UNIVERSITÀ DEGLI STUDI DI PAVIA
Facoltà di Ingegneria
Dipartimento di Ingegneria Industriale e dell'Informazione
Corso di Laurea in Ingegneria Elettronica e delle Telecomunicazioni

A readout system for ionizing and non-ionizing radiation dosimeters

Sistema di misura per dosimetri di radiazione ionizzante e non-ionizzante

Supervisor: Professor
Lodovico Ratti

Co - supervisors:
Maurice Glaser
Federico Ravotti

Author:
Georgi Gorine
UIN **380734**

Academic year 2011/2012

Contents

Introduction

In High Energy Physics experiments high levels of radiation may be generated, which can damage the electronics and could be really harmful for workers. In such environments the deployment of an adequate monitoring system is therefore mandatory. At CERN (*European Organization for Nuclear Research*), in the framework of the radiation protection program, the RADMON (RADiation MONitoring) project provides a passive dosimeter solution for ionizing and non-ionizing radiation level detection. The readout of RADMON dosimeters is currently performed using complex existing electronics or expensive workbench instrumentation. This thesis will discuss in detail the hardware and software development of a new portable, smaller and cheaper, RADMON readout system that will be produced in many copies and used directly by the end users.

The first chapter will describe the RADMON devices, by showing the differences between ionizing and non-ionizing dosimeters and by explaining how the readout is performed. The second chapter will initially show the present RADMON readout system, focusing on its main features, and will then describe in detail the design of each hardware part of the new readout system. The third and final chapter is devoted to describing the interface developed in LabVIEW for the present system and the terminal based software for the new readout system. At the end of this work, Appendix A shows the block diagrams of the individual circuits described in the second chapter while Appendix B shows the entire source code of the terminal based software described in the third chapter.

Chapter 1

Radiation Monitors at CERN

This chapter will introduce the role within the CERN of the *Detector Technology* group and in particular will present the aim of the RADMON project. The last section will describe the two types of dosimeters used in the project by showing their mode of operation and the characteristics of the different technologies.

1.1 Radiation monitoring in the LHC tunnel

The CERN is a world leading physics laboratory. It was established in 1954, in Geneva (Switzerland), with the goal of unifying all the European scientist. The aim of CERN is to build particle accelerators and experiments for high energy physics that are needed for the study of new sub-nuclear particles and the verification of models that explain their interactions. The last experiment built at CERN for this purpose is the LHC *(Large Hadron Collider)*. This circular accelerator is the most powerful in the world, being able to produce proton-proton collisions at a center-of-mass energy of 14 TeV. In such an environment, high levels of radiation are generated, making it necessary to have an adequate monitoring systems capable of

- protecting the personnel from stray radiation and material activation,
- assessing the radiation damage on detectors and electronics components in the experiments and in the LHC.

The radiation environment met in the LHC accelerator is completely different from standard applications in which existing dosimetric technologies are used like for example in space, medical and nuclear applications. In this environment, many secondary particles at lower energies like protons, neutrons, pions are produced from the interaction of the high energy beams. If the energy of these secondary particles is sufficiently high, they may cause interactions with matter and produce more particles in the form of showers. The *Detector Technology* group at the *Physics Department (PH-DT)* supports the development, construction, operation and maintenance of particle detectors for the experiments at CERN. In particular, the *Irradiation Facility* [1] laboratory developed several types of radiation sensors, named RADMON, which aim is to monitor those secondary lower energy particles to predict possible malfunctions of the electronics in the experiments. A picture of the LHC and its main experiments is shown in Figure 1.1 .

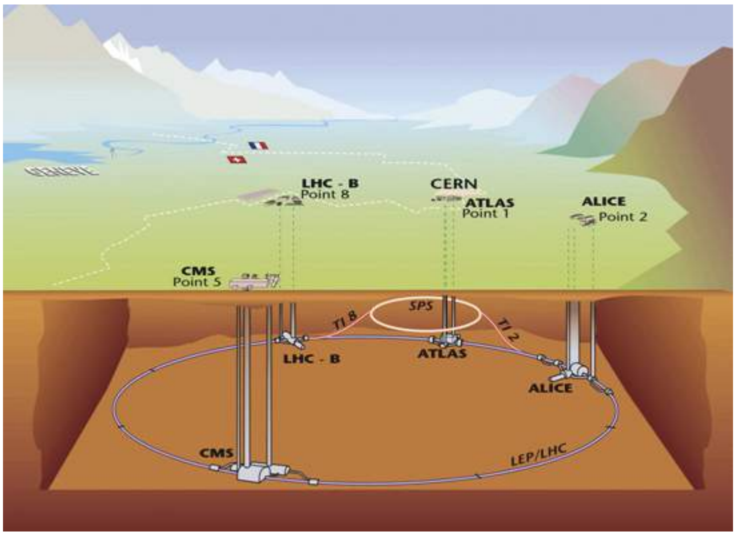


Figure 1.1: overall view of LHC and its experiments.

1.2 The RADMON monitoring system

Starting from 2002, the RADMON working group [2] developed and characterized a set of sensors to be used for radiation monitoring in the high energy physics experiments of the LHC [3]. The purpose of the RADMON group activity, is to develop different techonlogies of radiation sensors able to assess the radiation damage on electronics, in particular by

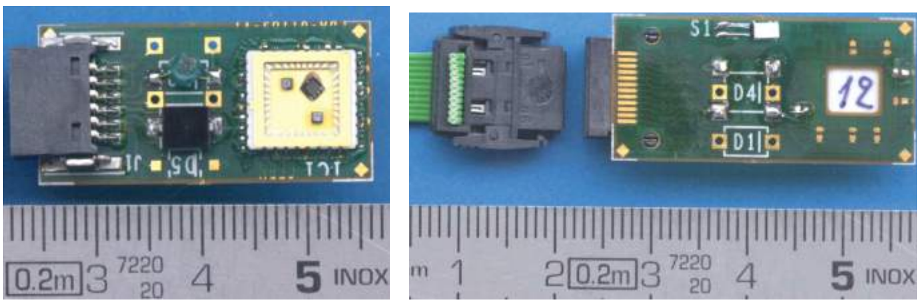
- providing a mapping of radiation dose and fluence around the electronic equipment to improve radiation shielding where and when needed,
- allowing for an early warning when levels of radiation are too high or are increasing faster than expected [4].

1.3 RADMON Radiation Monitor description

In a RADMON dosimeter, two types of sensors are used: RadFETs, which are used for monitoring the radiation dose (measured in Gy) by indirectly measuring the charge build up in the gate oxide of a MOS structure, and $p^+/n/n^+$ diodes, where, by measuring the bulk damage inside the *n-Si* base, it is possible to evaluate the particle fluence (generally expressed in particle/cm^2). These dosimeters have the great advantage of providing a real-time information without the need to remove the sensors from the operating field. The RADMON sensor PCB (Printed Circuit Board) can mount

- two types RadFET
  - LAAS , *Laboratory of Analysis and Architecture of Systems*,
  - REM, *Radiation Experiments and Monitors* ,
- three types p-i-n diodes
  - BPW34, produced by Siemens,
  - CMRP, *Center for Medical Radiation Physics*,
  - LBSD, *Low-Barrier Schottky Diode.*

Since the idea is to use the same readout scheme for both RadFETs and p-i-n diodes, they have been mounted together on a single "Integrated Sensor Carrier", shown in Figure 1.2. On a single RADMON sensor carrier there is place for a max of 11 dosimeters.



(a) (b)

Figure 1.2: PCB front-side with 2 p-i-n diodes and 3 RadFET dies on board (a) and PCB back-side with additional place for more dosimeters and external cabling (b).

1.3.1 Ionizing radiation dosimetry with the RadFET transistor

The RadFET *(Radiation-sensitive Field-Effect Transistor)* device was invented by Poch and Holmes-Siedle in 1969. The operation of RadFET dosimeter is based on the principle of charge trapping that occurs in the gate oxide layer of the MOS transistors exposed to ionizing radiation. RadFET transistors are large p-channel devices with a gate width varying from 300 μm to 700 μm, a length varying from 50 μm to 150 μm and a gate oxide thickness varying from 250 nm to 1600 nm [5]. The choice of the transistor size is crucial in determining the overall sensitivity of the system since the larger is the oxide thickness, the greater will be the amount of charge captured in the gate oxide.

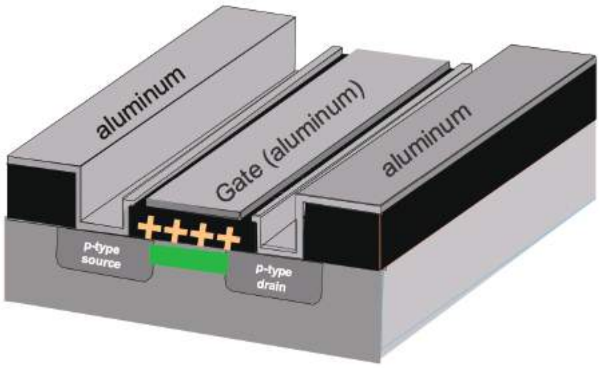


Figure 1.3: structure of a typical *RadFET*. The oxide layer under the gate electrode is the main sensor volume and the sheet of trapped positive charge shown in that layer generates the main dosimetric signal in the p-type silicon channel[5].

The mechanism of charge trapping is shown in Figure 1.4. The number of electron-hole pairs in the oxide of the RadFET determines the threshold voltage. The threshold voltage V_T of a MOS transistor is the gate voltage at which the holes (or electrons) forms a conducting path along the gate oxide and between the *source* and *drain* diffusions. When the RADMON p-channel REM or LAAS sensor is irradiated,while electrons are easily swept away by a small electric field applied across the gate oxide, holes, whose mobility in SiO_2 is very small, remain trapped in it. Therefore the threshold voltage is modified by the accumulated charge. According to a well known model [6] its

Table 1.1: parameters for RADOMON's RadFET

Type	LAAS	REM
a	0.01854 V/Gy	0.02921 V/Gy
b	0.91072	0.78778
D	$10^{-1}Gy < D < 40Gy$	$40Gy < D < 10^3Gy$

variation $\Delta\ V_t$ can be expressed as:

$$\Delta V_t = a \times D^b \tag{1.1}$$

Where a and b are two fitting parameters which depend on the oxide thickness and the technology and D is the dose. The particular parameter values for CERN RadFET are shown in Table 1.1.

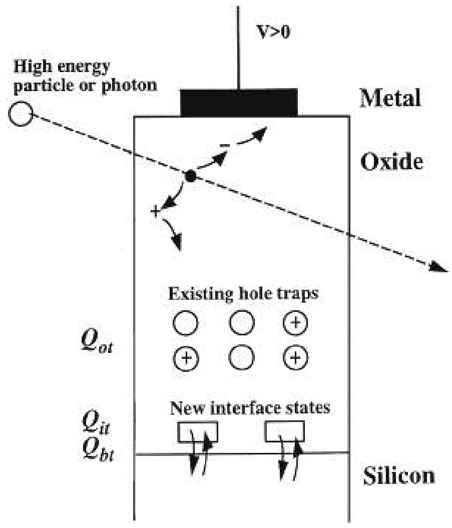


Figure 1.4: the principle of hole trapping in the gate oxide film.[7].

Figure 1.5 shows different I_D vs V_G characteristics of a p-MOS RadFET obtained for different values of the threshold voltage V_T as modified by exposure to increasing levels of ionising radiation. In this way, by setting a constant current I_D, a different V_G voltage can be read depending on the radiation dose.

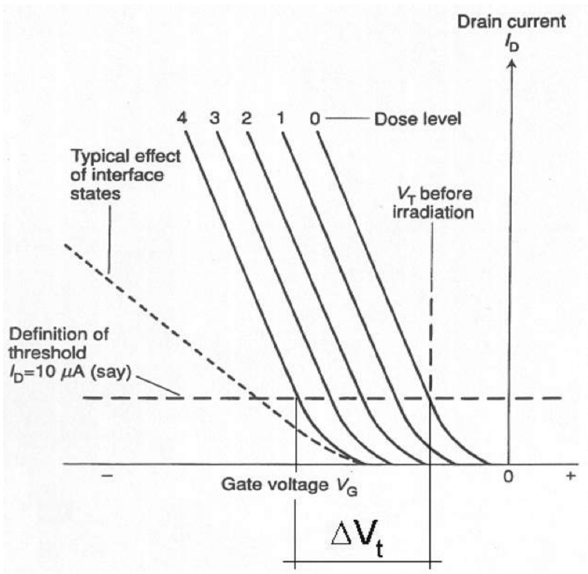


Figure 1.5: the shift in the characteristic of a p-channel RadFET as the dose increases (from 0 to 4) [3].

