modulation
    P5SEL |= 0x0E;                //P5.1 to P5.3 SPI option select
    P5DIR |= 0x01;                //P3.0 output direction
    ME2 |= USPIE1;                //Enable USART1 SPI mode

    IE2 |= 0x30;                  //enable interupt spi tx and rx
    IFG2 &= ~(BIT4+BIT5);         //desactivate startup tx/rx interupt

    TXBUF1 = 0x11;                //Put a value in the SPI TX Buffer

    _EINT();                      //Enable interrupts

    //Set the output LOW then HIGH to generate an interupt on the ARM
    P1OUT  = 0x00;                //Set the output low
    P1OUT  = 0x20;                //Set the output high

    i=0;

   while ( 1 )
   {

   }
}
//Interrupt routine to put the new value on the SPI TX Buffer
//after the Master has initiate the transfer
USART1TX_ISR(SPI_TX_ISR)
__interrupt void SPI_TX_ISR(void)
{
    //Put the data in the SPI TX buffer
    TXBUF1 = 0xAA;
}

//Interrupt routine to store the recevied data from the master
USART1RX_ISR(SPI_RX_ISR)
__interrupt void SPI_RX_ISR(void)
{
    //Store the received value in a local table
    bufRX[i]=RXBUF1;
    i++;
}
```

i2c.c

```c
#include  <msp430x16x.h>

void main (void)
{
  WDTCTL = WDTPW + WDTHOLD;                 // Stop Watchdog Timer
  P3SEL |= 0x0A;                            // Select I2C pins (P3.1 and P3.3)
  U0CTL |= I2C + SYNC;                      // Select I2C mode
  U0CTL &= ~I2CEN;                          // Disable the I2C module
  I2CTCTL |= I2CSSEL1;                      // Select the I2C clock
  I2CNDAT = 0x01;                           // Transfer one byte
  I2CSA = 0x48;                             // Slave Address is 048h
  U0CTL |= I2CEN;                           // Enable I2C
  U0CTL |= MST;                             // Master mode


  I2CTCTL |= I2CSTT + I2CSTP;               // Initiate transfer
                                            // Start Bit + Slave Adress + Stop bit

  while(1);

