# Remote plasma experiments on e-lab

**João Pedro Simões Loureiro**

Thesis to obtain the Master of Science Degree in

## Engineering Physics

### Examination Committee

Chairperson: Professor Doctor Luís Filipe Moreira Mendes
Supervisor: Professor Doctor Horácio João Matos Fernandes
Co-Supervisor: Doctor Johan Willem Oosterbeek
Members of the Committee: Doctor Carlos Alberto Nogueira Garcia da Silva
Professor Doctor Bernardo Brotas de Carvalho

**October 2013**

## ABSTRACT

In the work that led to this thesis, two plasma experiments have been developed and added to the remote laboratory e-lab. This was done using the e-lab framework which is based on ReC (Remote experienced Control) and programmed in Java.

The two apparatus developed were designed to address basic processes regarding plasma physics: "When is a plasma formed?" - the Paschen Curve - and "What are the plasma most basic characteristics?" - the Langmuir Probe.

In the Paschen Curve the conditions at which the transition from gas to plasma occurs are studied by investigating the gas breakdown DC voltage between two parallel electrode plates at a certain distance as a function of pressure.

The Langmuir Probe is related to the electrical characteristic measurement of a thin tungsten filament placed inside the plasma. From the curve analysis one can extract the electron temperature and density of the plasma.

Plasma experiments are difficult to maintain and normally only a few European schools offer access to them and usually only to their own students. By including these apparatus on the e-lab network they become accessible to anyone in the internet and opens the possibly to include them in future massive open online courses (MOOCs).

**Keywords:** Remotely Controlled Experiments, Plasma measurements, Langmuir Probe, Paschen Curve

# RESUMO

Durante o trabalho que levou à redacção desta tese foram desenvolvidas e adicionadas duas experiências de plasmas ao laboratório remoto e-lab. Para tal utilizou-se a infraestrutura pré-existente do e-lab programada em Java e baseada no ReC (Remote experienced Control).

As duas experiências desenvolvidas foram concebidas para retratar dois assuntos básicos da física de plasmas: "Como é um plasma formado?" - Curva de Paschen - e "Quais são as propriedades básicas de um plasma?" - Sonda de Langmuir.

Na experiencia da Curva de Paschen estudam-se as condições para as quais ocorre a transição de um gás para um plasma averiguando a relação existente entre a tensão de disrupção, a pressão desse gás e a distância dos dois eléctrodos paralelos. Na experiência da Sonda de Langmuir determina-se a caraterística elétrica dum filamento de tungsténio colocado no plasma. A partir da análise desta curva pode-se retirar informação relativa à temperatura e densidade do plasma.

As experiências com plasmas são de difícil manutenção e normalmente apenas um pequeno número de escolas Europeias faculta o acesso a estas experiências com fins didáticos. As poucas que o fazem apenas permitem o acesso de utilização aos seus alunos. Com a adição destas experiências na rede do e-lab elas ficam disponíveis para qualquer pessoa com acesso à internet e abre a possibilidade para que sejam usadas no futuro em cursos online massivos (MOOCs).

**Palavras-chave:** Experiências Controladas Remotamente, Medições de Plasma, Sonda de Langmuir Probe, Curva de Paschen

# ACKNOWLEDGEMENTS

This work, supported by the European Communities and "Instituto Superior Técnico", has been carried out within the Contract of Association between EURATOM and IST. Financial support was also received from "Fundação para a Ciência e Tecnologia" in the frame of the Contract of Associated Laboratory and from Erasmus Mundus scholarship. Both experiments were supported by FUSENET, and the Paschen Curve was also supported by the "Technische Universiteit Eindhoven" as part of Plasma Lab.

Regardless of having only one author, writing this thesis was by no means a solo accomplishment. If were not for the contributions of many persons, I would've never been successful at this task. I feel very fortunate to have to been surrounded by all those who offered support when I needed it.

First of all, I would like to thank my supervisor, Horácio Fernandes, for his guidance throughout this work. I'm very thankful for the continuous support you gave me and for the patience you had during all this time.

I would also like to express my gratitude Hans Oosterbeek who challenged me to work with Plasma Lab and who was my close supervisor during my stay in the Netherlands. Here, I also extend my thanks to Herman Jong without whom the Lab couldn't work.

Further, I would like to praise the team at IPFN who helped me raising my enthusiasm for the field of plasma diagnostics, who advised me during all the stages of development of the Langmuir Probe setup. In particular the support of Carlos Silva on the plasma probe theory and João Fortunato for assisting in the development of the experiment in Lisbon while I was in Eindhoven.

I also can't forget all the support José Pereira gave me ( and multiple international calls) which helped me with all things Java.

I would also like to acknowledge my friends whose main contribution was to keep my mind off and maintain my sanity. Namely i would like to thank: Alexandre Vaz for being always there for me and always ready to help. Isabel Almeida e João Penedo, the best study companions, whose company during the hardest study nights made most of my grades possible. And most recently, Ana Leitão and Lúcia Fonseca who made my time in Eindhoven most enjoyable and most memorable.

Finally I would like to extend my highest appreciation to my family, specially my parents. Without their unconditional love and support this would've never been possible.

# CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

## LIST OF ABBREVIATIONS

- AC - Alternating Current
- ADC - Analog to Digital Converter
- DAC - Digital to Analog Converter
- DC - Direct Current
- GUI - Graphical User Interface
- HV - High Voltage
- IO - Input Output
- IST - Instituto Superior Técnico
- JNA - Java Native Access
- OC - Output Compare
- PID - Proportional integral differential
- PWM - Pulse Width Modulation
- RF - Radio Frequency
- RCL - Remote controlled laboratory
- ReC - Remote experienced Control
- TU/e - Technical University Eindhoven
- UART - Universal asynchronous receiver/transmitter
- USB - Universal Serial Bus
- XML - eXtensible Markup Language

## LIST OF VARIABLES

### VARIABLES:

- $k$ - Boltzmann constant
- $V_b$ - breakdown voltage
- $j_{sat}^{+}$ - current density
- $n$ - density of the plasma
- $E_{breakdown}$ - electrical breakdown field
- $E$ - electrical field
- $e$ - electron charge
- $V_f$ - floating potential
- $\gamma$ - flow constant
- $d$ - gap distance
- $i_{sat}^{+}$ - ion saturation current
- $c_s$ - ion sound speed
- $m$ - mass
- $\Gamma$ - particle flux density
- $\epsilon_0$ permittivity of vacuum
- $V_p$ - plasma potential
- $p$ - pressure
- $V_s$ - probe voltage
- $v_{se}$ - speed at the sheath edge
- $A_s$ - surface of the probe
- $T$ - temperature
- $\alpha$ - Townsend parameter

### INDEXES:

- $e$ - electron
- $i$ - ion
- $l$ - left
- $lw$ - left wall
- $r$ - right
- $rw$ - right wall
- $se$ - sheath edge
- $sf$ - sheath floating

# 1

# INTRODUCTION

Plasmas are the fourth state of matter, by opposition to solid, liquid and gaseous states. This implies that they have different proprieties and specific characteristics, namely temperature, density and conductivity. They are usually regarded as an ionized gas. This suggests that, as with gases, pressure plays a role and indeed the fact is that to be able to create a plasma they usually have to be contained in a vessel and be held at very low pressures compared with atmospheric pressure. This introduces some of the difficulties to their study, namely economical and technical.

Plasmas and its related technologies are immense, ranging from illumination to micro-electronics and to various medical applications. They have been around for the last few decades, and, because of that, it is virtually impossible not to run across them in our day-to-day life. Not only they reshape the way we live our lives, they also present one reliable solution to the energy crisis. It is therefore expectable the increasing importance of the study of this subject.

To address the aforementioned problems it is proposed the creation of remote plasma experiments. Taking this approach will allow one single experiment to be available to a larger group of people, and also, since it's remote, to diminish the complexity in the control due to that fact that a remote user interface is used instead of manual control. Because of it's already established performance as a remote laboratory, e-lab [1] was chosen by Fusenet [2] to host the experiments.

## 1.1 REMOTE CONTROLLED LABORATORIES

A remote controlled laboratory (RCL), as the name suggests, provides remote control of real scientific experiments over the Internet. Such a platform allows its user to:

1. select an experiment that the user (client) wants to run;
2. configure the experiment (parameters that the user can change, e.g.: pressure);
3. visualize the experiment in real time (over a webcam and/or real-time plotted results)
4. collect and visualize results from the experiment.

"e-lab" [1] is a RCL located at Instituto Superior Técnico (IST) of Lisbon University. The bridge between the user and the experiment is made by a software framework where the user can configure, run, see in real-time, collect and visualize the data from said experiment. All of its content is free and can be accessed by anyone who has a computer with internet connection.

Using the e-lab platform, the user will be able to set some parameters for the experiment, perform the experiment, observe it in action and collect the data of interest. This is done without the trouble of obtaining access to a real laboratory. And since all this is done in a few minutes it's possible, for example, to have an entire class of students performing this experiment with different parameters in the same day.

The e-lab platform is based on ReC (Remote experienced Control) and programmed in Java. This is done in a generic way facilitating the integration of new experiments and the maintenance of new ones. This platform runs in a cloud computing infrastructure managed by a cluster of computers over a local network with distributed services.