1.3.2 Non-ionizing radiation dosimetry with p-i-n diodes

A p-i-n diode is a *p-n* junction with an intrinsic region *i* in between, called "base", made of a high-resistivity ($> k\Omega \cdot m$) *p* or *n* layer of usually 300 μm thickness. Due to the higher resistance of the intrinsic silicon path most of the potential drops across the base region. A p-i-n diode scheme is shown in Figure 1.6.

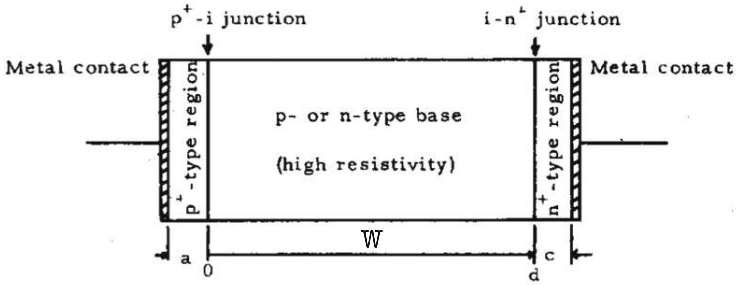


Figure 1.6: scheme of a *p-i-n* diode. [7].

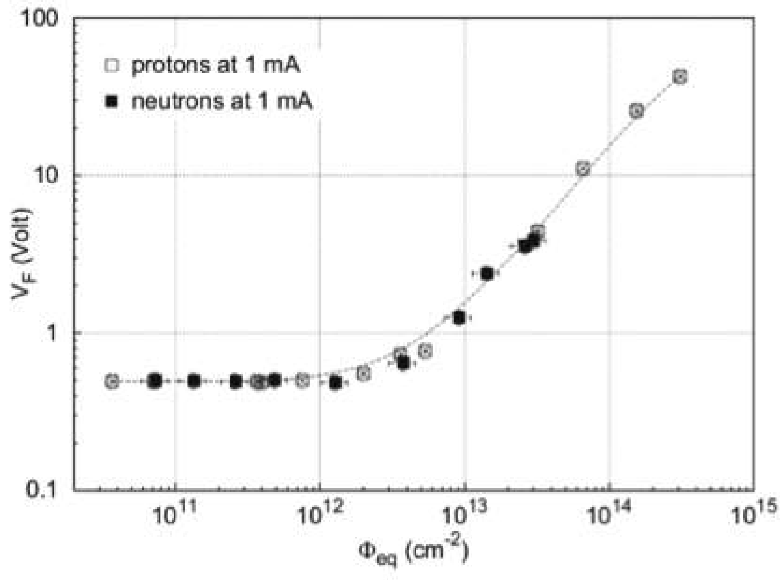
As written in equation (1.2), Due to exposure to non-ionizing radiation, the average lifetime of charge carriers τ decreases and the base resistance R_{base} increases. Consequently, following Ohm's law, this increase in resistance R_{base}, at a constant current I_F, leads to a variation in the diode forward voltage V_F, according to the equation

$$V_F = R_{base} \times I_d \propto \frac{W^2}{\tau} \times I_F \tag{1.2}$$

It can be demonstrated [5] that the dependence on the particle fluence Φ of the minority carrier lifetime τ in an irradiated p-i-n diode is described by the following equation

$$\frac{1}{\tau} - \frac{1}{\tau_0} = K_T \Phi \tag{1.3}$$

where τ_0 is the average minority carrier lifetime before the irradiation and K_T is a constant depending on the material, which accounts for the radiation damage. Figure 1.7 shows the V_F vs the particle fluence Φ results at I_0=1 mA obtained from the experimental characterisation of a BPW34F p-i-n diode exposed to protons and neutrons.



(a) I_F = 1 mA

Figure 1.7: calibration curve of a p-i-n diode BPW34F at 1 mA bias current.[8].

1.3.3 Readout configuration

As showed in Figure 1.8, both the RadFET and the p-i-n diode, appear to the readout electronics simply as passive loads and can be measured by connecting all of them them in parallel to a current generator.

The readout is then performed, first by connecting the desired dosimeter to the current generator (ON state) and then by reading the voltage drop V_t across the device under test when the bias current I_{ds} is forced through it. When no measurements are done, all the pins are connected to GND (OFF state) to avoid any fluctuation of the gate potential during the exposure (gate *zero-bias* configuration). A change in the gate-source potential would modify the RadFET sensibility and all the measurements would be wrong.

Both time and bias current to be applied on RADMON dosimeters, are precisely specified by the dosimeters producers and have been finely defined in Federico Ravotti's PHD thesis [8].

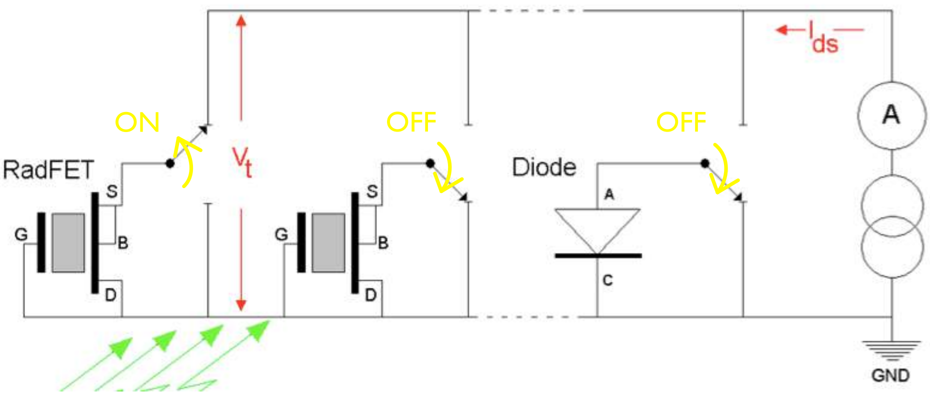


Figure 1.8: scheme of the dosimeter measuring circuit.

Chapter 2

Readout System Design and Test

Nowadays RADMON dosimeters are mainly used in the infrastructure of the LHC and its experiments with the purposes listed in Section 1.2. In such big experiments, all the generated data is collected by a complex system with its protocols and standard electronics. The same RADMON dosimeters, have been installed by the Detector Technology group in other smaller experiments or hazardous places to be monitored and, in order to avoid the use of the complex network of the LHC, another detached readout station has been set up using less complex but still expensive commercial bench electronics along with a LabView based software (Figure 2.1).

The work on a new RADMON readout system has been start to satisfy the need of having, instead of one big readout station, many smaller, lighter and cheaper systems to be shipped, along with the dosimeters, to the end users.

Other two students worked in the past on the portable RADMON readout system: Vincent Schott, who worked on the software and on the first prototype, and Mougel Loic, who focused on the hardware. My role was to review and update the hardware part, by correcting the errors and the problems with the first versions of the RADMON readout system and to develop a new firmware and user interface for it.

This chapter will first briefly describe today's LabVIEW based equipment and then will focus on describing the hardware of the new RADMON readout system.

2.1 Present RADMON readout system

The present RADMON readout system, is based on commercial electronics. Sensors are biased and readout using lab instrumentation:

- a *Keithley SourceMeter* is used as constant current generator,
- an *Agilent 34970A Data Acquisition / Switch Unit* enables sequential reading of the devices under test connected to different channels.

A program implemented in the LabVIEW environment is used to control those instruments and to manage the readout. A picture of the setup is shown in Figure 2.1, while a block diagram of the same readout system setup is shown in Figure 2.2 .

Figure 2.1: today's dosimeter Data Acquisition System with LabVIEW based PC on the left and *Keithley SourceMeter* and *Agilent Switch Unit* on the right [3].

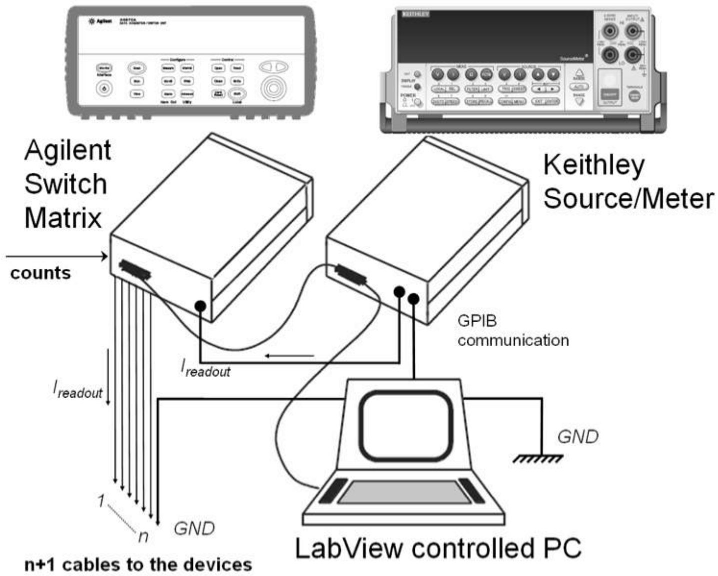


Figure 2.2: block scheme of the Data Acquisition System.

2.1.1 Design specifications for the new project

The new project is supposed to ensure performance be as close as possible to the present one providing in addition a cheaper "All-In-One" alternative, lighter and easy to carry around. The protocol for reading RADMON dosimeters, which remains unchanged [8], ensures the needed precision and the right current amplitudes to avoid the dosimeters from breaking. The protocol adopted is based on the experience accumulated in years of calibration and characterisation of each dosimeter [9]. As mentioned in Subsection 1.3.3, the readout of the RADMON dosimeters is performed treating them like if they were simple resistors. The current and the timing to be used for the measurements are provided in the following.

- For the RadFET dosimeters, the voltage is read once during a time varying from 1 to 5 seconds at a constant bias current of 100 μA for the LAAS and of 160 μA for the REM.
- For the p-i-n diodes, the bias current can not stay on for longer than 50/100 msec to avoid the deterioration of the junction. The bias current is of 1 mA for the BPW and CMRP, while the LBSD needs 25 mA.

2.2 Global view of the new RADMON readout system

The new readout system under design consist mainly of three parts:

- high voltage power supply,
- multiplexing circuit,
- the main-board,
- micro-controller.

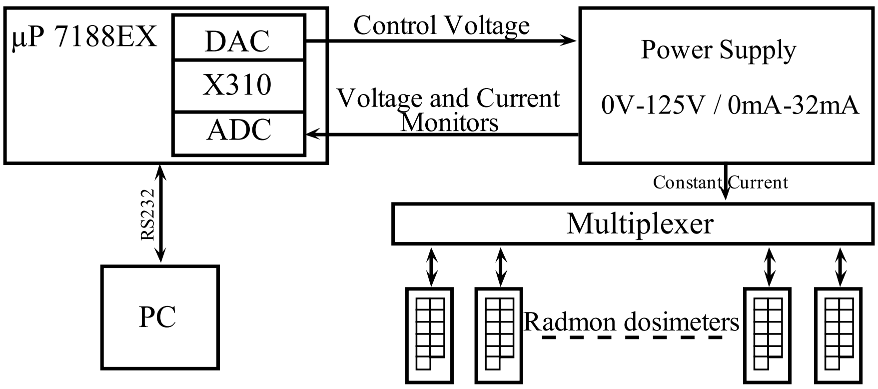


Figure 2.3: schematic diagram of the new readout system [3].

As shown in Figure 2.3, the new system is controlled by a PC, which drives the ICPDAS's *7188EX* micro-controller, expanded with an external 12 bit DAC and 12 bit ADC of the *X310* board. Through the PC it is possible to communicate with the micro-controller and than to drive the UltraVolt's HVPS power supply in order to generate the bias current needed for the readout. The *7188EX* then reads, from the power supply monitor, the analog voltage, which is the voltage across the dosimeter.