}
```

APPENDIX J PSA ANALYSIS REPORT

Circuit Name	Item	Value	Parameter	Rated Value	Derating	Allowed Value	Value	Stress	Status
CDMS qualif. Board	**R1**	RS0603 100k 75V 125mW 1%	V_max	75V	80%	60V	3.6 V	6.000%	ok
			Pw_max	125 mW	75% at 70°C	93.75 mW	0.00013 mW	0.0%	ok
			T_max	125°C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**R2**	RS0603 100k 75V 125mW 1%	V_max	75V	80%	60V	3.6 V	6.0%	ok
			Pw_max	125 mW	75% at 70°C	93.75 mW	0.00013 mW	0.0%	ok
			T_max	125°C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**R3**	RS0603 100k 75V 125mW 1%	V_max	75V	80%	60V	3.6 V	6.0%	ok
			Pw_max	125 mW	75% at 70°C	93.75 mW	0.00013 mW	0.0%	ok
			T_max	125°C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**R4**	RS0603 100k 75V 125mW 1%	V_max	75V	80%	60V	3.6 V	6.0%	ok
			Pw_max	125 mW	75% at 70°C	93.75 mW	0.00013 mW	0.0%	ok
			T_max	125°C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**R5**	RS0603 100k 75V 125mW 1%	V_max	75V	80%	60V	3.6 V	6.0%	ok
			Pw_max	125 mW	75% at 70°C	93.75 mW	0.00013 mW	0.0%	ok
			T_max	125°C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**R6**	RS0603 100k 75V 125mW 1%	V_max	75V	80%	60V	3.6 V	6.0%	ok
			Pw_max	125 mW	75% at 70°C	93.75 mW	0.00013 mW	0.0%	ok
			T_max	125°C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**R7**	RS0603 100k 75V 125mW 1%	V_max	75V	80%	60V	3.6 V	6.0%	ok
			Pw_max	125 mW	75% at 70°C	93.75 mW	0.00013 mW	0.0%	ok
			T_max	125°C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**R8**	RS0603 100k 75V 125mW 1%	V_max	75V	80%	60V	3.6 V	6.0%	ok
			Pw_max	125 mW	75% at 70°C	93.75 mW	0.00013 mW	0.0%	ok
			T_max	125°C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**R9**	RS0603 100k 75V 125mW 1%	V_max	75V	80%	60V	3.6 V	6.0%	ok
			Pw_max	125 mW	75% at 70°C	93.75 mW	0.00013 mW	0.0%	ok
			T_max	125°C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**R10**	RS0603 100k 75V 125mW 1%	V_max	75V	80%	60V	3.6 V	6.0%	ok
			Pw_max	125 mW	75% at 70°C	93.75 mW	0.00013 mW	0.0%	ok
			T_max	125°C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**R11**	RS0603 100k 75V 125mW 1%	V_max	75V	80%	60V	3.6 V	6.0%	ok
			Pw_max	125 mW	75% at 70°C	93.75 mW	0.00013 mW	0.0%	ok
			T_max	125°C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**R12**	RS0603 100k 75V 125mW 1%	V_max	75V	80%	60V	3.6 V	6.0%	ok
			Pw_max	125 mW	75% at 70°C	93.75 mW	0.00013 mW	0.0%	ok
			T_max	125°C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**R13**	RS0603 100k 75V 125mW 1%	V_max	75V	80%	60V	3.6 V	6.0%	ok
			Pw_max	125 mW	75% at 70°C	93.75 mW	0.00013 mW	0.0%	ok
			T_max	125°C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**R14**	RS0603 100k 75V 125mW 1%	V_max	75V	80%	60V	3.6 V	6.0%	ok
			Pw_max	125 mW	75% at 70°C	93.75 mW	0.00013 mW	0.0%	ok
			T_max	125°C	100%	125 °C	60 °C	48.0%	ok

Circuit Name	Item	Value	Parameter	Rated Value	Derating	Allowed Value	Value	Stress	Status