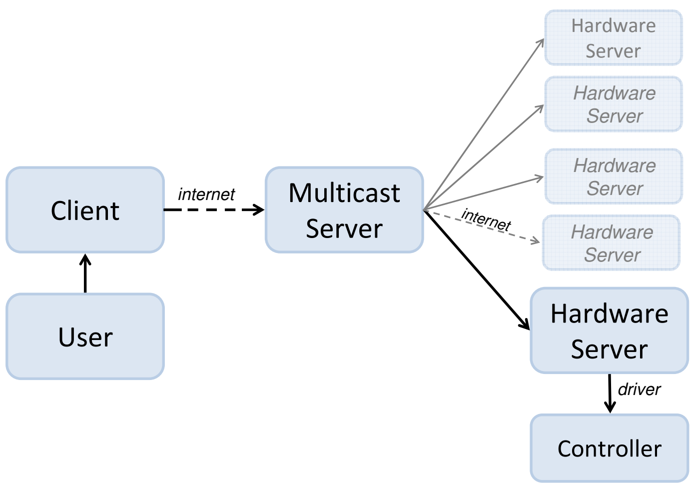



Figure 1.1.: Simplified block diagram of the e-lab network in its main components

Figure 1.1 shows a simplified diagram of the e-lab system. A user connects to e-lab using the client, that runs locally on the user's computer, which, via the internet connection, talks to the central "multicast" server. This server relays the communication from the client to a hardware server, which is the software that makes the interface between e-lab and the controlling hardware. This can be in the same computer cluster as the multicast.

## 1.2 REMOTE EXPERIMENTS

A remotely controlled experiment is an apparatus totally automatised which can be managed remotely by using a computer network. This allows a user to control it at distance. In the case of e-lab adding an experiment means not only adding the physical apparatus and its controlling hardware but also the pieces of software that define the user interface and the hardware server.

e-lab already has several experiments and with this work two more were added, they are the "Langmuir Probe" and the "Paschen Curve" described further on.

### 1.2.1 *Langmuir Probe*

As stated, plasmas have different characteristics from other states of matter and in order to measure them many diagnostic tools have been developed. This experiment allows a user to measure some of these characteristics using an electrical probe, also known as the Langmuir Probe [3]. This is one of the most simple methods of plasma diagnostics. It consists of a thin filament made of conductive material, placed inside the plasma, which either attracts or repels the electrons in the plasma according to its biasing. Measuring the probe I-V characteristic, that is, the relationship between the biasing voltage and the respective current going trough it, one can extrapolate the electron temperature and density of the plasma.

### 1.2.2 *Paschen Curve*

Being a different state of matter, there must be conditions at which a gaseous material progresses into plasma state. This experiment explores the conditions at which this transition occurs for a gas by studying the breakdown voltage, that is the voltage at which the plasma is formed, between parallel plates as a function of pressure and the distance between plates. Since the plasma is a good conductor, as opposed to the gas, it is fairly easy to verify this transition by watching the current drawn from power supply.

## 1.3 RESEARCH QUESTION AND DOCUMENT OUTLINE

This thesis has the objective to show "How can we teach plasma physics concepts using remote experiments?".

So, in order to answer this question we will, in the second chapter, make some considerations regarding the usage of remote labs as a teaching tool. The requirements for the development and construction of this tool are presented. Then as a practical implementation of the ideas discussed two remote experiments are added to e-lab: The Langmuir Probe and the Paschen Curve. Both of these experiments introduce basic concepts that are essential to the understanding of plasma physics.

The first one, the Langmuir Probe, is developed from scratch and it's the first plasma experiment in e-lab. It was developed following the usual e-lab approach, using the standard control hardware with a dspicnode. This dictates the structure of the third chapter of this thesis in which the following topics are discussed:

- The selection and assembly of the experiment hardware, ready for manual control are discussed, followed by the development of hardware for automatic and remote control of the experiment;
- The development of software for control of the hardware is discussed. This is done within the framework of the e-lab ReC generic driver;
- The development of the Client, including the Graphical User Interface (GUI), which was done in Java, is presented;

The second experiment, the Paschen Curve, has already been built and automatized, however with a completely different controller hardware from the usual e-lab experiments, a PicoGiant. This board was developed by the Experiment Automation Group from the Technical University of Eindhoven. In this case, advantage is taken of the e-lab versatility to interface with new devices. The fourth chapter relates to the addition of this experiment and divides into the following topics:

- The experimental apparatus is described, ready for both manual and automated control;
- A Java interface to the PicoGiant is developed and used to communicate with the ReC server;
- The development of the GUI for the experiment, done in Java, is presented;

The fifth chapter is meant to be used as guide to anyone that wants to explore the experiments. For each experiment there is a scientific introduction to each experiment, followed by the experimental protocol in which instructions are given on how to use the experiment GUI, there are also exemples of possible data sets given by the experiment and a discussion of results.

Finally, in the sixth chapter, there is a conclusion on the work done in this experiment followed by suggestions on future work to enhance. A discussion on the different technologies used in each experiment is also presented here.

# 2

# REMOTE CONTROLLED LABORATORIES

## 2.1 TYPICAL LANGMUIR PROBE LESSON

Langmuir probes are widely regarded as the most common low temperature plasma diagnostic. This is due to their very simple construction and geometry. The application of these probes is very wide from industrial plasmas to fusion plasmas where they are used to study plasma edge. Therefore they are an unavoidable subject in any plasma physics course. The typical course on this subject will consist of a presentation on the demonstration of the electrical characteristic of the Langmuir probe resembling that of appendix A.1. The students are then taken to the lab where they can validate the theory from the class.

While the experimental apparatus doesn't differ much between institutions there is one big difference in how the plasma is generated which in turn affects the ionization fraction. The most simple and inexpensive apparatus use AC discharge, like that which is used in light bulbs, while other more expensive solutions can be used like a high voltage DC discharge, electron beam gun or even microwave injection which lead respectively to better ionized plasmas. Regardless of the plasma generations most devices consist of a controlled low-pressure chamber with a probe where:

- the pressure can be controlled with help of a gas injection and vacuum pump and monitored using a pressure gauge,
- the biasing voltage of the probe can be imposed using a voltage source while the current and voltage at the probe can be measured with a multimeter or an oscilloscope,

The experimental protocol resembles the one in the Protocol Chapter. After the plasma is generated a voltage sweep is performed on the Langmuir probe thus allowing the determination of its electrical characteristic.

## 2.2 CLASSICAL PASCHEN'S LAW LESSON

A classical lesson on the Paschen's Law usually consists of two parts. First there is usually a scientific introduction on the subject, generally including a derivation of the Paschen's Law similar to that seen in appendix A.2. Afterwards an experimental activity is performed in a lab. The experimental apparatus used isn't very different from that used originally by Paschen, Townsend or the manual operation of device described in the Chapter 4. It consists of a controlled pressure chamber with parallel electrode plates where:

- the pressure can be controlled with help of a gas injection and vacuum pump,
- the distance between electrodes can be altered via a manipulator,
- the voltage across the electrodes can be imposed using a DC voltage source,
- the pressure, the distance and the voltage can be monitored using a pressure gauge, a ruler and a voltmeter, respectively.

The protocol followed is also very similar to that which is suggested in the Protocol Chapter. Depending on the apparatus the users make either a pressure or a distance sweep in which at regular intervals the breakdown voltage is checked by raising the voltage across the electrodes until the state transition occurs.

## 2.3 RCL CONTEXT

Experimental laboratory courses play a determining part in scientific education and the evolution of computer aided instrumentation has led to a significant change on how those courses are lectured. Where previously one could only reinforce the information of a course by means of a hands-on laboratory, currently that is no longer the case due to the existence of remote laboratories, which dislocate the user from the location of the experimental apparatus, or due to simulated laboratories, that completely disconnect the user from the real world. Consequently, there is a ongoing long-running discussion regarding the effectiveness of these new types of laboratories. Throughout literature, some conclusions and evaluations have been made [4].

All the experiments performed are mediated by the elements of measurement and, as one can easily notice by entering a lab, most of these measurement devices are already electrical in nature and require a computer or any other form of digital interface to output their results. In those cases, users usually have to only press a combination of buttons to receive their data and the experiment is automated in such a way that nothing else has to be done. In such a case the interactivity of the experiment is highly compromised and it is arguable if the local presence of the student is necessary or not, clearly showing that there is no difference between such an experiment done in a lab and done in a remote laboratory. This format of a computer mediated hands-on lab has been shown to be useful [5] [6]. These studies suggest that technology used, hands-on or computer mediated doesn't matter as long it suits the theme being discussed and how appropriately it suits each situation.

The effectiveness of labwork appears to be correlated to how these experiments are related to their real-world counterpart [7] [8]. Therefore a conclusion can be extracted that hands-on experiments drive the capacity to not only perform experiments but also to design them while the remote laboratories are better to enhance the focus on the underlying concept driving the experiment.

One final remark that should be made: attention should be drawn to the fact that the above discussion is taken the possibility that students, or even teaching institutions, have unlimited access to both types of laboratories. That is not true, often is the case that schools might not have the budget to acquire their own versions of the experimental apparatus required to perform these activities, while on the other hand, access to computers and an internet connection has been made increasingly easy, thus driving the development of remote laboratories.

## 2.4 PRESENT WORLDWIDE IMPLEMENTATIONS

Nowadays a quick search on the internet will show that there are plenty of remote laboratories. However when taken these searches more seriously one will quickly notice that most websites offering such services are actually not remote laboratories but repositories of links to such laboratories [9], which raises attention to the problem that although there are plenty of these labs they often have just one or two experiments thus driving the hard work of setting up a remote laboratory infrastructure quite useless.