2.3 High voltage power supply circuit

A high voltage power supply is fundamental for reading the RADMON dosimeters. As showed in the first chapter, in general, the RADMON dosimeters provide a voltage at their terminals depending on the current that is forced into them. This voltage covers quite a large range (from 0 V up to 80 V), compared to the usual voltages used in the rest of the readout system (24 VDC for the microprocessor's VCC). To simplify the design of this high voltage circuit, we decided to use a commercial product.

2.3.1 Detailed specifications

The specifics for the high voltage power supply are:

- an output voltage varying from 0 V to 80 V, with an admissible ripple of 30 mV_{pp},
- an output current going from 100 μA to 25 mA, with a ripple lower than 10 μA,
- capable of dealing with the worst case operating conditions and time constrains which are:
  - 10 V @ 100 μA for up to 5 sec (LAAS),
  - 40 V @ 160 μA for up to 5 sec (REM),
  - 80 V @ 1 mA for no more than 100 msec (BPW),
  - 10 V @ 25 mA for no more than 100 msec (LBSD).
- one output current monitor and one output voltage monitor to be used by the microprocessor for the readout.

2.3.2 UltraVolt's high voltage power supply

The first power supply used in the project was the *1/8AA12-N4*, purchased (in 2007) directly from UltraVolt, Inc., a world-leading manufacturer of high voltage power supplies. That model, at that time, was the only one providing a good trade-off between performance and size for portable system.

The *1/8AA12-N4* works with a V_{cc} of 12 VDC and provides a negative output voltage V_{hv}, varying from -125 V to 0 V and proportionally to the voltage (from 0 V to 5 V) applied on V_{remote}. The maximum output power, declared on the data-sheet, is 4 W, while the maximum output current is 32 mA.

In order to use that voltage generator as a (constant) current generator, an additional outside controlling circuit is needed. As showed in Figure 2.4 it is possible to control the current I_{out} by feeding back to the input the voltage from the current monitor pin. This terminal provides a voltage V_{mon}, ranging from 0 V to 5 V and proportional to the output current, which is first amplified through U1.A and is then compared with the controlling voltage V_{in} in U1.B, giving the negative feedback for keeping the output current constant. When the load increases, the output voltage increases as well, because the current remains constant. Both values of the output voltage and the output current are then read by the micro-controller and processed. In this way, by adjusting the U1.A stage gain, it is possible to reach an optimal ratio between V_{in} and I_{out}. For example, by setting a higher gain, the feedback response becomes much faster and sensitive to variations of the control signal, but at the same time it sets a limit on the maximum applicable signal V_{remote} (the zener diode D6 prevents V_{remote} from getting higher than V_{zener})

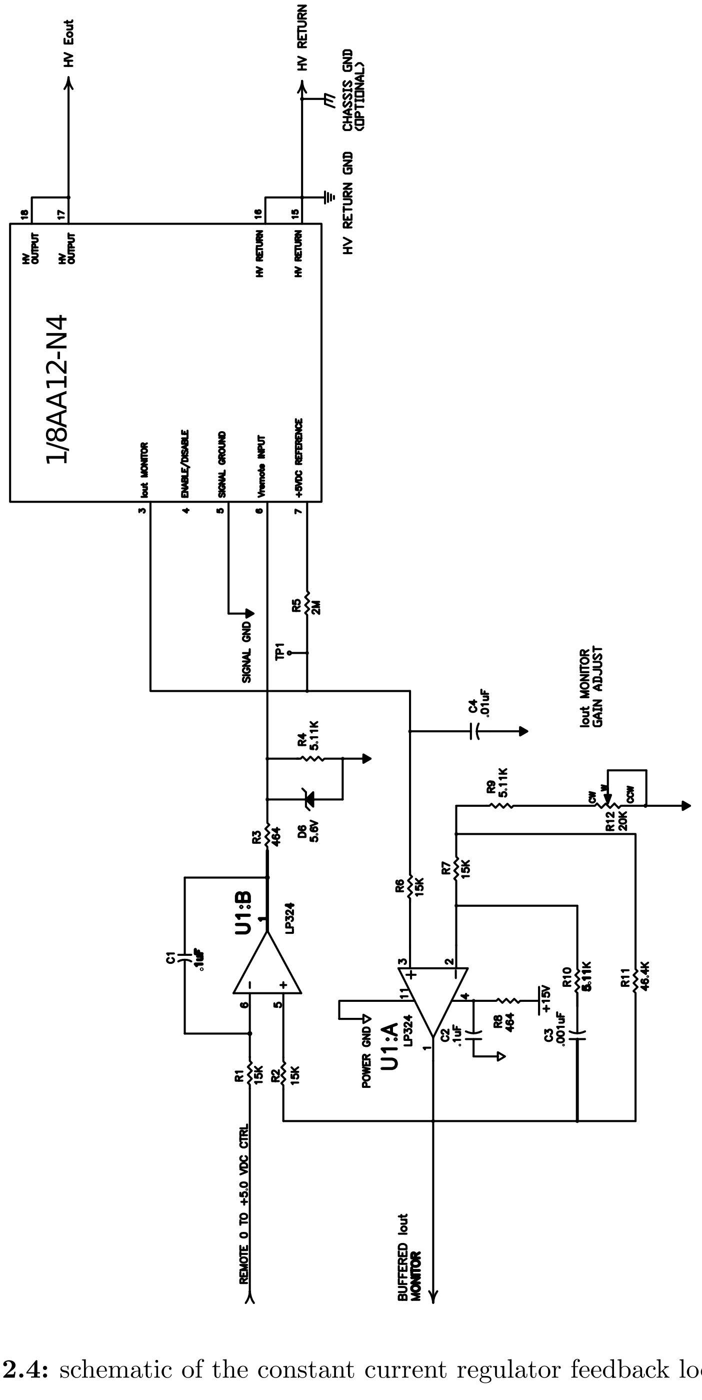
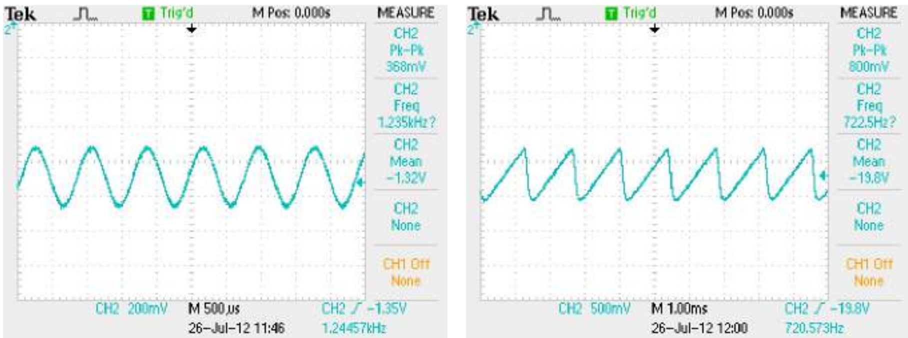


Figure 2.4: schematic of the constant current regulator feedback loop [10].



(a) **(b)**

Figure 2.5: screen shots of the output voltage as seen at the oscilloscope: (a) R_{load} = 60 Ω , V_{in} = 5 V, LoopGain = 2.60 V/V; (b) R_{load} = 130 kΩ , V_{in} = 137.5 mV, LoopGain = 2.60 V/V.

UltraVolt does not provide any type of PSpice model or detailed data sheet. In order to perform a circuit simulation, in particular to study stability issues, a model for the *1/8AA12-N4* power supply had to be created.

In order to make a stability simulation of the circuit in PSpice, it was decided to model the power supply using an ideal voltage controlled voltage source with finite gain, in series with an RC filter (necessary to account for the delay introduced by the power supply itself). The gain has been measured by studying the input-output DC characteristic of the block, while, to be able to quantify the delay, the output response to a step was evaluated.

While the circuit was found to work correctly and to be able to comply with the specifications, the experimental characterization of the power supply provided an unexpected, seemingly oscillatory behaviour. As showed in Figure 2.6, the high voltage power supply circuit has been tested in various configurations looking for the causes of those oscillations.

Figure 2.5 shows the oscillation at the power supply output as obtained in two different configurations. The measurements have been done following the worst cases listed in subsection 2.3.1. In Figure 2.5a the higher current of 25 mA is forced through the lower resistive load of 60 Ω and the gain is chosen accordingly to the input voltage V_{in} which was set to the maximum value of 5 V; while in Figure 2.5b, the measurement has been done on the higher load of 130 kΩ, forcing the lower current of 160 μA, setting the lower

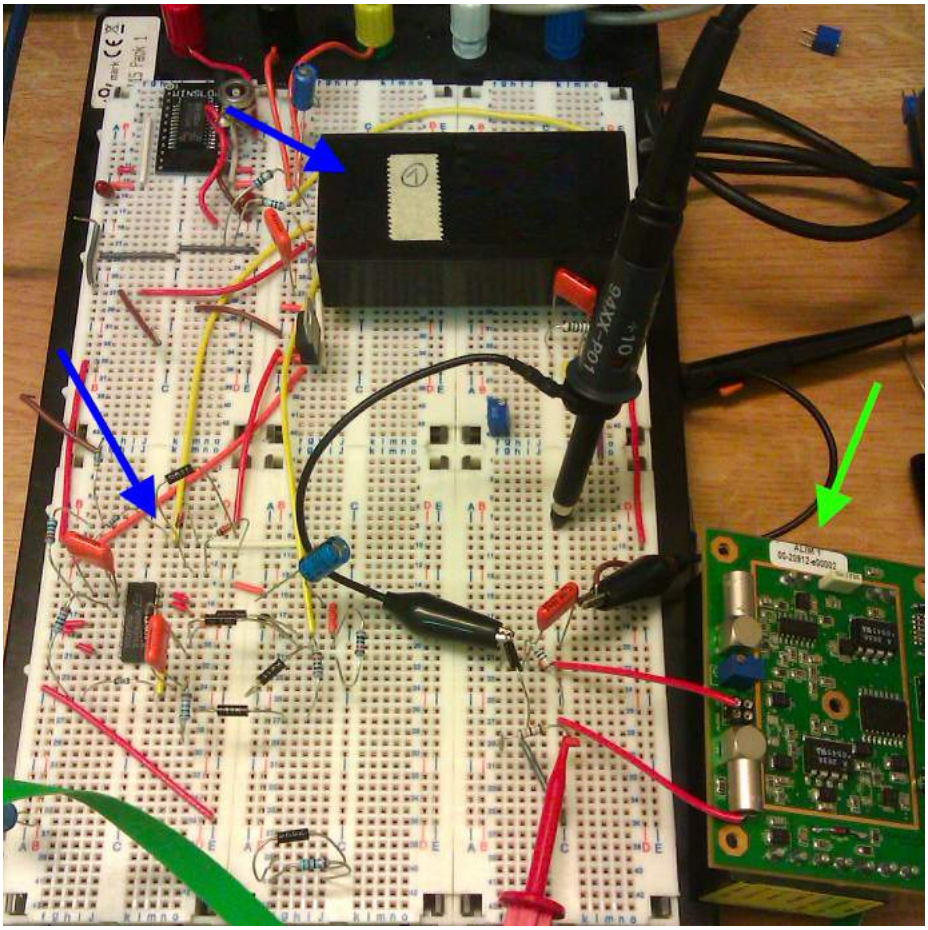


Figure 2.6: breadboard used for testing. A *1/8AA12-N4* power supply is tested standalone or by using the external circuit (blue arrows) and another *1/8AA12-N4* is tested using a custom PCB with the circuit in Figure 2.4 (green arrow).

V_{in} of 137.5 mV. The oscillations seen in both the measurements in Figure 2.5, do not comply with the design specifications and may be dangerous if applied to a dosimeter. Better performance can be obtained by using large capacitors to shunt the $V_{monitor}$ and V_{remote} pins, but in that case the time constraints required are not satisfied any more.

2.3.3 UltraVolt's -I10 prototype

By directly contacting the UltraVolt company, it became clear that the *1/8AA12-N4* power supplies we had were not working properly. According to the company some internal compensation capacitors had got broken or were experiencing manufacturing defects. As a replacement for the faulty ones, we have been informed of a new model introduced in June 2012 [11]. The new feature of the power supply consists in the enhanced interface option -I10, which includes low output impedance current and voltage monitors, and the possibility to choose between a constant output voltage or current just by setting a maximum value on one controlling pin and by varying the other.