CDMS qualif. Board	**R15**	RS0603 100k 75V 125mW 1%	V_max	75V	80%	60V	3.6 V	6.0%	ok
			Pw_max	125 mW	75% at 70°C	93.75 mW	0.00013 mW	0.0%	ok
			T_max	125°C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**R16**	RS0603 100k 75V 125mW 1%	V_max	75V	80%	60V	3.6 V	6.0%	ok
			Pw_max	125 mW	75% at 70°C	93.75 mW	0.00013 mW	0.0%	ok
			T_max	125°C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**R17**	RS0603 100k 75V 125mW 1%	V_max	75V	80%	60V	3.6 V	6.0%	ok
			Pw_max	125 mW	75% at 70°C	93.75 mW	0.00013 mW	0.0%	ok
			T_max	125°C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**R18**	RS0603 100k 75V 125mW 1%	V_max	75V	80%	60V	3.6 V	6.0%	ok
			Pw_max	125 mW	75% at 70°C	93.75 mW	0.00013 mW	0.0%	ok
			T_max	125°C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**R19**	RS0603 100k 75V 125mW 1%	V_max	75V	80%	60V	3.6 V	6.0%	ok
			Pw_max	125 mW	75% at 70°C	93.75 mW	0.00013 mW	0.0%	ok
			T_max	125°C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**R20**	RS0603 100k 75V 125mW 1%	V_max	75V	80%	60V	3.6 V	6.0%	ok
			Pw_max	125 mW	75% at 70°C	93.75 mW	0.00013 mW	0.0%	ok
			T_max	125°C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**R21**	RS0603 400k 75V 125mW 1%	V_max	75V	80%	60V	3.6 V	6.0%	ok
			Pw_max	125 mW	75% at 70°C	93.75 mW	3.2E-05 mW	0.0%	ok
			T_max	125°C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**R22**	RS0603 100k 75V 125mW 1%	V_max	75V	80%	60V	3.6 V	6.0%	ok
			Pw_max	125 mW	75% at 70°C	93.75 mW	0.00013 mW	0.0%	ok
			T_max	125°C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**R23**	RS0603 20R 75V 125mW 1%	V_max	75V	80%	60V	1.955 V	3.3%	ok
			Pw_max	125 mW	75% at 70°C	93.75 mW	0.648 mW	0.7%	ok
			T_max	125°C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**R24**	RS0603 100k 75V 125mW 1%	V_max	75V	80%	60V	3.6 V	6.0%	ok
			Pw_max	125 mW	75% at 70°C	93.75 mW	0.00013 mW	0.0%	ok
			T_max	125°C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**R25**	RS0603 100k 75V 125mW 1%	V_max	75V	80%	60V	3.6 V	6.0%	ok
			Pw_max	125 mW	75% at 70°C	93.75 mW	0.00013 mW	0.0%	ok
			T_max	125°C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**R26**	RS0603 100k 75V 125mW 1%	V_max	75V	80%	60V	3.6 V	6.0%	ok
			Pw_max	125 mW	75% at 70°C	93.75 mW	0.00013 mW	0.0%	ok
			T_max	125°C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**R27**	RS0603 10k 75V 125mW 1%	V_max	75V	80%	60V	3.6 V	6.0%	ok
			Pw_max	125 mW	75% at 70°C	93.75 mW	0.0013 mW	0.0%	ok
			T_max	125°C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**R28**	RS0603 100k 75V 125mW 1%	V_max	75V	80%	60V	3.6 V	6.0%	ok
			Pw_max	125 mW	75% at 70°C	93.75 mW	0.00013 mW	0.0%	ok
			T_max	125°C	100%	125 °C	60 °C	48.0%	ok

Circuit Name	Item	Value	Parameter	Rated Value	Derating	Allowed Value	Value	Stress	Status
CDMS qualif. Board	**R29**	RS0603 100k 75V 125mW 1%	V_max	75V	80%	60V	3.6 V	6.0%	ok
			Pw_max	125 mW	75% at 70°C	93.75 mW	0.00013 mW	0.0%	ok