There are however 3 main contestants that make full use of these such an infrastructure, adding more that just a couple of experiments to their collection. This is the iLab [10] from the Massachusetts Institute of Technology in the United States of America, the UNED Labs [11] from the Universidad Nacional de Educación a Distancia in Spain and finally the e-lab [1] from IST in Portugal.

All these remote laboratories provides a complementary website (usually a wiki) which briefly explains the experiment's concepts and the experimental protocol that should be followed in order to do the activity. The extent of detail and quality of these texts varies from experiment to experiment and the format varies from lab to lab.

Although the iLab claims to have countless experiments available most of them are closed and only available to students from certain courses in the university, rendering the availability of these experiments to a very small group and questioning the necessity for a remote laboratory. Something similar happens with the second one, which has a much smaller selection of remote experiments which are also only available to the students of this university, thus having only open to all public a virtual counterpart of these experiments. Finally there is e-lab, which has exclusively remote experiments and gives access to these experiments to everyone, regardless of their knowledge or affiliation to the hosting university.

Because e-lab is the only remote laboratory with a variety of experiments already available with open access it was decided to use this infrastructure for the new experiments in this work.

It is also worth to mention that, regarding the subject of plasma physics there simple are no remotely controlled experiments reported in the literature which are in any way similar to those that are being added in this work, or any that allow students to test fundamental plasma physics. However there is no novelty in automatizing a plasma physics experiments, since most of them are already mostly mediated by computers. This is specially true for the big fusion experiments which due to complexity must rely on automation and remote control.

## 2.5 E-LAB SUPPORTING NETWORK

Underlying to the e-lab's infrastructure there is a set of servers, linked together into a network, supporting the e-lab framework. All the experiments are connected to this network and all the services available on-line run in these servers.

This network mainly consists on five types of servers: the Production server, the Pre-production server, the Multicast servers, the Glassfish application server and the Media Center.

The Production server handles all the applications released for public access while the Pre-production supports all development and testing activities. This allows the verification of all newly developed features or bug fixes before they're released into production. The structure of these two servers is similar allowing for an easy migration from Pre-production to Production. These are the servers where the Hardware Server process runs and these are the computers to which the experiments are physically connected.

The Multicast servers deal with the connection between the Client software and the Hardware Servers. It is also in this servers that the experiments are grouped into laboratories. Therefore to each laboratory there is one Multicast server.

The Glassfish application server which is where the e-lab rec.web java application is deployed and thus made available to the public. This is the server that allows a user to access and download the Client software.

Finally there's the e-lab Media Center which is a server that deals with the video streaming for all experiments.

Each of the components communicate using the CORBA/IIOP protocol making use of the computers network to reach each other.

Physically these servers are constituted by 2 cluster systems composed of blade PCs. These are IAtom with 1G RAM and 12V multi-rail power supply. Each of these sets shares a hard disk drive from where they boot and where the running software is stored. Neither of these clusters have access to the exterior of their network. They support the experimental apparatus by running the Hardware server software which drives the experiments that are connected to them via a RS232 port.

Using a private intranet allows the intercommunication of the cluster's computers without exposure to the outside networks. IST grants gigabit access to the Géant infrastructure, which is the pan-European research and education network that interconnects Europe's National Research and Education Networks (NRENs).

# 3

# LANGMUIR PROBE

## 3.1 HARDWARE

### 3.1.1 *Experimental Apparatus*

The experimental apparatus can be summarized into the diagram of figure 3.1. We have a central chamber, in which all the events will occur and a array of actuators to establish the parameters inside the chamber. Outside there is a dspicnode board using its peripherals to control the actuators.

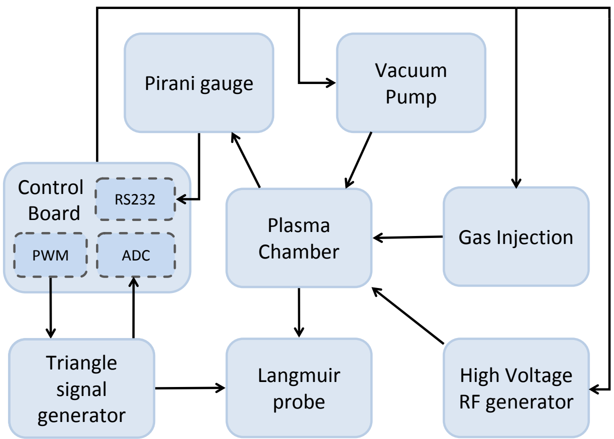



Figure 3.1.: Diagram of the Langmuir Probe experiment

Connected to the main chamber we have both the Gas Injection System and a Vacuum Pump to achieve the necessary pressure conditions. The Gas Injection System has a flow valve but the pump has only a switch valve. At the bottom of the chamber there is a modified ion gauge head from Edwards Vacuum that has a set of filaments (seen in figure 3.3). Two of them, the thicker ones, are connected to a high voltage (HV) radio frequency (RF) generator that creates the discharge to generate the plasma inside the chamber. The other two filaments, the thinner one and the coil, are used as Langmuir Probe

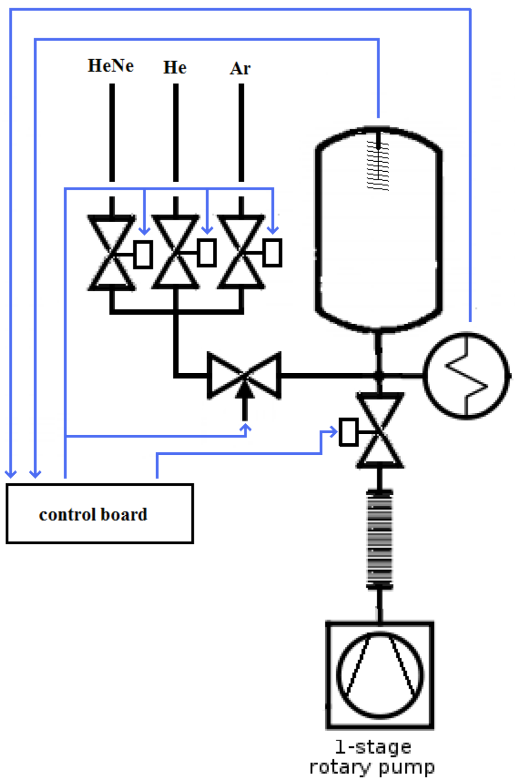



Figure 3.2.: Mechanical System Diagram for the Langmuir Probe experiment

and as ground, respectively. To measure the pressure inside the chamber a Pirani Gauge is used and it is constantly monitored by the board. This controls the vacuum quality and the gas. To function, the Langmuir probe needs a bias voltage sweeping which will be generated by one of the board's pulse width modulated signal generator (PWM) and additional circuitry.

To measure the the probe characteristic we use the board ADC and the correct electronics to provide offset adjustments and signal filtering.

### 3.1.2 *Hardware Description*

The mechanical system for the experiment can be summarized in the diagram of the figure 3.2. The various gases are provided and selected by the array of switch valves and then controlled by the flow valve. The connection from the high pressure bottles to the switch valves is made via Swagelok fittings and a 6 mm diameter tube.

The switching valves have a common exit which is directly connected to the flow valve, the 248A from MKS, also with Swagelok fittings. The exit of the flow valve is then connected to another 6 mm diameter tube which feeds the main chamber via an Swagelok to NW10 adapter connected to a cross-head.

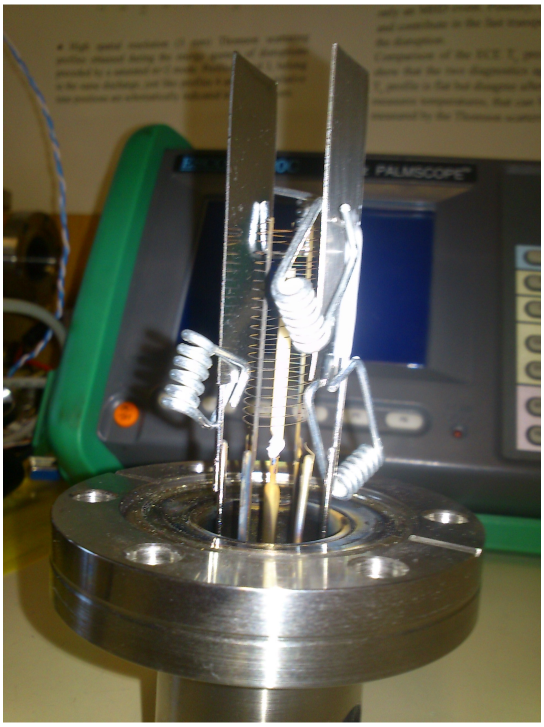


Figure 3.3.: Modification of the Ion Head to turn it into a Langmuir Probe (the molybdenum plaques where later trimmed to the specified size)

Also connected to the cross head is a NW25 switch valve that connects to a bellow and the rotary vacuum pump and a Pirani gauge, the PPT100 from Pfeiffer, with a NW10 head. All the coupling using the NW10 or NW25 system are made with the appropriate Neoprene o-ring and clamp. Finally, there is a NW25 to CF40 that connects to the main chamber.

The main chamber consists of a pirex cylinder with CF40 endings. On the side opposite to cross-head is Ion Gauge Head from Edwards. Originally it was a Ion Gauge Head D02998140, 2 3/4 inch ConFlat, Kovar 3/4 inch port dual tungsten filaments for IGC Ion Gauge Controller $5 \times 10^{-3}$ to $2 \times 10^{-11}$ mbar. This head has been modified so that it can function as a Langmuir probe, as pictured in figure 3.3.