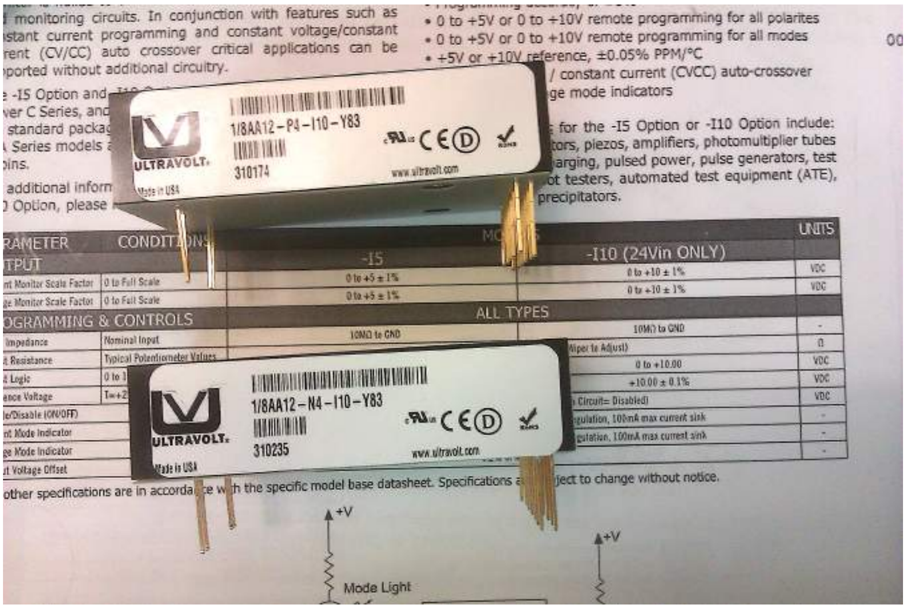


Figure 2.7: photo of two UltraVolt brand new prototypes (-Y83), a positive output 1/8AA12-P4-I10 and a negative output 1/8AA12-N4-I10.

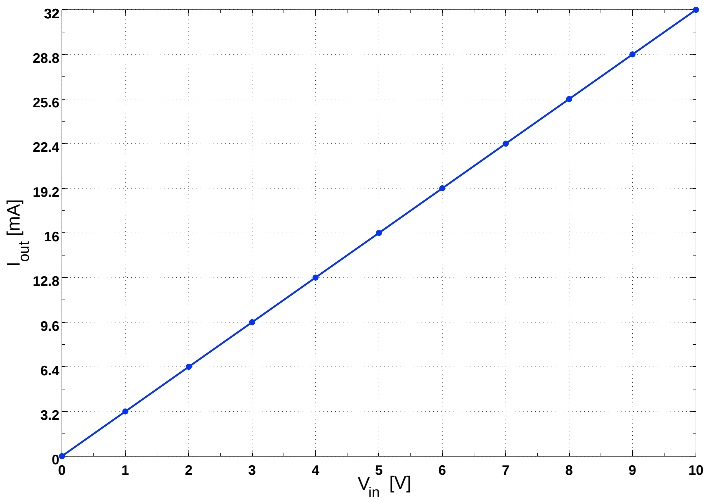


Figure 2.8: V_{in} vs I_{out} characteristic of the positive 1/8AA12-P4-I10.

The two prototypes we received from UltraVolt (in Figure 2.7) need a VCC of only 15 VDC, while having a controlling voltage V_{in} from 0 V to 10 V, applied on the voltage programming pin V_{prog} or on the current programming pin V_{prog}, with an output voltage V_{out} from 0 V to 125 V or an output current I_{out} varying from 0 mA to 32 mA (the voltage/current characteristic is showed in Figure 2.8). The data sheet for this model was not available yet when that samples were sent, but all the necessary information were given directly by the manufacturer. Because of its internal feedback and compensation circuits, there is no more need of an external controlling circuit. Therefore, the testing were much easier to do by just connecting the power supply to the rest of the circuit (as showed in Figure 2.9).

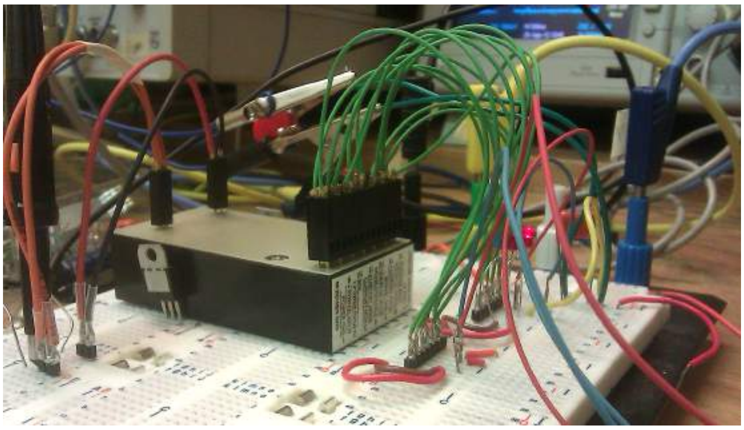


Figure 2.9: test bench for the new power supply.

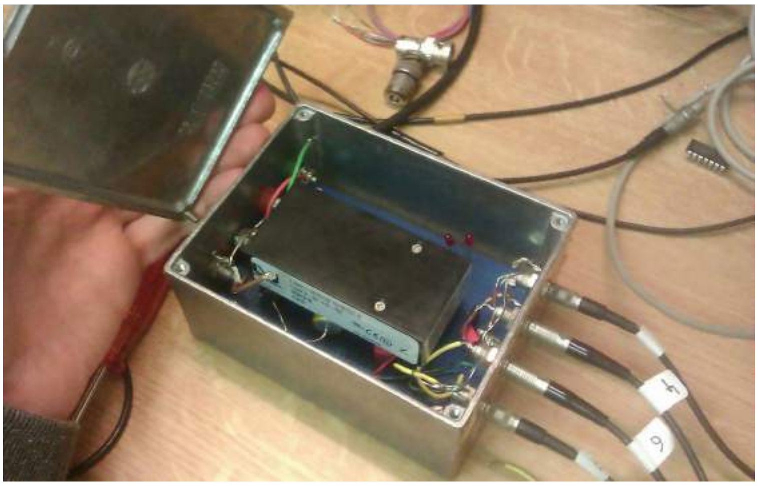


Figure 2.10: metal box for the new power supply with two leds indicating if the block is working as current or voltage generator.

Schematic and layout

An actual PCB for the new power supply have not been produced because of the lack of time. However, the power supply has been inserted, with all the necessary shunt capacitors and led indicators soldered on a stripboard, in a dedicated metal box to avoid noisy floating connection and to clean up the work bench (Figure 2.10). A schematic circuit and design for the future PCB are showed below in Figure 2.11 and Figure 2.12.

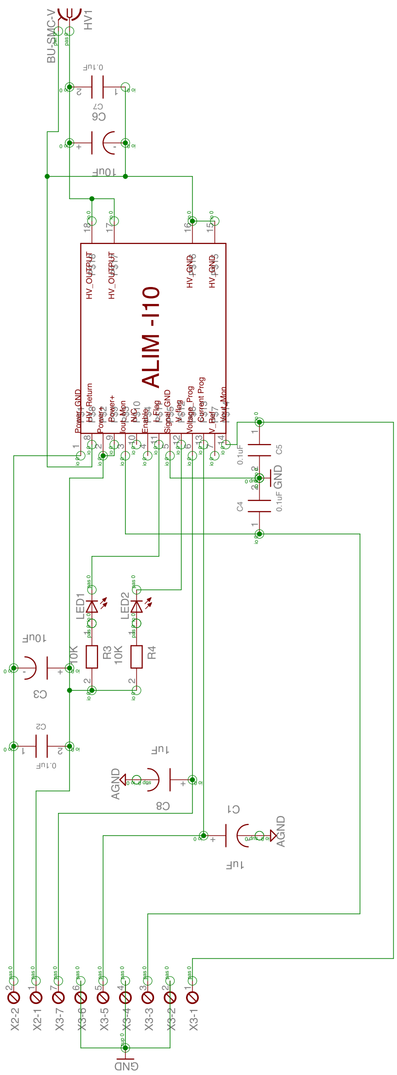


Figure 2.11: schematic of the controlling circuit for a *1/8AA12-N4-I10-Y83* new power supply.

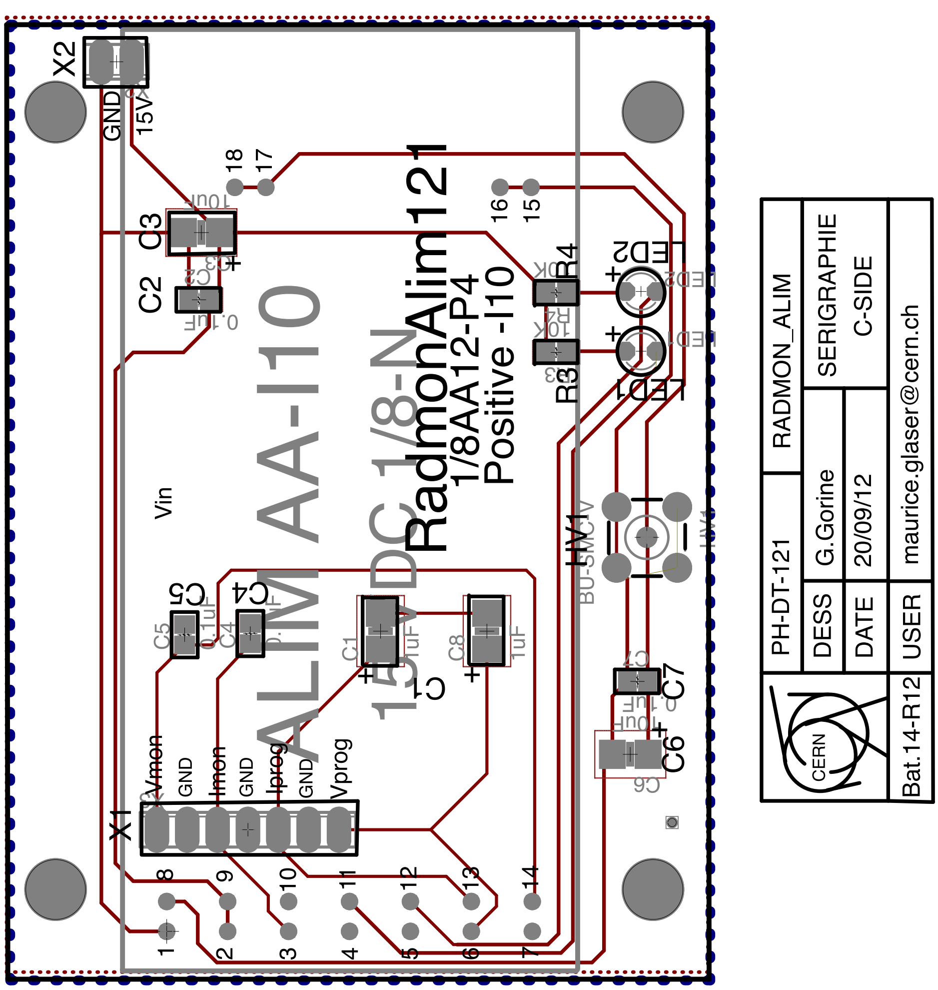
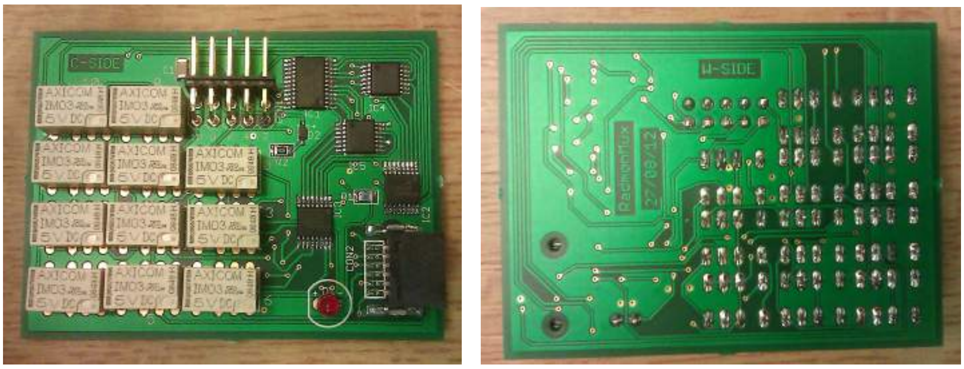


Figure 2.12: possible layout for a *1/8AA12-N4-I10-Y83* new power supply board.