			T_max	125°C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**R30**	RS0603 100k 75V 125mW 1%	V_max	75V	80%	60V	3.6 V	6.0%	ok
			Pw_max	125 mW	75% at 70°C	93.75 mW	0.00013 mW	0.0%	ok
			T_max	125°C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**R31**	RS0603 100k 75V 125mW 1%	V_max	75V	80%	60V	3.6 V	6.0%	ok
			Pw_max	125 mW	75% at 70°C	93.75 mW	0.00013 mW	0.0%	ok
			T_max	125°C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**R32**	RS0603 100k 75V 125mW 1%	V_max	75V	80%	60V	3.6 V	6.0%	ok
			Pw_max	125 mW	75% at 70°C	93.75 mW	0.00013 mW	0.0%	ok
			T_max	125°C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**R33**	RS0603 1k 75V 125mW 1%	V_max	75V	80%	60V	3.6 V	6.0%	ok
			Pw_max	125 mW	75% at 70°C	93.75 mW	0.01296 mW	0.0%	ok
			T_max	125°C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**R34**	RS0603 287R 50V 63mW 1%	V_max	50V	80%	40V	3.6 V	9.0%	ok
			Pw_max	63 mW	75% at 70°C	47.25 mW	0.00013 mW	0.0%	ok
			T_max	125°C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**R35**	RS0603 100k 75V 125mW 1%	V_max	75V	80%	60V	3.6 V	6.000%	ok
			Pw_max	125 mW	75% at 70°C	93.75 mW	0.00013 mW	0.0%	ok
			T_max	125°C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**R36**	RS0603 100k 75V 125mW 1%	V_max	75V	80%	60V	3.6 V	6.000%	ok
			Pw_max	125 mW	75% at 70°C	93.75 mW	0.00013 mW	0.0%	ok
			T_max	125°C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**R37**	RS0603 10k 75V 125mW 1%	V_max	75V	80%	60V	2.9 V	4.833%	ok
			Pw_max	125 mW	75% at 70°C	93.75 mW	0.0013 mW	0.0%	ok
			T_max	125°C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**R38**	RS0603 100k 75V 125mW 1%	V_max	75V	80%	60V	2.9 V	4.833%	ok
			Pw_max	125 mW	75% at 70°C	93.75 mW	0.00013 mW	0.0%	ok
			T_max	125°C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**R39**	RS0603 100k 75V 125mW 1%	V_max	75V	80%	60V	2.9 V	4.833%	ok
			Pw_max	125 mW	75% at 70°C	93.75 mW	0.00013 mW	0.0%	ok
			T_max	125°C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**R40**	RS0603 200k 75V 125mW 1%	V_max	75V	80%	60V	2.9 V	4.833%	ok
			Pw_max	125 mW	75% at 70°C	93.75 mW	6.5E-05 mW	0.0%	ok
			T_max	125°C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**R44**	RS0603 100k 75V 125mW 1%	V_max	75V	80%	60V	2.9 V	4.833%	ok
			Pw_max	125 mW	75% at 70°C	93.75 mW	0.00013 mW	0.0%	ok
			T_max	125°C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**R45**	RS0603 100k 75V 125mW 1%	V_max	75V	80%	60V	2.9 V	4.833%	ok
			Pw_max	125 mW	75% at 70°C	93.75 mW	0.00013 mW	0.0%	ok
			T_max	125°C	100%	125 °C	60 °C	48.0%	ok

Circuit Name	Item	Value	Parameter	Rated Value	Derating	Allowed Value	Value	Stress	Status
CDMS qualif. Board	**C1**	CAPS0603 Ceramic 100nF 25V 10%	V_max	25 V	80%	20 V	3.6 V	18.0%	ok
			T_max	125 °C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**C2**	CAPS0603 Ceramic 100nF 25V 10%	V_max	25 V	80%	20 V	2.5 V	12.5%	ok
			T_max	125 °C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**C3**	CAPS0603 Ceramic 100nF 25V 10%	V_max	25 V	80%	20 V	3.6 V	18.0%	ok