The central filament has been covered with alumina tube leaving only a $10mm$ tip exposed to the plasma. The two outer filaments are connected to an "of the shelf" RF generator which ionizes the gas and generates the plasma. This is a simple $3.5W$ generator which outputs a $50kHz$ AC signal with $1kV$ amplitude. Since the electrical field was not uniform the plasma would have a tendency to escape the region where the probe is, thus, to improve stability and homogeneity of the electrical field two molybdenum plaques have been attached, one to each outer filament. These plaques are $20mm \times 50mm$ and are attached to the filaments using low vapour pressure epoxy. Figure 3.3 was taken while performing the modifications so the clamps that where used to hold the plaques can still be seen in the picture but they were later removed. On figure 3.4 one can see the final assembly of the head during operation of the experiment.

Figure 3.4.: Glow Discharge in the Chamber

### 3.1.3 *Control and Data Acquisition Hardware*

#### *Control Board*

Most experiments in e-lab, including this one, use a dspicnode controller board. This board is discussed further in the next section whereas here only the dedicated hardware is considered. For more information on the board itself the next section or appendix C with the schematic of the board should be consulted.

Connected to this board is an auxiliary board that expands its capabilities. This board's schematics is in appendix D while the following sections explain each functionality of this board.

#### *Relay Control*

To operate the relays that control the power supply to the devices such as the vacuum pump or the switch valves, a simple pull down transistor configuration is used. Due to the power involved we use Darlingtons already in use in the standard control board. Connected to the base is a IO pin, from the dsPIC, in output mode. To avoid damaging transients a capacitor is placed in parallel with the solenoid from the relay. Also to protect the solenoid a diode is also placed in parallel.

*Flow Valve Control*

To operate the solenoid that controls the opening of flow valve we use a similar topology to the one before, a pull down transistor. Due to the power involved we use Darlingtons already in use in the standard control board. Instead of a simple output we use the Output Compare (OC) peripheral to generate a simple pulse width modulated signal (PWM). Choosing the correct capacitor to filter the 50kHz PWM signal, one of about 220$\mu F$ for a solenoid with 100$\Omega$ impedance, we can control the voltage applied to the solenoid by varying the pulse duty cycle.

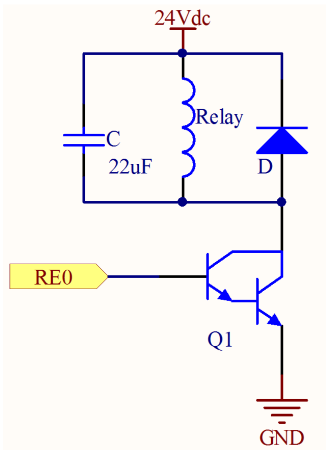



Figure 3.5.: Relay Control Circuit Schematic

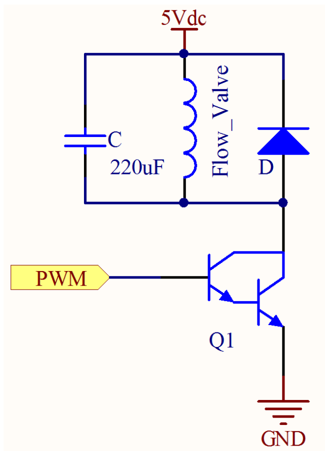



Figure 3.6.: Flow Control Circuit Schematic

*Signal Generator*

We wanted to be able to make a generator capable of generating a signal with any shape, amplitude and frequency to sweep the probe. To achieve that we use a Sallen−Key low-pass filter which is a 2nd Order Non-Inverting filter, with a cutting frequency at 2kHz. This filter averages the signal coming from the PWM with 50kHz, giving a constant output signal proportional to the duty cycle. Changing the duty cycle with the right speed and by the correct amount will output a signal with any desired shape under the limitations of this circuit. In the next chapter the code to generate a triangle signal will be discussed.

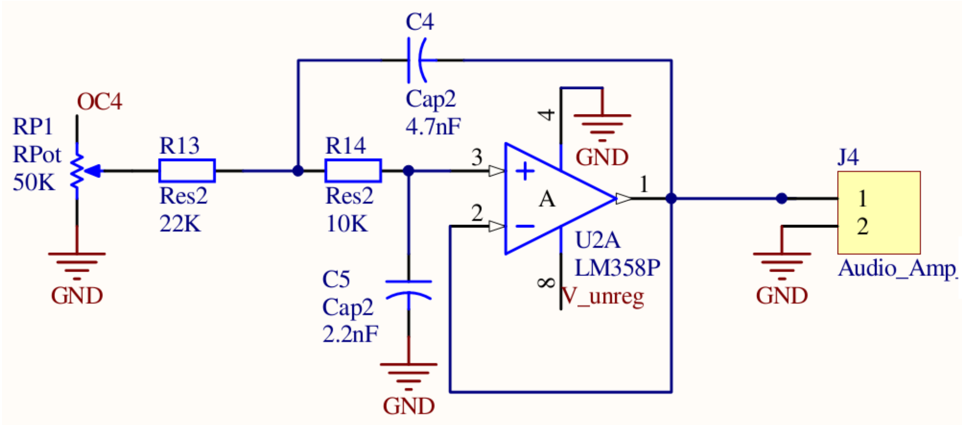



Figure 3.7.: Signal Generator Circuit Schematic

The output of this stage is a small amplitude signal. Since we need a large signal we connect the output to a two stage amplifier, each stage providing about 20dB of gain. The first stage is a audio mono amplifier, the K8066 kit from Velleman, chosen to provide a high fanout enough to drive the second stage, which is a 24V to 220V transformer. The output will go into the probe and a voltage divider.

*Voltage Measurement*

In this experiment we use the 10-bit ADC from the PIC to measure the probe characteristic. Since it uses a 5V regulated supply as reference we must make sure the place where we measure has the biggest excursion possible, to provide the biggest precision possible, giving the ADC used.

To do that in the voltage measurement we simply place a voltage divider at the transformer output. This was done taking into account the fact that the level of noise is much smaller than the signal. Another source was introduced in a voltage adder configuration to offset the signal by 2.5V which corresponds to half the reference signal to the ADC. This allows the measurement of both positive and negative bias of the probe. The following circuit was designed to do this.

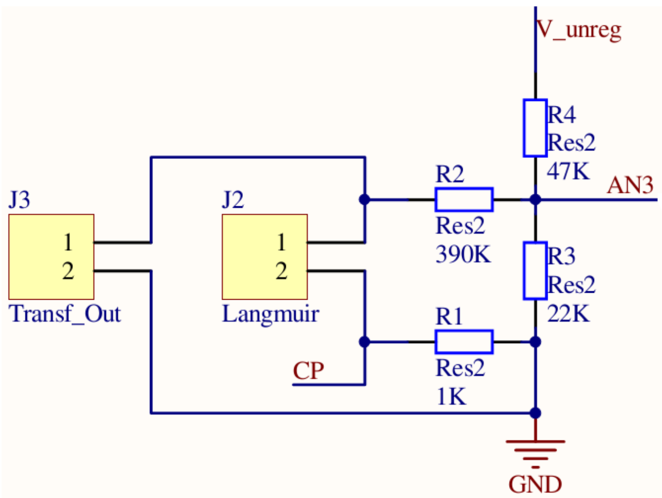



Figure 3.8.: Voltage Measurement Circuit Schematic

*Current Measurement*

As said before we use the ADC from the PIC to measure the probe characteristic. The collected current from the probe is a very small signal, some hundreds of micro-amperes in the best cases. On top of this signal we have the 50kHz discharge used to create the plasma as well as the 50Hz from the surroundings making this a noisy line. Therefore to be able to measure it special conditioning has to be taken into account. The main source for the 50Hz noise is the pickup made by the cables that connect the experiment to the electronic instrumentation. Since this frequency is in the range of the signal we want to measure it was opted not to introduce a filter as it would mean a further reduction of the allowed band for the sweeping signal. Instead to shield from this a coaxial cable was used for return signal with the outside of the cable connected to the ground. Furthermore a capacitor of 100nF is placed from the probe to the reference on the probe side to avoid capacitive couplings. Finally a 3rd order low pass filter is placed. This filter has a pole at 2kHz and has two stages: the first is a 2nd order Sallen-key with a small gain and the second is a first order filter with higher gain. The overall gain is of about 23.7dB. For the same reason as in the voltage measurement the filters also have an offset however in this case of only 0.4V. This introduces a necessary asymmetry to compensate for the exponential behaviour of the characteristic.