2.4 Other electronic boards

2.4.1 Multiplexing circuit

The multiplexing circuit, showed in Figure 2.13, was designed for making the new RADMON readout system capable of multichannel reading. The main IC used is the 4-to-16 multiplexer *MM74HC154*, which is mounted on each of the 15 multiplexing circuits connected to the main-board. The multiplexer needs 4 selecting bits (2^4=16 configurations) to pull down one of the 16 output connected to an inverter *(MC74VHC14DTG)* and then through a Darlington buffer *(DS2003TMT)*, connected to each of 11 relay *(IM03TS)* needed to manage the 11 dosimeters contained on a single RADMON sensor carrier (Section 1.3 and Figure 1.2). The inverter was needed because there were not any negative-logic multiplexer on the market and the selection is made by pulling up only the needed relay. The Darlington buffers are used to drive the relays. When changing the state of a relay switch, a large current spike can occur and buffers are needed for providing the extra current. The *MC74VHC14DTG* can also protect the controlling circuit from backwards currents thanks to an internal protecting diode. The relay is made of two switches that commute simultaneously as soon as the control pin is pulled up to 5 V. When off, the relay connects both the terminal of the dosimeter to GND, while when it is on, it connects the terminals to pin HV and HV_{gnd} of the UltraVolt power supply. In that way, when not doing measurements, the dosimeters are not affected by the power supplying circuit and most important, no potential could rise in the RadFETs gate oxide, avoiding modifications to their sensibility (see Section 1.3.3). A 3 mm red led is mounted on top of each board and lights up whenever the enable signal of the selected multiplexing board is high.



(a) **(b)**

Figure 2.13: multiplexing PCB front-side (a) and back-side (b).

Detailed specifications

The multiplexing circuit is required to

- be capable of selecting the 11 channels of the Integrated Sensor Carrier,
- guarantee the connection of both the terminals to ground of a dosimeter, when it is not under measurement,
- be able to deal with high voltage signals,
- be compatible with the 10 pin socket already used in the main board,
- fit inside the metal box with other 14 multiplexing circuits.

Schematic and layout

The complete sequence of ICs described in Section 2.4.1 is shown in the schematic (Figure 2.14) and in the layout (Figure 2.15) both designed using the CAD software Eagle and printed by a French company named Cirly [12].

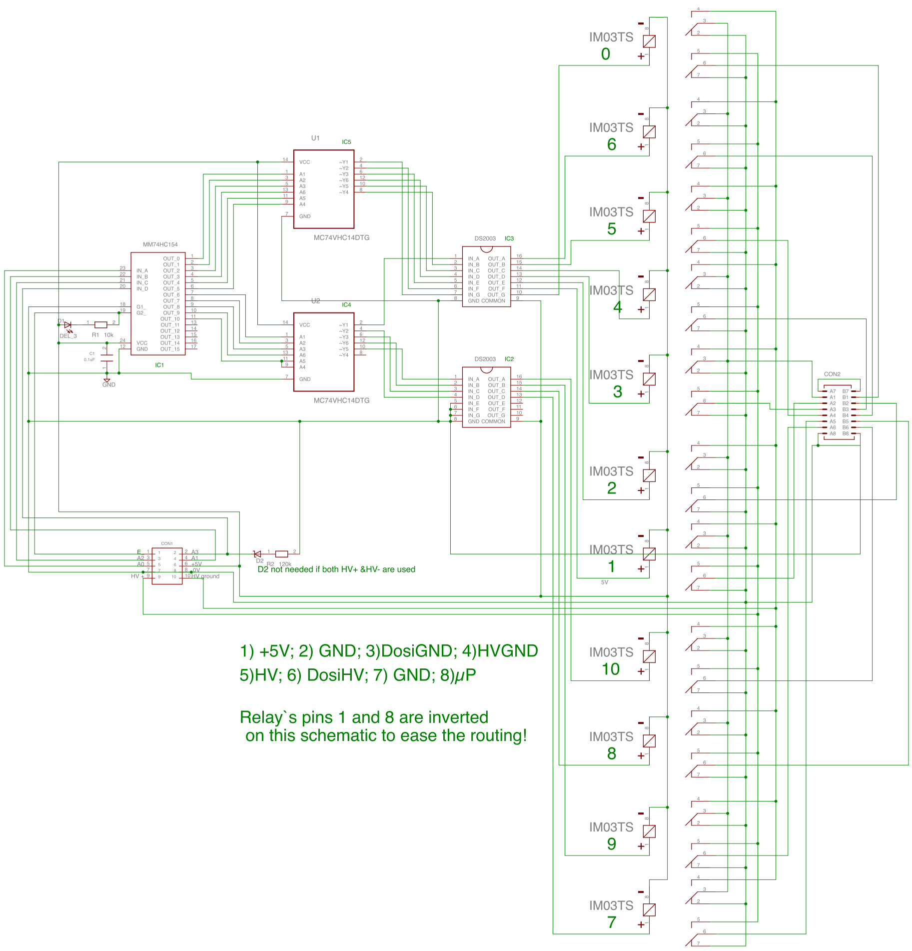
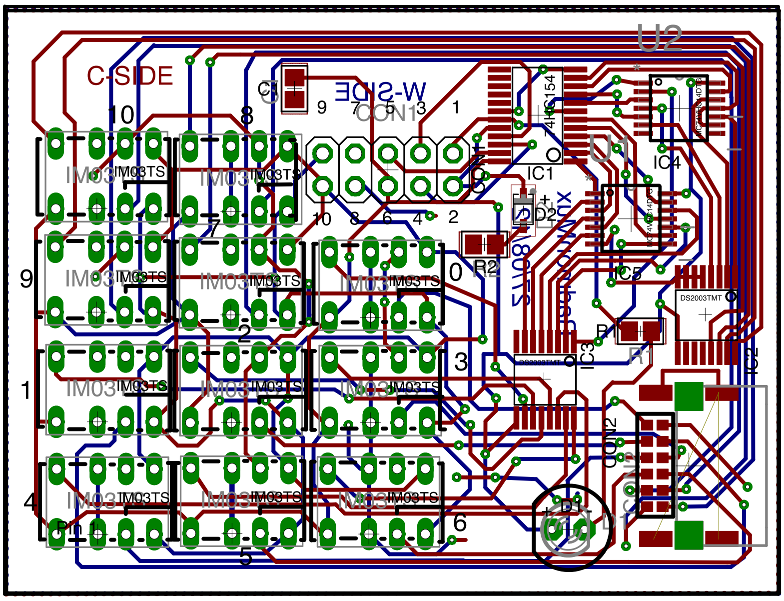


Figure 2.14: schematic of the 11 channel multiplexer circuit.



CERN	PH-DT2-SD-001	RADMON_MUX_001	
	DESS	G.Gorine	SERIGRAPHIE C-SIDE
	DATE	25/08/12	
Bat.14-R12	USER	maurice.glaser@cern.ch	

Figure 2.15: layout of the 11 channel multiplexer circuit.

2.4.2 Main-board circuit

The main board circuit is the most important of the whole RADMON readout system as it collects all the signals, provides the power for all the ICs and manages the first stage of the high voltage routing. The first prototype, provides sockets for the UltraVolt *1/8AA12-N4*, ICPDAS *I-7188EXD* micro controller and the 15 multiplexing boards showed in the previous section. The logic on this board is managed by three 8-bit shift registers *74HC594*, which are needed to cope with the lack of digital output pins on the micro controller. This series of ICs, not only can generate a sequence of $8 \times 3 = 24$ bits but also provide an additional output register. The first 4 bit, connected to four 3 mm red leds, are used to select the right channel on the multiplexing board (they directly drive the 4-to-10 demux *MM74HC154* described in Section 2.4.1). The remaining bits are used to create a sequence (later inverted by the inverting buffers *74HC240* to maintain consistent logic signals) of zeros except for one used to enable a specific demux on a specific multiplexing board. As a result, the micro controller is able to deal with all the channels just by using 3 digital outputs. The schematic of the prototype used during my intern-ship is showed in Figure 2.16. Even if, after a review, a new design was needed, in order to save time and money, the modifications have been done directly on the board by soldering missing elements or by removing unnecessary ones.

Detailed specifications

The main board is required to

- provide all the voltages needed in the system (24 V for the microprocessor, 15 V for the power supply, 5 V for the ICs),
- route all the signals and interconnect all the single boards,
- fit the metal box chosen for the system.

Schematic and layout

In Figures 2.16 and 2.17 below, are showed the schematic and the layout of the board currently used. A new layout including the necessary corrections will be issued when the design is complete.

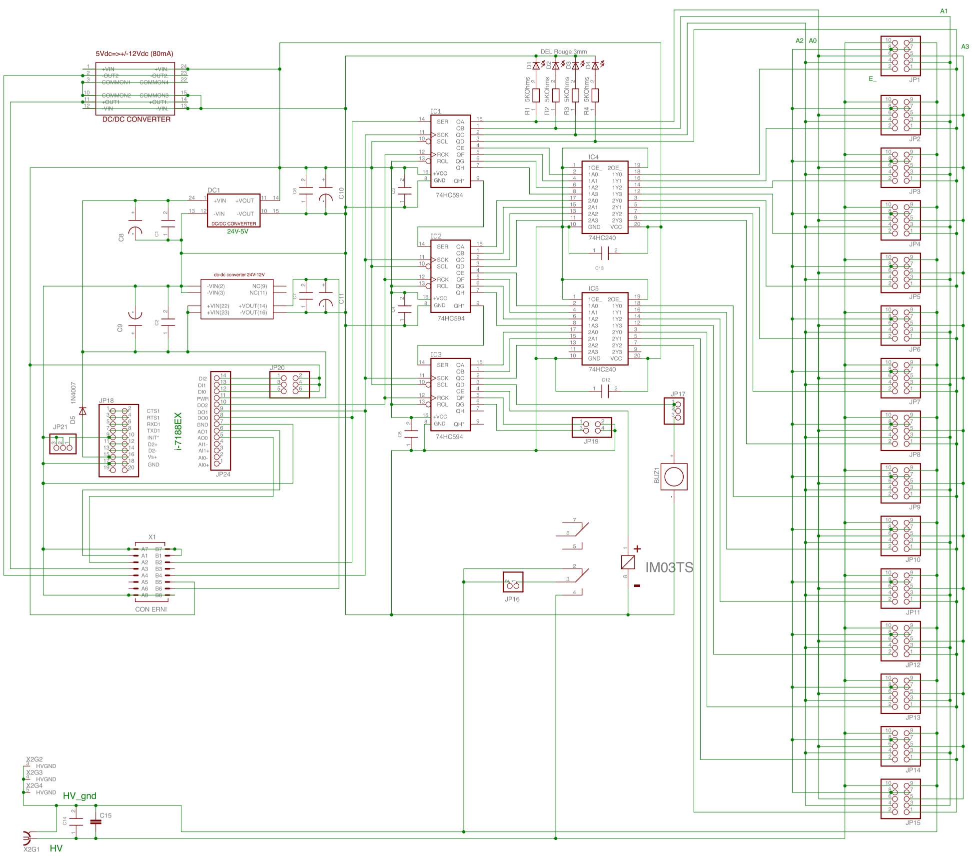


Figure 2.16: schematic of the main-board.