			T_max	125 °C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**C4**	CAPS0603 Ceramic 100nF 25V 10%	V_max	25 V	80%	20 V	3.6 V	18.0%	ok
			T_max	125 °C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**C5**	CAPS0603 Ceramic 100nF 25V 10%	V_max	25 V	80%	20 V	3.6 V	18.0%	ok
			T_max	125 °C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**C6**	CAPS0603 Ceramic 100nF 25V 10%	V_max	25 V	80%	20 V	3.6 V	18.0%	ok
			T_max	125 °C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**C7**	CAPS0603 Ceramic 100nF 25V 10%	V_max	25 V	80%	20 V	3.6 V	18.0%	ok
			T_max	125 °C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**C8**	CAPS0603 Ceramic 100nF 25V 10%	V_max	25 V	80%	20 V	3.6 V	18.0%	ok
			T_max	125 °C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**C9**	CAPS0603 Ceramic 100nF 25V 10%	V_max	25 V	80%	20 V	3.6 V	18.0%	ok
			T_max	125 °C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**C10**	CAPS0603 Ceramic 100nF 25V 10%	V_max	25 V	80%	20 V	3.6 V	18.0%	ok
			T_max	125 °C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**C11**	CAPS0603 Ceramic 100nF 25V 10%	V_max	25 V	80%	20 V	3.6 V	18.0%	ok
			T_max	125 °C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**C12**	CAPS0603 Ceramic 100nF 25V 10%	V_max	25 V	80%	20 V	3.6 V	18.0%	ok
			T_max	125 °C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**C13**	CAPS0603 Ceramic 100nF 25V 10%	V_max	25 V	80%	20 V	3.6 V	18.0%	ok
			T_max	125 °C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**C14**	CAPS0603 Ceramic 100nF 25V 10%	V_max	25 V	80%	20 V	3.6 V	18.0%	ok
			T_max	125 °C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**C15**	CAPS0603 Ceramic 100nF 25V 10%	V_max	25 V	80%	20 V	3.6 V	18.0%	ok
			T_max	125 °C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**C16**	CAPS0603 Ceramic 100nF 25V 10%	V_max	25 V	80%	20 V	3.6 V	18.0%	ok
			T_max	125 °C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**C17**	CAPS0603 Ceramic 100nF 25V 10%	V_max	25 V	80%	20 V	3.6 V	18.0%	ok
			T_max	125 °C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**C18**	CAPS0603 Ceramic 100nF 25V 10%	V_max	25 V	80%	20 V	3.6 V	18.0%	ok
			T_max	125 °C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**C20**	CAPS0603 Ceramic 100nF 25V 10%	V_max	25 V	80%	20 V	3.6 V	18.0%	ok
			T_max	125 °C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**C22**	CAPS0603 Ceramic 100nF 25V 10%	V_max	25 V	80%	20 V	3.6 V	18.0%	ok
			T_max	125 °C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**C23**	CAPS0603 Solid Tantalum TAJ-C 10u 16V 10%	V_max	16 V	60%	9.6 V	3.6 V	37.5%	ok
			T_max	85 °C	100%	85 °C	60 °C	70.6%	ok

Circuit Name	Item	Value	Parameter	Rated Value	Derating	Allowed Value	Value	Stress	Status
CDMS qualif. Board	**C24**	CAPS0603 Ceramic 100nF 25V 10%	V_max	25 V	80%	20 V	3.6 V	18.0%	ok