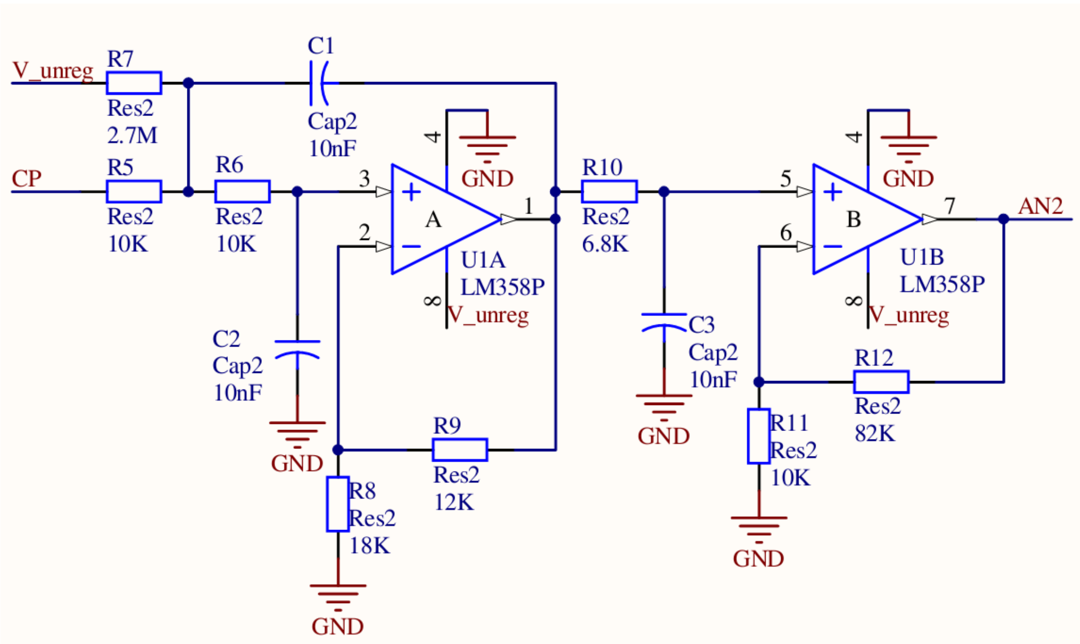



Figure 3.9.: Current Filter Circuit Schematic

## 3.2 CONTROL BOARD SOFTWARE

### 3.2.1 *dspicnode*

The dspicnode control and data acquisition board is a generic controller based on a dsPIC30F4011 for simple general purposes applications, allowing fast development times as it is C programmable. It is also suitable for real-time control or data acquisition. It's built in the Eurocard format, with $100mm \times 160mm$ in size featuring a DIN96 connector on one of the ends. The board includes:

- a 30 MIPS dsPIC30F4011 microprocessor [12] (which has a 24K instructions program memory, 2KB of ram memory and an 1KB EEPROM memory);
- multiple interfaces and communication ports (Optical, CAN, RS485, RS232, I2C, SPI and RJ11);
- IO pins (8 power pull-down, 6 analog inputs and 31 general purpose digital IO);
- a $7,3728MHZ$ crystal for microprocessor clock with multiple PLL factors ($4\times$, $8\times$ and $16\times$).

It needs a 9V DC power supply and features both a 5V DC regulated output as well as access to the unregulated input power.

### 3.2.2 *Sweeping Signal Generator*

As described before the hardware expects a 50kHz PWM with varying duty cycle. To achieve this we must first configure a Timer and then associate a Output Compare module. In this microprocessor there is only two timers capable of doing this: Timer 2 and 3. Taking into account the position of the output pin on the board Timer 3 was chosen to facilitate the design. In order to configure it one must use a code block similar to:

```
void open_timer3(){
  PR3 = 589;//Timer3 Period =>50kHz
  TMR3=0;//Timer3 Initial value
  T3CONbits.TCS = 0; //Use Internal Clock as Source
  T3CONbits.TCKPS = 0; //Prescaler 1:1
  T3CONbits.TGATE=0;// Gated time accumulation
  T3CONbits.TSIDL=1;  // Operation in Idle mode (1|0)
  T3CONbits.TON =0;//Disable timer
}
```

After this we have to configure a pin to work in Output Compare mode:

```
void PWM_init(){
  TRISDbits.TRISD2 = 0;//Set D2 as a output
  LATDbits.LATD2 = 0;//Set D2 to low
  OC3RS=512;//Duty_Cycle
```

```
  OC3R=0;    //Delay
  OC3CONbits.OCM=0b110;//Use Output Compare as Simple PWM mode
  OC3CONbits.OCTSEL=1;//Use Timer3 as timer for OC
}
```

Finally, we use it in the main routine to generate the signal, we take as inputs the maximum duty cycle (which translates into maximum amplitude), the period of the desired signal, the number of samples per period and the number of periods. In the main cycle the configuration parameters are acquired and processed. The ADC and the Timer 3 are activated and then we simply wait for the signal to finish. While we are waiting for this, there is an interrupt that is thrown at every period of the Timer 3. This interrupt makes the change in duty cycle that will generates the signal with the aid of the circuit. It also requests a new sample to the ADC and this makes it so that the sampling is synchronized with the signal and provides a steady sample rate.

```
  max_duty = param_1; //Should be <500
  t_sinal = param_2;  //Tsinal real = input2*PR3*34ns
  n_samp = param_3;   //No of points per period
  n_period = param_4; //No of periods
  timeout_nwait = t_sinal/(2*n_samp); // Must be >0
  duty_inc = (2*max_duty)/n_samp; // Must be >0

  if(timeout_nwait == 0) {timeout_nwait = 1;}
  if(duty_inc == 0) {duty_inc = 1;}

  printf("DAT\r");     //Begin of data stream

  timeout_count = 0;
  points_count = 0; //Clears acquired points counter
  OC3RS = 0;       //sets duty-cycle to 0 and
  T3CONbits.TON = 1;  //Enable Timer3
  ADCON1bits.ADON = 1;//Enable ADC
  IFS0bits.T3IF = 0;  //Clears Flag
  IFS0bits.ADIF = 0;  //Clears Flag

  while(points_count < n_samp * n_period){}

  printf("END\r");     //End of data stream

  T3CONbits.TON= 0; //Disable Timer3
  ADCON1bits.ADON = 0;//Disable ADC
  IFS0bits.T3IF = 0;  //Clears Flag
  T2CONbits.TON= 0; //Disable Timer2
```

```
void __attribute__((__interrupt__,auto_psv)) _T3Interrupt(void)
{
  IFS0bits.T3IF = 0;//resets the T3 interrupt flag
```

```
  timeout_count++;

  if(timeout_count>=timeout_nwait){
    timeout_count=0; //Resets timeout_count
    ADCON1bits.SAMP = 1; //Activates sampling

    OC3RS += duty_inc;
    if((OC3RS > max_duty) || (OC3RS < 10))
            { duty_inc = -duty_inc;}
    ClrWdt();
  }
}
```

### 3.2.3 *ADC Configuration and Data Acquisition*

The ADC has to be configured so that it samples two channels at once.

```
void Double_ADC_init(){
  ADCON1bits.ADON=0;
  ADPCFG = 0xFFF3;//all PORTB = Digital; AN2 e o AN3 analog input
  ADCSSL = 0x000C;//Samples AN2 and AN3
  ADCON1 = 0x0000;//Clear Everything
  ADCON2 = 0x0000;//Clear Everything
  ADCON3 = 0x0000;//Clear Everything

  ADCON1bits.FORM = 0b00;//Signed Integer
  ADCON1bits.SSRC = 0b010;//GP Timer3 compare ends
  ADCON1bits.SIMSAM = 1; //Samples CH0 and CH1 simultaneously
  ADCON1bits.ASAM = 1;//SAMP bit is auto set
  ADCON2bits.CHPS = 0b01; //Converts CH0 and CH1
  ADCON2bits.SMPI = 0b0000;//Interrupts at the completion of
          //conversion for each sample/convert sequence
  ADCON3bits.SAMC = 5; //Auto Sample
  ADCON3bits.ADCS = 5; //TAD=TCY(ADCS+1)/2
  ADCHSbits.CH0SA = 0b0010;//Channel 0 positive input is AN2
  ADCHSbits.CH123SA = 1;
}
```

As stated, the sampling and respective conversion are triggered by the Timer 3 timeout interrupt. When the conversion ends a new interrupt is thrown which can used to print the results to the terminal. Unfortunately `printf` is extremely slow function and couldn't be used. Furthermore the space available to store variable data in the program memory is too small and could only store about a hundred points which is not enough for a extensive data acquisition. Therefore a solution to publish the data as fast as possible had to be developed.

This was done using the `putchar` function which is a lot faster than the `printf`. Using this function a strategy was devised where the ADC conversion interrupt sets a variable used as a flag which in turn activates a routine that publishes the data in formatted string form. This format is required by the hardware server software that listens to the board communications. Also to save some speed a limit was set on the 10-bit words to 999. This saves time both in the conversion of integer to character as well as in the two extra `putchar` that would be wasted at the cost of the information regarding the high exponential region of the characteristic. If the format wasn't necessary the two 10-bit values could be stored in 3 8-bit words each represented each as character in the terminal. This is done in the following interrupt in conjunction with the code in the main function.

```
void __attribute__((__interrupt__,auto_psv)) _ADCInterrupt(void)
{
  IFS0bits.ADIF = 0; // clear interrupt
  data[0] = ADCBUF1;//passing the value
  data[1] = ADCBUF0;//to global variable
  myflag = 1;//Sets my flag
  ++points_count;//data points counter
}
```

```
if(myflag == 1){
  myflag=0;
  if (data[0]>1000){data[0] = 999;}
  if (data[1]>1000){data[1] = 999;}
  c[0] = data[0]/100;
  c[1] = (data[0]-100*c[0])/10;
  c[2] = data[0]%10;
  c[3] = data[1]/100;
  c[4] = (data[1]-100*c[3])/10;
  c[5] = data[1]%10;
  putchar('\r');
  putchar('0'+ (char)c[0]);
  putchar('0'+ (char)c[1]);
  putchar('0'+ (char)c[2]);
  putchar('\t');
  putchar('0'+ (char)c[3]);
  putchar('0'+ (char)c[4]);
  putchar('0'+ (char)c[5]);
  putchar('\t');
  putchar('0'); //Place holder for pressure channel
  putchar('\n');
}
```

### 3.2.4 *Communication with Pirani Gauge*

To measure the pressure inside the chamber we use a Pirani Gauge, the PPT100 from Pfeiffer, which talks via RS232 to the PIC using the protocol described by Pfeiffer in [13]. Originally the PIC board has the LTC485, a UART to RS485 converter, which is meant to be used with the UART1 however access to this pins is also made in the back connector. Therefore an extra Max3227 was used to have a second RS232 port. Therefore we simply need to configure the UART1 and use it for the serial communication with the PPT100.