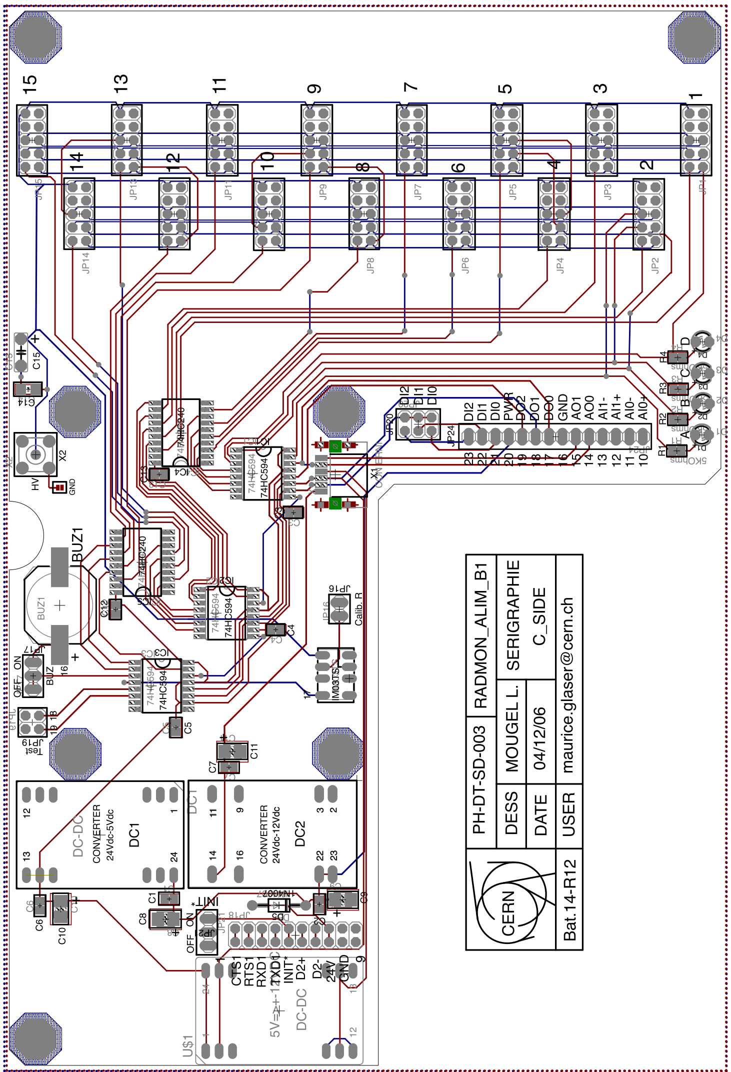


Figure 2.17: produced layout of the main board.

2.5 Final RADMON readout prototype

The finished prototype, containing all the boards described before, is shown in Figure 2.18. The main-board is inside a metal box with a power line socket and an RS232 female connector. The box is equipped with an AC/DC power supply (220V to 24V), which powers up the voltage regulators on the main-board. The 15 multiplexing circuits (in Figure 2.18 there is only one) are connected one next to each other to the main-board, and then, through the green flat cable, the Integrated Sensors Carriers are connected to the multiplexing circuits. In the other box there is the UltraVolt power supply which is connected to the ICPDAS micro controller and brings the high voltage to the main-board through the brown lemo cable. In the figures of the Appendix A are shown useful schemes to better understand the interconnections between each circuit.

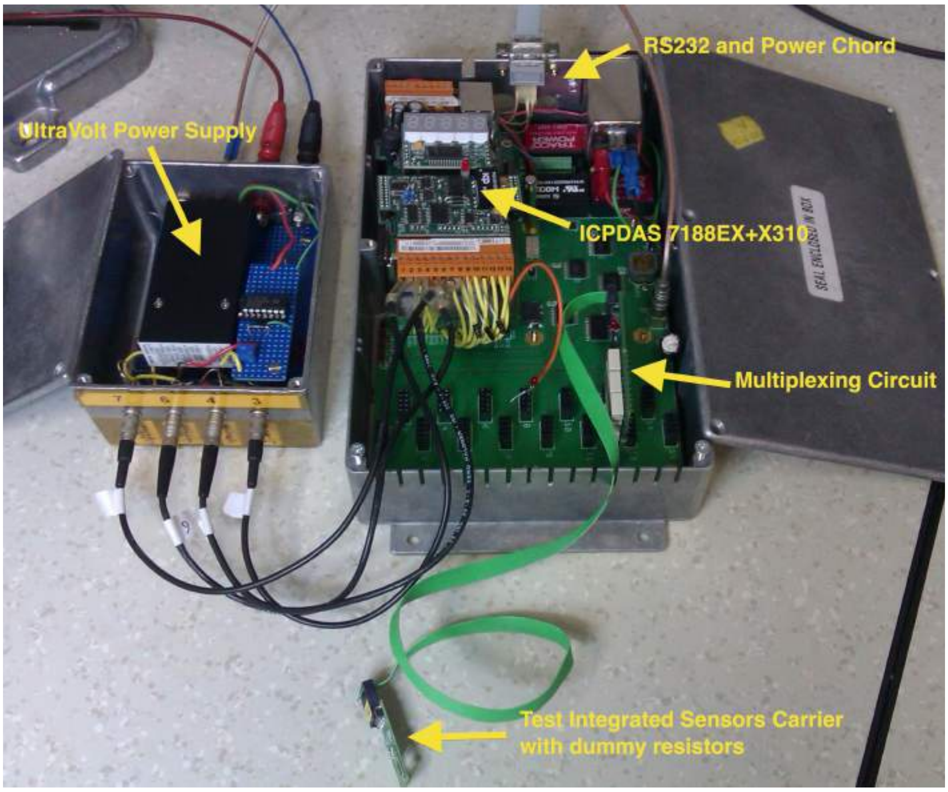


Figure 2.18: overall view of the RADMON readout system prototype.

Chapter 3

User Interface and Embedded Programming

As mentioned in the previous chapters, in order to control the RADMON readout system, a microprocessor built by ICPDAS has been used. Because of the totally new hardware developed during my intern-ship, it became necessary to develop a new software interface to first debug and then program the dosimeter readout.

This chapter will show and describe the source code of the new programs written to control the RADMON readout system prototype.

3.1 Current LabVIEW program

The characterization and measurement results discussed in Chapter 1 along with today readout, are handled by a program written in LabVIEW (by National Instruments) and the setup described in Section 2.1. The interface, shown in Figure 3.1, makes it possible to start a new set of measurements by just uploading a configuration file and pushing "Start Acquisition". There is also a manual way of setting the configuration by choosing a channel and changing the readout time, the number of acquisitions and the amplitude of the output current to provide. To communicate with the *Keithley SourceMeter* and the *Agilent 34970A Data Acquisition/Switch Unit*, the LabVIEW based software exploits the NI-VISA (National Instruments VXI plug & play Systems Alliance), a standard I/O language for instrumentation programming, via the GPIB (General Purpose Interface Bus) interface.

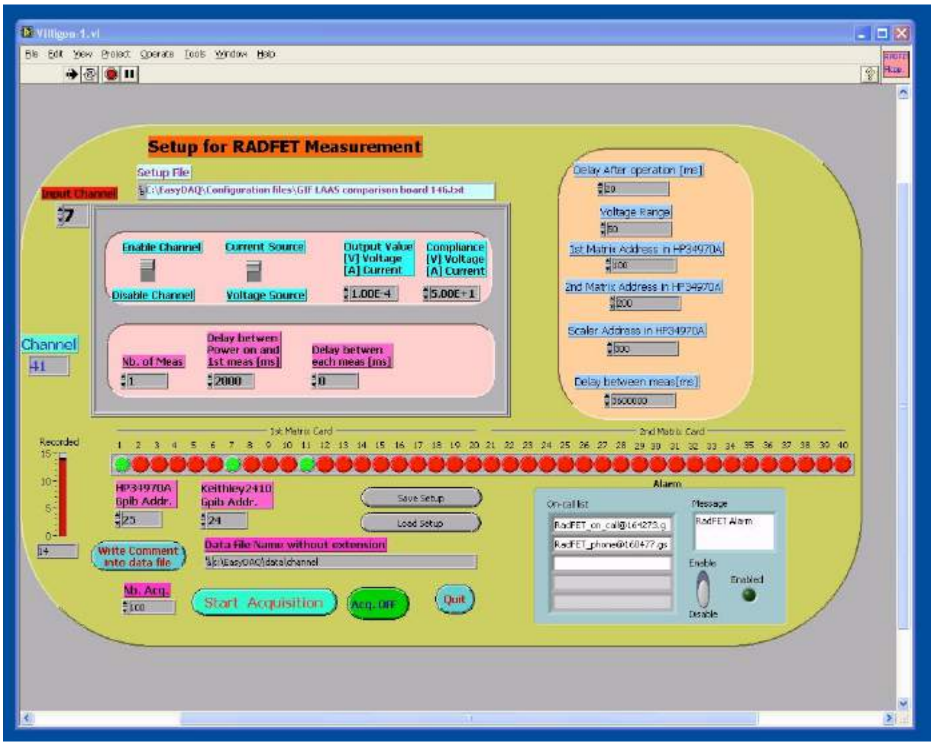


Figure 3.1: Labview based software, written by M.Glaser, controlling the today RADMON readout system.

3.2 Software details

The general functions to be implemented by the software inside the RADMON readout system are as follows:

- the system has to be autonomous, meaning that it should be able to take measurements without the need of human interactions.
- the system must be capable of performing multi data acquisition; the hardware of the RADMON readout system has been designed to control up to 15 boards with 11 dosimeters on each one, meaning that the software has to manage up to 161 consecutive readout operations.
- the system should store in a buffer memory not only the dosimeter voltage drop but also the temperature at which the measurement has been done and the precise start and stop times.
- the system should comply with the timing and amplitude specifications listed in section 2.1.1.

3.3 ICPDAS I-7188EXD microprocessor

The first beta software was supposed to work on a *I-7188EX* embedded micro-controller, produced by ICPDAS, already available in the Detector Technologies labs. The *I-7188EX* is powered by an AMD 80188-40 processor with 512K bytes of static RAM, and 512K bytes of Flash memory [13]. It provides 14 user defined I/O pins which can be upgraded by an external I/O expansion board with improved ADC and DAC. For our application the X310 expansion board has been chosen providing a 12 bit ADC and a 12 bit DAC. In this way the micro controller provides

- 2 differential analog inputs (AI0/AI1 from 0 to +10V),
- 2 analog outputs (AO0/AO1 from 0 to +10V),
- 3 open collector digital outputs (DO0/DO1/DO2),
- 3 common ground digital input (DI0/DI1/DI2).

The *I-7188EX* can be connected to a PC (using a serial RS-232 port) and programmed in C language (using the MiniOS7 development tool).

3.4 Terminal based debugging software

A “terminal based” software indicates a program which runs without offering a graphic interface but communicates with the user through a command-prompt shell. The software is obtained by compiling the file *terminal.c* with *BorlandC ++3.1*, an IDE environment capable of building executable files for 80186/88 processors.