			T_max	125 °C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**C25**	CAPS0603 Solid Tantalum TAJ-B 1u 35V 10%	V_max	35 V	60%	21 V	2.5 V	11.9%	ok
			T_max	85 °C	100%	85 °C	60 °C	70.6%	ok
CDMS qualif. Board	**C26**	CAPS0603 Ceramic 100nF 25V 10%	V_max	25 V	80%	20 V	2.5 V	12.5%	ok
			T_max	125 °C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**C27**	CAPS0603 Ceramic 100nF 25V 10%	V_max	25 V	80%	20 V	2.5 V	12.5%	ok
			T_max	125 °C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**C28**	CAPS0603 Ceramic 150nF 10V 10%	V_max	10 V	80%	8 V	1.955 V	24.4%	ok
			T_max	125 °C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**C29**	CAPS0603 Ceramic 100nF 25V 10%	V_max	25 V	80%	20 V	2.5 V	12.5%	ok
			T_max	125 °C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**C30**	CAPS0603 Ceramic 100nF 25V 10%	V_max	25 V	80%	20 V	3.6 V	18.0%	ok
			T_max	125 °C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**C31**	CAPS0603 Ceramic 100nF 25V 10%	V_max	25 V	80%	20 V	3.6 V	18.0%	ok
			T_max	125 °C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**C32**	CAPS0603 Ceramic 100nF 25V 10%	V_max	25 V	80%	20 V	3.6 V	18.0%	ok
			T_max	125 °C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**C33**	CAPS0603 Ceramic 100nF 25V 10%	V_max	25 V	80%	20 V	3.6 V	18.0%	ok
			T_max	125 °C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**C34**	CAPS0603 Ceramic 100nF 25V 10%	V_max	25 V	80%	20 V	3.6 V	18.0%	ok
			T_max	125 °C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**C35**	CAPS0603 Ceramic 100nF 25V 10%	V_max	25 V	80%	20 V	3.6 V	18.0%	ok
			T_max	125 °C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**C36**	CAPS0603 Ceramic 100nF 25V 10%	V_max	25 V	80%	20 V	3.6 V	18.0%	ok
			T_max	125 °C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**C37**	CAPS0603 Ceramic 100nF 25V 10%	V_max	25 V	80%	20 V	3.6 V	18.0%	ok
			T_max	125 °C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**C42**	CAPS0603 Ceramic 2.7nF 50V 10%	V_max	50 V	80%	40 V	3.6 V	9.0%	ok
			T_max	125 °C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**C43**	CAPS0603 Ceramic 270pF 50V 5%	V_max	50 V	80%	40 V	3.6 V	9.0%	ok
			T_max	125 °C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**C44**	CAPS0805 Ceramic 10uF 6.3V 10%	V_max	6.3 V	80%	5.04 V	3.6 V	71.4%	ok
			T_max	85 °C	100%	85 °C	60 °C	70.6%	ok
CDMS qualif. Board	**C45**	CAPS0603 Ceramic 100nF 25V 10%	V_max	25 V	80%	20 V	3.6 V	18.0%	ok
			T_max	125 °C	100%	125 °C	60 °C	48.0%	ok
CDMS qualif. Board	**C46**	CAPS0805 Ceramic 10uF 6.3V 10%	V_max	6.3 V	80%	5.04 V	3.6 V	71.4%	ok
			T_max	85 °C	100%	85 °C	60 °C	70.6%	ok
CDMS qualif. Board	**C47**	CAPS0805 Ceramic 10uF 6.3V 10%	V_max	6.3 V	80%	5.04 V	3.6 V	71.4%	ok
			T_max	85 °C	100%	85 °C	60 °C	70.6%	ok

Item	Value	Parameter	Rated Value	Derating	Allowed Value	Value	Stress	Status
D1	DIODE BAR43FILM (schottky)	$I_{FSM\ non\ répétitive}$	0.75A	75%	0.56A	-	-	-