To query the pressure to the device we must send a command to the probe, after that the answer must be decoded, the following functions do that:

```
float acquire_gauge_01_pressure () {
  unsigned char query[25];
  int j;
  static int error_count = 0;
  float retVal, pressure_gauge_01_F = 1;
  memcpy ( query , "0010074002=?106" , 15 );
  query[15] = 0x0D;
  hw_uart1_send_string ( query , 16 );
  j = hw_uart1_receive_string ( query , 25 );
  memcpy ( pressure_gauge_01_S , query + 10 , 6 );
  pressure_gauge_01_F=
  convert_pressure_string_2_float(pressure_gauge_01_S);
  return pressure_gauge_01_F;
}
```

```
float convert_pressure_string_2_float ( unsigned char * buf ) {
  float pressure;
  int exponent, mantissa;
  char mant[5], exp1[3];
  strncpy (mant, str1, 4);
  mant[4]= '\0';   /* null character manually added */
  strcpy (exp1, str1+4);
  mantissa = atoi(mant);
  exponent = atoi(exp1)-20;
  pressure = mantissa/1000. * powf(10, exponent);
  return pressure;
}
```

### 3.2.5 *Pressure Control with PID*

One of the strategies considered to achieve the pressure inside the chamber was use a proportional integral differential (PID) controller. As the name suggests the controller uses three therms in the transfer function: one proportional, one differential and one integral. This routine takes as input the desired pressure in micro-bars. To determine the time constant for the integral and derivative term we made a output pin of the PIC toggle after each cycle. Measuring that signal with an oscilloscope it was possible to determine the cycle length. We also wanted the PID to be adimensional. To do that we divided the result, which comes in mbars by the lowest value of pressure in the chamber, $2.4 \times 10^{-2}$ mbars and multiplied by the one of the lowest values of duty cycle that opens the valve, thus making the resulting value have units of duty cycle. To make sure there is no overflow in the output compare register a lower and a higher limit values were set. The tuning of the PID coefficients was such that there was no overshoot in most cases. The cycle stops once the value converges to the set-point and, because of this, the stop condition is programmed to make sure the set point has been reached.

```
kp= 0.35;
ki= 0.05;
kd= 0.01;

error =0, integral1 = 0, derivative =0, value =0;

T2CONbits.TON= 1; //Enable Timer2
PR2 = 589;       //Timer2 Period = PR2*34ns, used in PID

pressure = acquire_gauge_01_pressure();
error = setpoint - pressure;

while(estab < 20){
  pressure = acquire_gauge_01_pressure();
  error = setpoint - pressure;
  integral1 +=error * 0.148;
  derivative=(error-previous_error)/0.148;
  value = 10*(kp * error  +
    ki * integral1 + kd * derivative)/(2.4e-2);
  previous_error = error;
  if( value>500){value = 500;}
  else if( value<0){ value = 0;}
  OC4RS = (int) value;
  if(fabs(error/setpoint)<0.02){estab++;}
  ClrWdt();
}

//...Main routine...//

OC4RS= 0;        //Closes Valve
delay_ms(10);    //Waits 10ms
T2CONbits.TON= 0; //Disable Timer2
```

### 3.2.6 *Experimental Protocol Routine*

The final routine combines all those above so that the experiment can be performed. It takes as input the characteristics of the signal, the sample rate and the desired pressure in the chamber. After acknowledging these values, it opens the connection to the vacuum pump until the pressure is lowered to the requested value. After that runs the a routine to set the pressure inside the chamber and once it's stable, both the flow valve and the vacuum pump are closed. Then the signal generation process starts and the ADC is activated. After the desired data has been gathered the ADC and the timers are disabled and the respective flags are cleared. The data points are printed for the user in the terminal during the acquisition. However this data is in binary format so later it has to be decoded in the e-lab hardware server.

## 3.3 GRAPHICAL USER INTERFACE

The graphical user interface for this experiment was done following the e-lab default strategy described in [14]. There is a configuration window that allows the user to design the experiment by selecting the values for a variety of relevant parameters for the experiment.

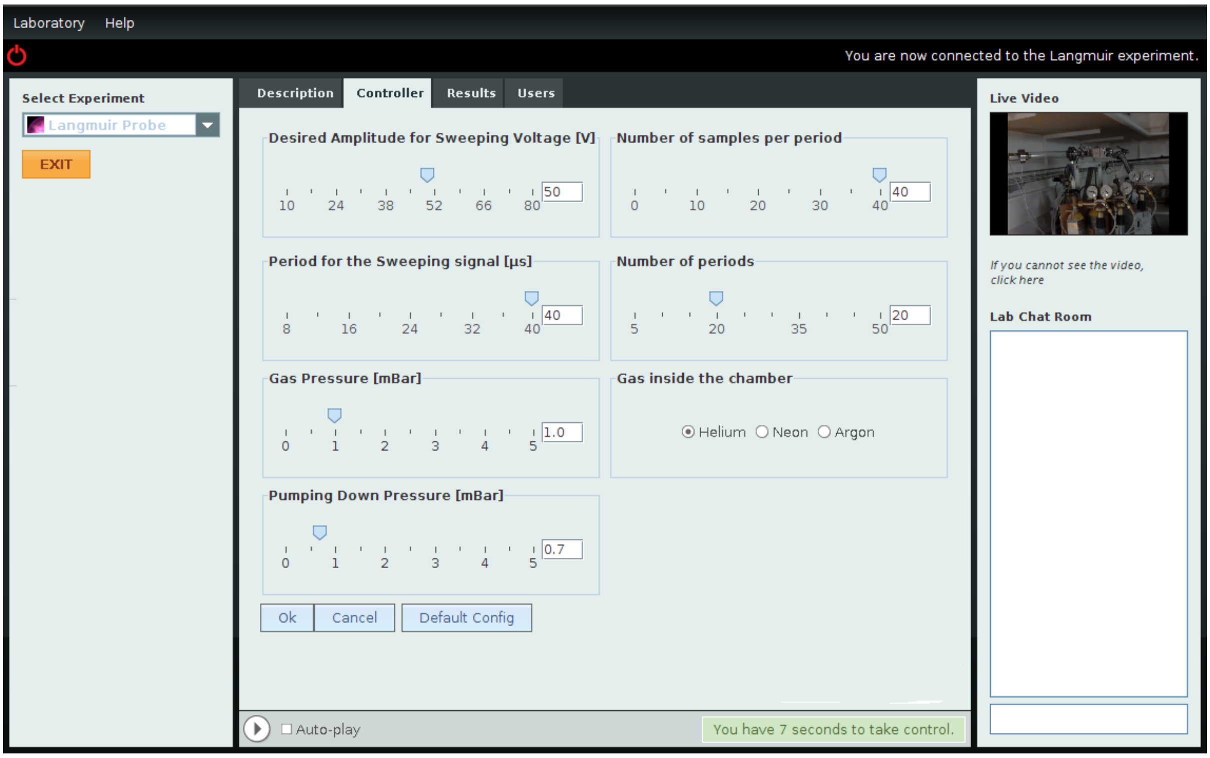



Figure 3.10.: e-lab user interface in configuration mode for the Langmuir Probe experiment. In this screen the user can configure all the parameters the experiment

In this particular case, seen in figure 3.10, the user can set:

1. the amplitude and the period of the plasma sweeping signal,
2. the number of periods for the sweeping signal,
3. the number of data points per period,
4. the pump down pressure and the working pressure,
5. the gas inside the vessel;

The maximum number of points a user can choose is limited by the period of the sweeping signal, if the period is too small there isn't enough time to acquire many data points. To allow the gas purity degree choice the pumping down of the pressure is done in the beginning of the experiment. Because of this it's also possible to mix two different gases. The working pressure is the one after the chamber is filled with gas and at which the experiment will run. For obvious reasons, the pressure at which the experiment is done has to be higher or equal than that until which it's pumped down to. Despite the fact that a default configuration is available, it is highly recommended that the user explores the experiment by himself.

Since both the gas injection and the vacuum pumping takes a long time, although the voltage and current are set to zero, the user can see the updates in pressure such that he can follow what is happening inside the chamber.

While the experiment is running, the client displays data from the experiment similar to what can be seen in figure 3.11. And in order for the user to be able to analyse it the data can also be displayed in the electrical characteristic form as seen in figure 3.12. For this experiment the user gets current and voltage at the Langmuir Probe and the pressure measured by the pressure gauge. After the pressure is set inside the chamber, and it becomes constant, it is artificially setted to zero because there is not enough time to get the value from the pressure gauge while generating the sweeping signal and performing the data acquisition. After a brief transitional period the signal goes into the normal triangular sweeping signal which is the relevant part for the data analysis. After the experiment is over the pressure is measured again in order to assure the user that it has remained constant throughout the whole experiment after which the experiment is over.