In order to use built-in functions and to make the programming of the *I-7188EX+X310* easier, different libraries are included in the file *terminal.c* with the following code lines:

```
#include <stdlib.h>
#include <stdio.h>
#include "lib/7188e.h"
#include "lib/X310.h"
```

The *stdlib.h* and *stdio.h* are standard libraries containing definitions for common types, variables, and functions along with definitions for basic input/output stream. Both *7188e.h* and *X310.h* are libraries provided by ICPDAS containing functions for interacting with the *I-7188EX* micro-controller and the *X310* expansion. While the *7188e.h* is used for lower level interactions with the module, like writing and reading the EEPROM or the flash memory, the *X310.h* library is used for directly managing the input/output pins, by means of the functions listed in the following:

```
float X310_AnalogIn(int iChannel);
void X310_AnalogOut(int iChannel,float fValue);
int X310_Read_All_DI(void);
int X310_Read_One_DI(int iChannel);
int X310_Write_All_DO(int iOutValue);
int X310_Write_One_DO(int iChannel, int iStatus);
int X310_Read_All_DO(void);
int X310_Read_One_DO(int iChannel);
```

As soon as the *I-7188EX* module is powered up, with the auto-run pin set high so that the program *terminal.exe* inside the ROM is executed, and as soon as the connection with the PC is established through the *7188x.exe* (a command-line utility provided by ICPDAS for sending data through RS-232), the text in figure 3.2 shows up on the screen prompting the user to choose between different debugging options listed below

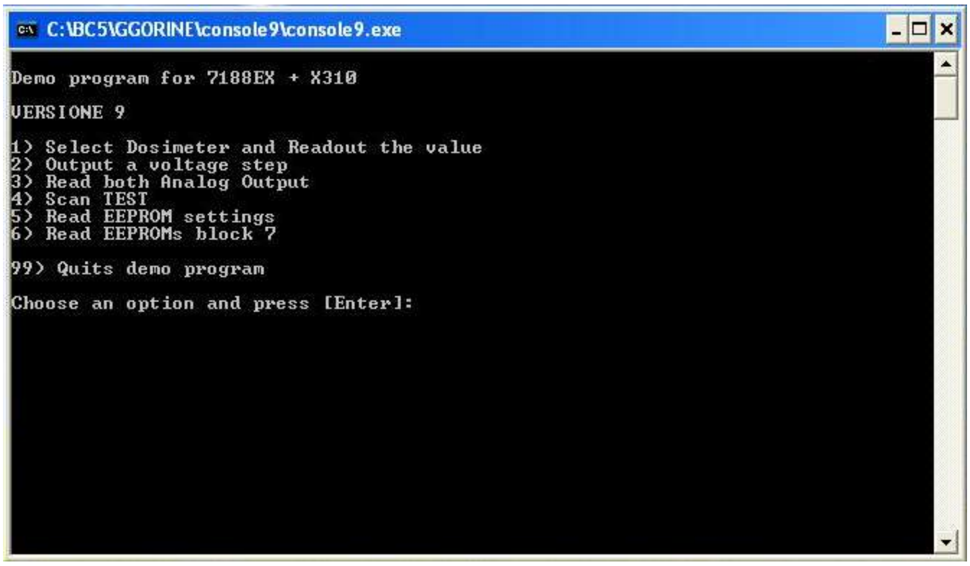


Figure 3.2: Terminal based software interface

- Option 1 executes the actual readout, prompting the user to choose the dosimeter and its channel, to set a maximum output voltage, the constant output current amplitude and the time for the measurement. Then a routine starts the readout of the selected dosimeter showing on the screen the voltage and current read on the monitors with the final mean values and equivalent dosimeter resistance.
- Option 2 starts the routine used for testing the HV power supply performances, such as rise time and output ripple.
- Option 3 prints, once per second, the values of the analog inputs of the *I-7188EX*, making possible to calibrate the monitors and checking the noise red when the readout is off.
- Option 4 scans all the dosimeters, checking their connections and the relays operation on the multiplexer boards.
- Option 5 and 6 read and set the calibration values of the ADC and DAC on the *X310* board.

A number of user defined functions are included *terminal.c* to manage different tasks, namely

```
void resetALL(void);
int invertNumber(int);
void display7(int,int);
void clockSR(int);
void stepMaker(int, float, int); //function that allows to generate steps
void selectDosi(int,int);
void setHV(float,float,int); //sets current and voltage compliance
void readDosimeter(unsigned int);
void analogRead(int);
void scanTest(void);
```

The task performed by each of the above function is described by the following: (the line numbers in square brackets are referred to *terminal.c* source code in Appendix B)

- **resetALL()**[182-188] resets the system by selecting the free channel 15 (multiplexing boards go from channel 0 to 14) and setting the high voltage V_{hv} to zero;
- **invertNumber()**[190-201] inverts bitwise a binary number (e.g. 0100 becomes 0010);
- **display7()**[203-215] shows numbers on the 7-segments display of the module;
- **clockSR()**[217-222] makes a 1 to 0 transition on the selected output pin.
- **stepMaker()**[225-231] generates a step of a given amplitude on a selected pin;
- **selectDosi()**[233-275] selects the given dosimeter on a given board by sending serially a controlling sequence to the multiplexing circuit;
- **setHV()**[277-299] sets the output high voltage to the needed level by a ramp;
- **readDosimeter()**[301-322] reads the two monitors of the HV power supply and displays the measured resistance;
- **analogRead()**[334-345] reads continuously the monitors and stops when they are both zero (HV power supply is switched off);

- **scanTest()**[347-361] starts a debugging test which scans all the addresses in order to check if the software makes it possible to reach all the 161 dosimeters.

Conclusions

This thesis discussed the realization of a new readout system for RADMON, ionizing and non-ionizing, dosimeters.

The new RADMON readout system have not been used yet to read out real dosimeters. Being still a prototype under development, before trying its performance with an ISC (Integrated Sensors Carrier), some more testing is needed along with a more stable software.

The development of the hardware, as mentioned in the second chapter, will proceed with some planned upgrades:

- the AC/DC adapter will be replaced externally by a commercial one, lowering both the noise and the price of the entire system;
- the PCB showed in Figure 2.12 will be printed;
- a new and smaller main board with less debugging components (such as controlling led, buzzers and extra pins needed for checking signals) will be produced as soon as all the other parts will be definitive;
- a new, more suitable, metal box equipped with standard external sockets for the AC/DC adapter, for the USB/RS232 and a 10 pin socket for all the controlling signals, will be used.

The development of the software described in the last chapter will also continue along two lines:

- the terminal based software will be completed and implemented, and a GUI will be designed to make the program more user friendly;
- the ICPDAS micro controller will be replaced with an Arduino *DUE*.

The Arduino *DUE* [14], showed in Figure 3.3 is an “open source hardware” micro controller of smaller size, cheaper price and with a worldwide community of supporters. An Arduino could perfectly substitute the *I-7188EX*, as it offers the same 12 bit ADC, a greater amount of analog outputs, and can also be programmed in C language. Moreover, by using the Arduino *Ethernet Shield*, it will be possible to give the RADMON system a connection to internet, along with the capability of off-line storage by exploiting the built in microSD card slot.

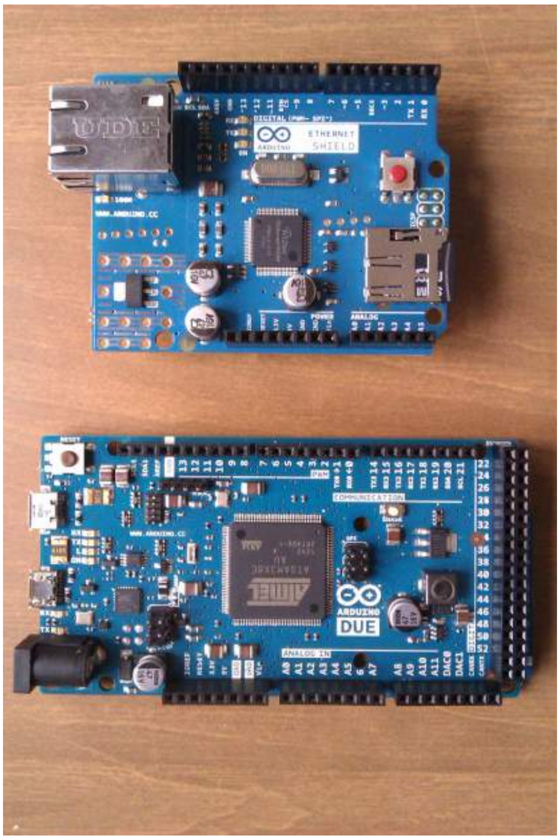


Figure 3.3: An Arduino Ethernet Shield on the top and an Arduino DUE on the bottom.

Appendix A

General interconnections and block diagrams

Figures below show the main signals in each developed board.

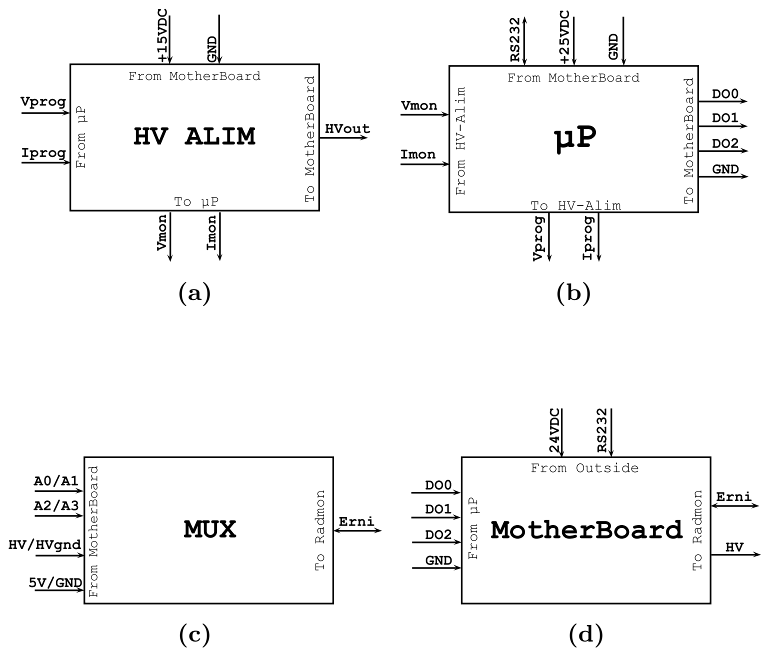


Figure 4: (a) UltraVolt power supply (b) ICPDAS micro-processor (c) Multiplexer Board (d) Main-board

Appendix B

Here follows the source code of *terminal.c*.