		$I_{FSM\ répétitive}$	0.1A	50%	0.05A	0.0001125A	0.2%	ok
		V_R	30 V	80%	24 V	3.6 V	15.0%	ok
		P_D	250 mW	80%	200 mW	7.8E-05 mW	0.0%	ok
		Tj	150 °C	Tjmax-40°C	110°C	60 °C	54.5%	ok
Q2	MOSFET	Vds	-12 V	75%	-9 V	-3.6 V	40.0%	ok
		Vgs	+/-8V	75%	+/-6.V	+/-3.6V	60.0%	ok
		PD	0.8W	80%	0.64W	0.25 mW	39.1%	ok
		Tj	150 °C	Tjmax-40°C	110 °C	60 °C	54.5%	ok
Q1	NPN TRANSISTOR BC817	V_{CE}	45 V	80%	36 V	3.6 V	10.0%	ok
		V_{CB}	50V	80%	40 V	2.9 V	7.3%	ok
		V_{EB}	50V	80%	40 V	0.7 V	1.8%	ok
		$I_c$max	500 mA	80%	400 mA	17.4 mA	4.4%	ok
		$I_b$max	300 mA	80%	240 mA	0.29 mA	0.1%	ok
		Tj	125 °C	100%	125 °C	60 °C	48.0%	ok
U1	AT91M55800A Microcontroller	Vddcore	3.6 V	90%	3.24 V	3.6 V	111.1%	Nok
		Vdda	3.6 V	90%	3.24 V	3.6 V	111.1%	Nok
		Vddpll	3.6 V	90%	3.24 V	3.6 V	111.1%	Nok
		Vddbu	3.9 V	90%	3.51 V	3.6 V	102.6%	Nok
		Vddio	5.5 V	90%	4.95 V	3.6 V	72.7%	ok
		Tj	85°C	Tjmax-40°C	45 °C	60 °C	133.3%	Nok
U2	Microcontroller MSP430F1611IPM	Vcc	4.1 V	90%	3.69 V	3.6 V	97.6%	ok
		Tj	85°C	Tjmax-40°C	45 °C	60 °C	133.3%	Nok
U3	REF192GSZ Voltage ref +2.5V	Vcc	18V	90%	16.2 V	3.6 V	22.2%	ok
		Iout	30 mA	80%	24 mA	0.000637 mA	0.0%	ok
		Tj	125°C	Tjmax-40°C	85 °C	60 °C	70.6%	ok
U4	LM94022BIMG Temp. Sensor	Vcc	6V	90%	5.4 V	2.5 V	46.3%	ok
		Iout	7 mA	80%	5.6 mA	0.195 mA	3.5%	ok
		Tj	150°C	Tjmax-40°C	110 °C	60 °C	54.5%	ok
U5	OPA2333AID Ampli	Vcc	7V	90%	6.3 V	2.5 V	39.7%	ok
		Vin	7.3 V	70%	5.11 V	1.955 V	38.3%	ok
		Iout	10 mA	80%	8 mA	0.195 mA	2.4%	ok
		Tj	150°C	Tjmax-40°C	110 °C	60 °C	54.5%	ok
U6	R1LV0416CSB-7LI SRAM 256k x 16 bit	Vcc	3.6 V	90%	3.24 V	3.6 V	111.1%	Nok
		Tj	85 °C	Tmax-40°C	45 °C	60 °C	133.3%	Nok
U7	S29AL016D90TFI02 FLASH 1M x 16 bit	Vcc	4 V	90%	3.6 V	3.6 V (max)	100.0%	ok
		Iout	200 mA	80%	160 mA			
		Tmax	85 °C	Tmax-40°C	45 °C	60 °C	133.3%	Nok
U8	AT27BV4096-12VI	Vcc	7 V	90%	6.3 V	3.3 V	52.4%	ok
		Tmax	125 °C	Tmax-40°C	85 °C	60 °C	70.6%	ok
U9	S29AL016D90TFI02 FLASH 1M x 16 bit	Vcc	4 V	90%	3.6 V	3.6 V (max)	100.0%	ok
		Iout	200 mA	80%	160 mA			
		Tmax	85 °C	Tmax-40°C	45 °C	60 °C	133.3%	Nok

Item	Value	Parameter	Rated Value	Derating	Allowed Value	Value	Stress	Status
U10	S29AL016D90TFI02 FLASH 1M x 16 bit	Vcc	4 V	90%	3.6 V	3.6 V (max)	100.0%	ok
		Tmax	85 ℃	Tmax-40℃	45 ℃	60 ℃	133.3%	Nok
U11	MAX823 supervisor	Vcc	6V	100%	6V	3.6 V	60.0%	ok
		Tj	125℃	Tjmax-40℃	85℃	70 V	82.4%	ok
U12	SN74LV74D FLIP FLOP D	Vcc	7 V	90%	6.3 V	3.6 V	57.1%	ok
		Tj	125 ℃	Tjmax-40℃	85 ℃	60 ℃	70.6%	ok
U13	74LVX08M LOGIC AND GATE	Vcc	7 V	90%	6.3 V	3.6 V	57.1%	ok
		Tj	85 ℃	Tjmax-40℃	45 ℃	60 ℃	133.3%	Nok
SW	SW-NHDS-08T	ONLY	FOR	TEST	-	-	-	-
S1	PUSHBUTT-KSC341	ONLY	FOR	TEST	-	-	-	-
S2	PUSHBUTT-KSC341	ONLY	FOR	TEST	-	-	-	-
X1	Quartz MC-146 32.768kHz	DL	1 uW	100%	1 uW	0.693 uW	69.3%	ok
		Tmax	85℃	100%	85 ℃	60 ℃	70.6%	ok
X2	Quartz SMU-4	DL	500 uW	100%	500 uW	126.4 uW	25.3%	ok
		Tmax	70℃	100%	70 ℃	60 ℃	85.7%	ok
X3	Quartz MC-146 32.768kHz	DL	1 uW	100%	1 uW	0.693 uW	69.3%	ok
		Tmax	85℃	100%	85 ℃	60 ℃	70.6%	ok