All this data can be seen in table format that can be saved and exported for further analysis. Finally, the user can also see the video feed of the webcam pointed at the experiment in the top right corner of the user interface which, if double clicked, can go to full-screen mode.

## 3.4 GENERIC DRIVER

The communication between the Hardware Server and the dspicnode board are made using a protocol designated ReC Generic Driver [15]. It's based on a state-machine which allows re-using the same driver without the need to develop a different one for each hardware. The implementation of this protocol consists of two parts. One that integrates within the hardware server, is written in Java and was


Figure 3.11.: e-lab user interface in result mode displaying the parameters' evolution along time.

developed by Linkare [16]. The other must be implemented on the hardware and, in the case of the dspicnode, is written in C and was developed within e-lab [15]. However since the communication protocol differs for each specific hardware they are defined for each experiment in a XML (eXtensible Markup Language) file. Thus the deployment of a new hardware only requires creating a new XML configuration file. In this file all the elements of the communication protocol are defined, namely:

- an identification string;
- the port to which the device is connected;
- the characteristics of the configuration parameters (number of parameters, their string format, the maximum and minimum values and a possible transfer function);
- the characteristics of the output channels (number of channels, their string format, the maximum and minimum values and a possible transfer function);
- timeouts and errors

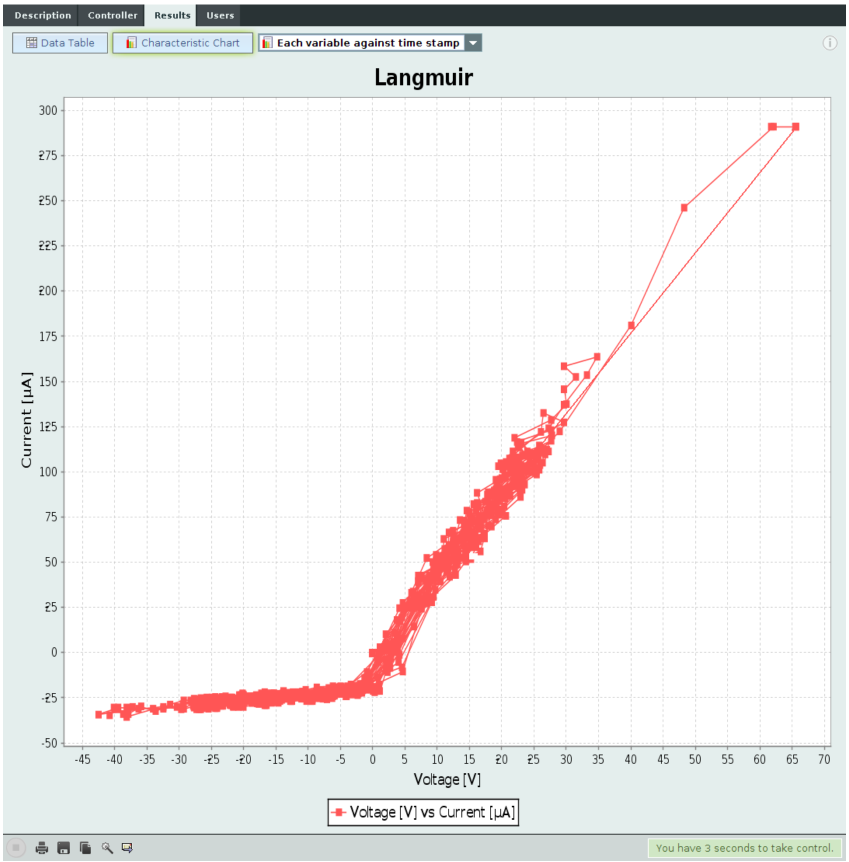



Figure 3.12.: e-lab user interface in result mode displaying electrical characteristic of the probe.

In this experiment there are 7 parameters coming from the GUI: (i) the sweeping signal amplitude and period (ii), (iii) the number of samples per period, (iv) the number of periods, (v) the pumping pressure and (vi) the operating pressure, and finally (vii) the gas type. The fact that these parameters are bounded with upper and lower limit allows validation of the configuration values. This adds one level of security to the experimental apparatus. Some of these parameters are simple integers which can be directly passed on to the board, however, other values, like the amplitude and period, must be converted into hardware dependent values, which make only sense for the dspicnode. Performing this conversion here makes the microprocessor run faster since it only has to work with integer values instead of floating values. In example we take the way amplitude is defined in this file:

```
<!-- amplitude -->
<parameter output="##0" input="##0" maxvalue="500" minvalue="50" order="0">
  <transfer_function type="output">
    <linear><param weight="6.429" center="14.286"/></linear>
  </transfer_function>
  <transfer_function type="input">
    <linear><param weight="6.429" center="14.286"/></linear>
  </transfer_function>
</parameter>
```

In a similar fashion there are also values that come from the hardware and that need to be transferred to the GUI, in the results screen: the bias voltage and current of the probe, as well as the pressure inside the vessel. As was discussed in the previous section the publication of this information is already critical in terms of speed. Therefore the possibility to give the data in raw integer format and delegate the conversion to analog values to the hardware server really makes a difference. Again the fact that the values are bounded gives us the opportunity to validate the data coming from the device. If a value is sent out of range might point to a possible issue with the apparatus. The XML syntax is identical to the block featured above apart from the fact that instead of <parameter> we use <channel>.

# 4

# PASCHEN CURVE

## 4.1 PASCHEN CURVE HARDWARE

### 4.1.1 *Experimental Apparatus*

The experimental apparatus (figure 4.1) consists of a main reaction chamber in which the variable distance electrodes are horizontally embedded and connected to a DC power supply. In the rear is a port connecting to the turbo-molecular vacuum pump whose opening can be controlled via a on-off valve.

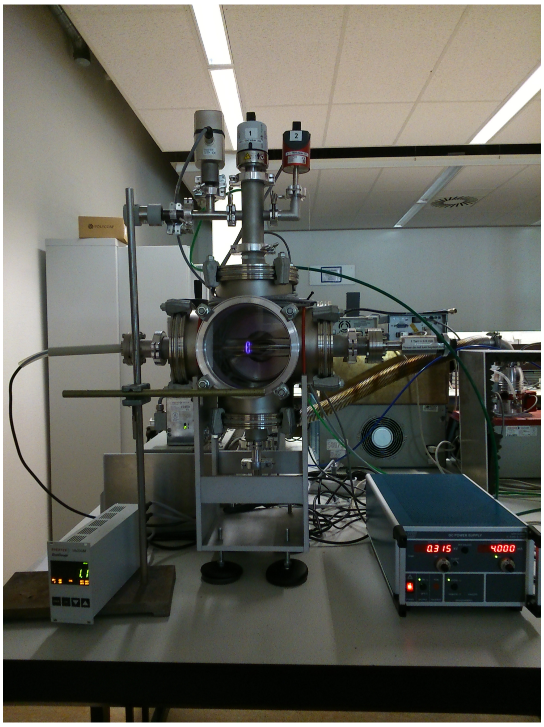


Figure 4.1.: Photograph of the experimental apparatus. The main chamber is to the left, while on the right is the power supply and the controller for the pressure gauge.

On the top a connection is made to the gas feeding line, in this case Argon was used. Its flow can be controlled via a electro-pneumatic needle valve. To measure the pressure inside the chamber a pressure gauges is used. There is a window in the front panel to allow a webcam to see inside the chamber.

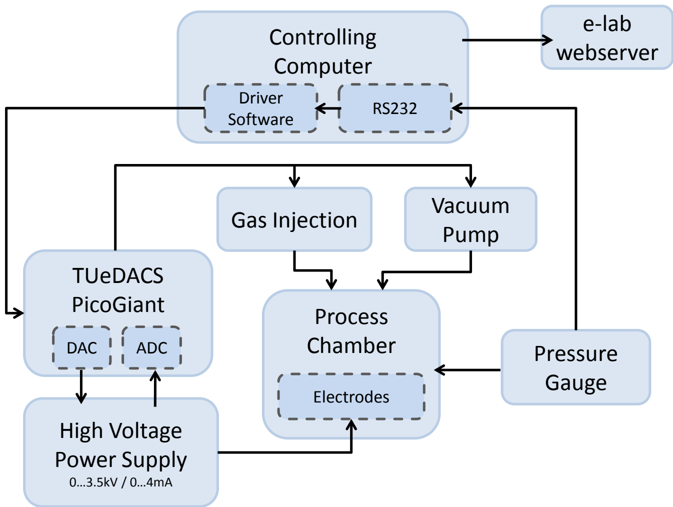



Figure 4.2.: Block diagram of the Paschen Curve experiment

Directly acting on the power supply and the flow valve is a TUeDACS PicoGiant [17]. This board has two 16-bit DACs that can go up to 15 Volt. One of the channels is used to control the 0 to 2000 volt power supply and the other is used to control the opening of the flow valve. There is also one digital channel connected to the pneumatic valve that gives access to the vacuum pump, thus allowing control over the pump. The voltage and current given by the power supply are acquired by the ADCs from the PicoGiant. The PicoGiant is then connected and controlled by a computer. This computer also interfaces directly with the pressure gage controller to get the data. All these connections are represented in the diagram from figure 4.2.