terminal.c

```c
#define _READNUM 20
#define _SLOPE 50

#include <stdlib.h>
#include <stdio.h>
#include "lib/7188e.h"
#include "lib/X310.h"


typedef struct{
        int h;  //Hours
        int m;  //Minutes
        int s;  //Seconds
    unsigned long total; //total execution time in [ms]
} TIME;

//MY FUNCTIONS

void resetALL(void);
int invertNumber(int);
void display7(int,int);
void clockSR(int);
void stepMaker(int, float, int); //function that allows to generate steps
```

```
void selectDosi(int,int);
void setHV(float,float,int); //sets current and voltage compliance
void readDosimeter(unsigned int);
void analogRead(int);
void scanTest(void);


void main(void)
{
    int quit,iAction, iFlag, iValue,choice,i;
    float temp=0;
        int channel;
    int boardNum=0,dosiNum=15; //user's choice of board number and dosimeter
    float voltage,current;
    unsigned int time;  //choose voltage (current) and pulse time
    TIME beginTIME,endTIME;

    int version= _VER;

    InitLib();
    X310_Init();
    resetALL();
    resetALL();
        iFlag=0;
        iAction=0;
        Puts("\n\rDemo program for 7188EX + X310\n\r");
        Print("\n\rVERSIONE %d\n\r",version);
    while(iAction!=99)
    {
        iAction=0;
        quit=0;
        Puts("\n\r");
        Puts("1) Select Dosimeter and Readout the value\n\r");
        Puts("2) Output a voltage step\n\r");
        Puts("3) Read both Analog Output\n\r");
        Puts("4) Scan TEST\n\r");
        Puts("5) Read EEPROM settings\n\r");
        Puts("6) Read EEPROMs block 7\n\r");
        Puts("\n\r");
        Puts("99) Quits demo program\n\r\n\r");
        Puts("Choose an option and press [Enter]:");
```

```
        Scanf("%d", &iAction);
        Puts("\n\r");
        display7(boardNum,1);
        Show5DigitLed(3,17); // writes "-"
        display7(dosiNum,4);

        switch(iAction)
        {
            case 1: //Promt user for DosiNum and reads it
                Puts("Insert BOARD No [1-15]: ");
                Scanf("%d", &boardNum);
                Puts("Insert DOSIMETER No [0-10]: ");
                Scanf("%d", &dosiNum);
                Puts("Insert Current No [0-32mA]: ");
                Scanf("%f", &current);
                Puts("Insert Max Voltage No [0-125V]: ");
                Scanf("%f", &voltage);
                Puts("Insert Time No [0-10]: ");
                Scanf("%d", &time);

                current=current/3.2; //conversion form [0-32mA] to [0-10V]
                voltage=voltage/12.5; //conversion form [0-125V] to [0-10V]

                X310_AnalogOut(0, 0);

                GetTime(&beginTIME.h,&beginTIME.m,&beginTIME.s);
                TimerOpen(); //starts counting time
                selectDosi(boardNum,dosiNum); //selects the right dosimeter
                setHV(current,voltage,1); //increase slowly the current to Imax
                readDosimeter(time); //read Voltage & Current Monitors.
                setHV(current,voltage,0); //decrease slowly the current to 0
                selectDosi(0,15); //select a free channel and deselect all dosimeters.
                endTIME.total=TimerReadValue(); //finishes counting time
                GetTime(&endTIME.h,&endTIME.m,&endTIME.s);
                TimerClose();
                Print("BEGIN TIME: %02d.%02d:%02d\n",beginTIME.h,beginTIME.m,beginTIME.s);
                Print("END TIME: %02d.%02d:%02d\n",endTIME.h,endTIME.m,endTIME.s);
                Print("Total measurement time: %ld\n",endTIME.total);
            break;
```

```
                case 2: //voltage step
            Puts("Insert Channel No [0-1]: ");
            Scanf("%f", &channel);
                        Puts("Insert Voltage No [0-10]: ");
            Scanf("%f", &voltage);
            Puts("Insert Time No [0-10]: ");
            Scanf("%d", &time);
            stepMaker(channel, voltage, time);
                break;

                case 3: //repeated reads
                        Puts("Insert time between reads: ");
                        Scanf("%d", &time);
                        analogRead(time);
                break;

                case 4: //test of MUX algorithm
                        scanTest();
                break;

                case 5: //EEPROM functions
        iValue=X310_Init();
        Print("Word di controllo: %d\r\n",iValue);
        for(i=0;i<2;i++)
        {
          Print("EEPROM A/D Gain  (Ch%d) ==> [%+8.6f]\r\n",i,
               Read_AD_CalibrationGain(i));
          Print("EEPROM A/D Offset (Ch%d) ==> [%+8.6f]\r\n",i,
               Read_AD_CalibrationOffset(i));
        }

        Puts("Change EEPROM A/D settings?: ");
        Scanf("%d", &choice);
        if(choice)
        {
            XEE_WriteEnable();
            while(i<17)
            {
                Print("EEPROM A/D Gain (Ch%d) ==> [%+8.6f]\r\n",i/4,
                     Read_AD_CalibrationGain(i/4));
```

```
                        Puts("Change with: ");
                        Scanf("%f", temp);
                        XEE_MultiWrite(7, i, 4, (char*)&temp);
                        Print("EEPROM A/D Offset (Ch%d) ==> [%+8.6f]\r\n",i/4,Read_AD_CalibrationGain(i/4));
                        Puts("Change with: ");
                        Scanf("%f", temp);
                        XEE_MultiWrite(7, i+4, 4, (char*)&temp);
                        i=i+8;
                    }
                    XEE_WriteProtect();
                }
            break;

                case 6:
            i=0;
            while(i<1000)
            {
                XEE_MultiRead(7, i, 4, (char*)&temp);
                Print("%+8.6f\n",temp);
               i=i+4;
            }
                        break;

            case 99:
                                if(iFlag==1)
                                {
                                        StopUserTimerFun();
                                }
            default:
                quit=1;
            break;
        }
        if(!quit)
        {
            Puts("Press any key to continue...\n\r");
            Getch();
        }
    }
}
```

```
void resetALL(void)
{
        X310_AnalogOut(0, 0);
        selectDosi(0,15);
        Disable5DigitLed();
    Init5DigitLed();
}

int invertNumber(int dosi)
{
        char mask=0x01;
        int dosi_inv=0;
        int i;
        for(i=3;i>=0;i--)
        {
                dosi_inv=dosi_inv+((dosi & mask)<<i); //masking of the first bit of dosiID
                dosi=dosi>>1;
        }
        return dosi_inv;
}

void display7(int num,int position)
{
        int unit=0;
        int decade=0;
        while(num>9)
        {
                num=num-10;
                decade++;
        }
   unit=num;
   Show5DigitLed(position,decade);
   Show5DigitLed(position+1,unit);
}

void clockSR(int channel)
{
    X310_Write_One_DO(channel , 0);
    Delay(1);
    X310_Write_One_DO(channel , 1);
```

```
}


void stepMaker(int channel,float voltage,int time)
{
        X310_AnalogOut(channel, voltage);
    Delay(time);
    X310_AnalogOut(channel, 0);
    Print("AO[%d] %5.3fV for %dmsec OK.\n\r", channel, voltage, time);
}

void selectDosi(int board, int dosi)
{
        char dosiID[3]; // (24 bit) 19 bit used
        int i=0,j=0;
        int bitCounter=0;
        int oneBit;
        char mask=0x01;

    for(j=0;j<3;j++) dosiID[j]=0;

    dosiID[0]= 0; //- IC3  - boards[9 to 15]
    dosiID[1]= 0; //- IC2 - boards[1 to 8]
    dosiID[2]= 0; //- LEDs - dosimeters
    //Inversion of dosiID sequence
    if(board<=8)
    {
                dosiID[0]= 0x00;
                dosiID[1]= (1<<(8-board));
    }
    if(board>8)
    {
                dosiID[0]= (1<<(16-board));
                dosiID[1]= 0x00;
    }
    dosiID[2]= invertNumber(dosi);

   //SENDING OUT SERIALLY THE ID
    for(j=0;j<3;j++)
    {
                for(i=0;i<8;i++)
```

```
        {
            if(j==2 && i<2) i=4;
            oneBit=(int)(dosiID[j] & mask); //masking of the first bit of dosiID
            dosiID[j]= dosiID[j] >> 1; //shift of dosiID for future processing
            oneBit=1-oneBit; //INVERT THE LOGIC
            X310_Write_One_DO(0 , oneBit); //bit is sent to shift register
            bitCounter++;
            clockSR(1); //clock cycle on pin DO1 to shift
                }

    }
    clockSR(2);  //write DATA VALID
}

void setHV(float current,float voltage,int status )
{
    float step;

    X310_AnalogOut(0, 0);
    X310_AnalogOut(1, voltage); //set maximum voltage
    step=current/10;

    while(step<current) //increase output current by steps of current/10
        {
                if(status) X310_AnalogOut(0, step);
        else  X310_AnalogOut(0, current-step);
                Delay(_SLOPE/10);
        step++;
    }
    if(status) X310_AnalogOut(0, current);
    else
    {
        X310_AnalogOut(0, 0);
        X310_AnalogOut(1, 0);
        while(X310_AnalogIn(1)>0.1) Delay_1(1); //waits until the output is 0
    }
}

void readDosimeter(unsigned int time)
{
        float vRead=0;
```

```
        float iRead=0;
        int i;
        float averageV=0.0;
        float averageI=0.0;
        float dosiR=0;

        if(time<50) time = 50;

    Print("TRUE VALUES\n");
        for(i=0;i<_READNUM;i++)
        {
                Delay(time/_READNUM);
                vRead = X310_AnalogIn(1)*12.5;
            iRead = X310_AnalogIn(0)*1.6;
            Print("Voltage n.%d - %6.4f ",i,vRead);
            Print("Current n.%d - %6.4f\n",i,iRead);
            averageV=averageV+vRead;
            averageI=averageI+iRead;
    }
    X310_AnalogOut(0, 0);

    averageV=(averageV/_READNUM); //Gives out the average value
    averageI=(averageI/_READNUM); //The resistor is already 10 Ohm
    dosiR=averageV/averageI;

    Print("Voltage -> %6.4f V\n ",averageV);
    Print("Current -> %6.4f mA\n ",averageI);
    Print("Dosimeter resistance -> %6.2f KOhm\n",dosiR);
}

void analogRead(int time)
{
     float current,voltage;
     do
     {
         current= X310_AnalogIn(0)*1.6;
         voltage= (X310_AnalogIn(1))*12.5;
         Print("Current %6.3f mA - Voltage %6.3f V - Resistance %6.2fKOhm\n",
             current,voltage,voltage/current);
         Delay(time);
     } while((Kbhit()==0));
```

```
        Print("Ultima misura -> R= %6.2f KOhm\n",voltage/current);
}

void scanTest()
{

            int i,board,time;
        Puts("Insert BOARD No [1-15]: ");
        Scanf("%d", &board);
        Puts("Insert Time No [0-10]: ");
        Scanf("%d", &time);

        for (i=0;i<12;i++)
            {
            selectDosi(board,i);
            Delay(time);
        }
}
```

Acknowledgements

This bachelor final project would not have been possible without the help and the suggestions of people who contributed and extended their valuable assistance in the preparation and completion of this work.

I must first express my gratitude towards my tutor, **Professor Lodovico Ratti**, who was always very patient and creative in finding the right solutions, very clear in correcting my mistakes and in particular in finding the right approach and way out when it looked like there was no one at all. He was transferring me his trust, and without that trust, I would have not been able even to start this adventure, and without his support, I would have never been able to, as it said in italian, *"unire l'utile al dilettevole"*, to combine business with pleasure, and what is more important, to achieve a concrete result.

I've really enjoyed working for this project, also because I was constantly under the supervision of two wonderful and delightful persons, **Maurice Glaser** and **Federico Ravotti**, to whom I owe hundreds of offered hours of help and... coffees. I really enjoyed the excellent professional environment that everyday occurred in the Irradiation Laboratory. It gave me the real view of how is it to work with only clever and knowledgeable people. They gave me an example of the Electronic Engineer I hope to become some day.

I would also like to show my gratitude to two (ex) PhD. students, **Alessia Manazza** and **Emanuele Quartieri**, who patiently taught me how to use half of the laboratory equipment and were concerned to make me always smile.

Thanks to my colleagues of the bachelor course, who have supported me before, during, and I hope after this project. In particular I would like to thank **Laura**, who always stayed in touch with me, updating me while I was working in Geneva.

I would like to thank my friends **Giulia** and **Francesco**, for keeping my moral high with all the phone calls, messages and jokes.

I'm more than thankful to **Elisa** for everything she has done for me, and I ask her to forgive me for every time i made her waiting me in the sunflowers fields outside the CERN entrance. Her support and understanding was always bringing me up and made me believe that sooner or later... I will also speak french.

Last but not least, I want to thank my sister **Natalia** and her husband **Vincenzo** for the encouragement I felt, for letting me play with my little **Annina** and in particular for making me feel at home, and thanks to my parents **Alexander** and **Tatiana**, who always believe in me (no matter what I'm doing), for their help, love and moral support.

Bibliography

[1] Maurice Glaser and Michael Moll. *CERN irradiation facilities (East Hall)*. Available online `https://irradiation.web.cern.ch/irradiation/`.

[2] Maurice Glaser Federico Ravotti and Michael Moll. *Radiation Sensor Working Group*. Available online `https://irradiation.web.cern.ch/irradiation/radmon/solid_state_group/solid_state.htm`.

[3] F Ravotti and B Camanzi. "Studio della risposta di dosimetri a stato solido da utilizzare per la misura dei campi di radiazione nell'esperimento CMS a LHC". PhD thesis. Geneva: Turin, 2002.

[4] M. Glaser, M. Moll, and F. Ravotti. "Radiation Sensors for the LHC Experiments". In: 2006.

[5] A. Holmes-Siedle, F. Ravotti, and M. Glaser. "The Dosimetric Performance of RADFETs in Radiation Test Beams". In: *Radiation Effects Data Workshop, 2007 IEEE*. 2007, pp. 42–57.

[6] F. Ravotti et al. "Status of the Radiation Monitors for the LHC Experiments". In: 2007.

[7] A. Holmes-Siedle and L. Adams. "Handbook of Radiation Effects". In: Oxford University Press, 2002. Chap. 4, pp. 132 –141.

[8] Federico Ravotti et al. "Development and Characterisation of Radiation Monitoring Sensors for the High Energy Physics Experiments of the CERN LHC Accelerator." PhD thesis. Geneva: Montpellier U., 2006.

[9] F Ravotti et al. "Radiation Monitoring in Mixed Environments at CERN: from the IRRAD6 Facility to the LHC Experiments." In: *IEEE Trans. Nucl. Sci.* 54 (2006), pp. 1170–1177.

[10] UltraVolt Inc. *Application Notes on HV Power Supplies*. Available online http://www.ultravolt.com/product/application-notes.

[11] UltraVolt Inc. *Ultravolt unveils new enhanced interface options for its high-voltage power supplies*. Available online http://www.ultravolt.com/news/261.

[12] Circly. Available online http://www.cirly.com.

[13] ICPDAS. *I-7188EX/I-7188EXD Embedded Ethernet / Internet Controller*. Available online http://www.icpdas.com/products/PAC/i-7188_7186/i-7188ex.htm.

[14] Arduino Due. Available online http://arduino.cc/en/Main/ArduinoBoardDue.