The hardware driver running on the computer will be constantly listening for commands from the e-lab server. If a certain configuration for the experiment passes the validation tests the driver will first close the line to the vacuum pump and then will attempt to raise the pressure by opening the valve for the selected amount. After 10 seconds it closes the valve and it begins to raise the voltage between the electrodes until the desired initial point is reached. The voltage has to checked because the voltage source has a high settling time and for high voltages it might take a couple hundred of milliseconds to reach the desired initial voltage. The voltage is increased until it reaches the final voltage. Every time the voltage is increased the voltage, current and pressure are measured and reported to the central server which in turn gives it to the client.

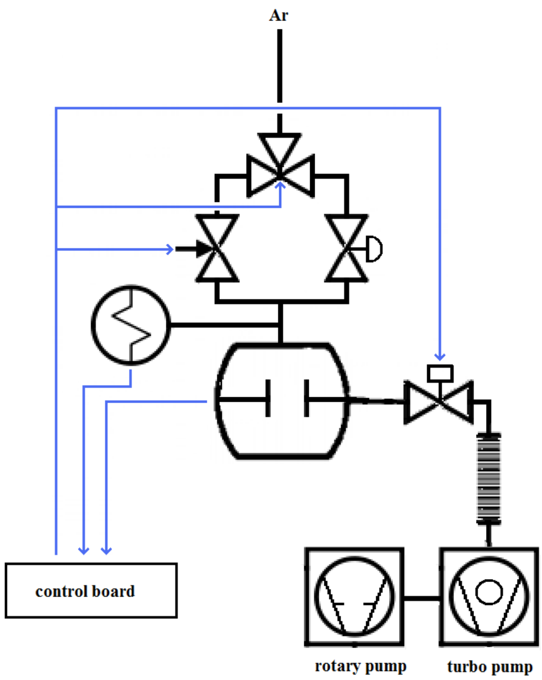



Figure 4.3.: Mechanical System Diagram for the Paschen Curve experiment

### 4.1.2 *Hardware Description*

The mechanical system for the experiment can be summarized in the diagram of the figure 4.3. This experiment has been designed to be operable in both manual and remote mode, and most of its components were purchased from Pfeiffer Vacuum.

The main reaction chamber is a DN60 ISO-KF 6-way cross head. The front side has a glass window that allows the user to see the experiment and detect by visual inspection when the breakdown is achieved.

The gas feed can be controlled both by a manual needle valve, the EVN 116, or by an electrical flow valve, the EVR 116. The input of these two valves is connected to a line alternator, the 439-032 from RS Componnents, so that only one of them is operational at the time. This selection is made using an interactive display on top of the controlling PicoGiant which allows the users to switch between manual and remote modes. The selection over one of the two inputs is the main difference between the modes of operation. The connection from the high pressure gas line in the building is made via Swagelok fittings and a 6 mm diameter tube. The electrical flow valve is controlled by the PicoGiant's 15V 16-bit DAC.

Both flow valves are connected to a NW10 cross-head in which also two pressure gauges, the PKR 251 and the CMR 361, are connected. Finally the cross head is also connected to an NW10 to DN60 ISO-KF adaptor which is bolted on top of the reaction chamber. The pressure gauges however, are not directly connected to the PicoGiant or the computer. Between them and the controlling board is a TPG 262 pressure gauge controller which reads both units and also serves as a local display for the chamber pressure.

The bottom of the reaction chamber has a check valve, the AVA 016 X, which checks for over pressure inside the vessel, acting as a last resort safety feature.

On the side panels are the electrodes. To the left side there is a static electrode which was machined from a blind plug for DN60. This is the one connected to the signal wire from the power supply. On the right, there is a movable electrode made using a mechanical mover from a Thermionics Northwood FLM-133-1 and again machined parts, including another blind plug for DN60. This electrode as well as all the vessel is connected to ground. As mentioned, connected to these electrodes is the output of a modified Fug DC Power Supply HCP 14-3500, which has been modified such that the PicoGiant can fully control its operational regime, it is, therefore, being used as an amplifier from 0-15V to 0-3500V. In order to force the breakdown to occur only between the electrodes, and not between the electrode and the wall, a glass tube has been placed around the electrodes increasing the isolation of the electrodes but still allowing for the gas to circulate.

On the back of the reaction chamber is the connection to the vacuum system, via a DN60 to DN40 adapter. The DN40 connects to a cut-off valve, the AVC 040 PA, which is controlled by the PicoGiant. In turn, this valve, connects to a manually actuated flow valve, the EVB 063 SA. This is a hand wheel valve which is regulated such that the evacuation flow doesn't damage the turbo-molecular pump. When in manual mode the cut-off valve is always open so this valve wheel valve allows control of the evacuation in the chamber. When in remote mode it is requested that the valve is always left in a determined position that allows pumping down with a limited flow that doesn't damage the turbo-molecular pump. This valve, in turn, connects to the Turbo-Drag Pumping Station, TSH071E DN63 which is the vacuum pump for this experiment. It consists of a combination of a turbo molecular pump and a regular rotary pump. The station also contains a controller for the pumps such that in cases of emergency, like overheating or high-pressure exposure the pump automatically switches down, avoiding permanent damage.

### 4.1.3 *Electronic circuits*

#### *Switching implementation*

The cut-off angle valve AVC 040 PA from Pfeiffer Vacuum doesn't have any version which can be directly controlled by any of the PicoGiant outputs. So a relay was placed between the digital IO pin 8 and the valve to switch the connection to the 220V AC which controls the opening of the valve.

*Power Supply modification*

The Fug DC Power Supply HCP 14-3500 offers an optional module that allows external analog programming. This module however assumes an analog input of 0-10V DC which is not the range of the PicoGiant's DAC. Therefore the application note provided by the manufacture company was followed and a voltage divider was installed to divide the 0-15V output from the DAC.

## 4.2 PICOGIANT CONTROL BOARD

### 4.2.1 *PicoGiant*

The PicoGiant [17] board is a multi-purpose low-end rapid prototyping data acquisition interface specifically designed for use in real-time closed-loop motion control systems. It is based on a field programmable gate array, the Cyclone II from Altera. It's built in the Eurocard format, with $100mm \times 100mm$ in size. It requires a 12V DC power supply. The board includes:

- 8 analog input channels with 12-bit resolution and programmable-gain amplifiers;
- 2 analog output channels with 16-bit resolution;
- 16-bit general purpose I/O port;
- 32-bit Output Compare;
- 32-bit Preset Scaler;
- Stepper Motor Controller;
- Programmable Clock;

### 4.2.2 *Application interface*

As stated previously, the PicoGiant is the main actuator of this experiment since it plays a role similar to the dsPic board on the Langmuir Probe experiment. However this device doesn't have a processor so, in order to control the experiment, a "master" device has to continuously tell the PicoGiant what to do. In most cases this device is a computer running a program capable of communicating with the board. Since e-lab runs on java the natural option was to create an interface between java and the already available native controlling libraries, the TUeDACS API. This implies that the computer running the hardware driver will talk to another program, designated the "Data Producer" which in turn tells the PicoGiant what to do using the board's USB connection. The interface between java and the native libraries is done using the Java Native Access (JNA), and in order to bridge the native libraries one must only create a java interface extending the Library interface defined in JNA. Afterwards all the functions provided in by our library should be defined. One can then use this class in the main program to have access to these functions.

```
import com.sun.jna.Library;
import com.sun.jna.Native;

public interface TUeTdIo extends Library {

  TUeTdIo INSTANCE = (TUeTdIo)
      Native.loadLibrary("tdIo", TUeTdIo.class);

  int tdOpen();
  int tdClose();
  ...
```

In order to control the PicoGiant (or any other TUeDACS device) two libraries are needed, the tdIo and the tdApi. The first one controls the communication with the device itself while the second provides access to the board peripherals. To properly control the board one must first open communications with it and then it should be configured. After this one can use all the functions that control the peripherals and in the end the communications with the board should be closed.

In java this is done by a block of code like:

```
  TUeTdIo TUeIO = TUeTdIo.INSTANCE;
  TUeTdApi TUeApi = TUeTdApi.INSTANCE;

  TUeIO.tdOpen();//Opens communication

  Memory config = new Memory(2);
  // allocating space for config

  config.setShort(0,(short) -1);
  // setting the -1 value, if pgConfig fails this stays -1;

  TUeApi.pgConfig(config); //Configures board for utilization

  /*
  controlling code....
  */

  TUeIO.tdClose();
```

### 4.2.3 *Voltage Ramp Generation*

The PigoGiant has two 16-bit digital-to-analog converters (DAC), so no complicated integration circuitry is needed to generate voltages. Instead one must configure the DAC and then update its output voltage to whenever is needed.

This is done by using the `pg_dacSetModeLv` and then `pg_dacPutValue` functions. Since we are interested in using the most simple mode of operation most of the advanced features of the DAC are deactivated. This is because these features make the DAC faster, which in our case is not needed since the reaction time of the controlled devices is much slower.

The DAC control is therefore achieved by the following lines (after the preamble of the previous section):

```
TUeApi.pg_dacSetModeLv((short) 0, (short) 1, false, false,
            false, (short) 0, (short) 0, false, false, 0);

TUeApi.pg_dacPutValue((short) 0, (short) 1, (short) dacValue);
```

### 4.2.4 *Flow Valve Control*

The flow valve has an opening controlled by and analog 0V to 15V signal so one of the DACs is used, in this case the channel 0, so this is a simple implementation of the previously shown code, in this case:

```
TUeApi.pg_dacSetModeLv((short) 0, (short) 0, false, false,
            false, (short) 0, (short) 0, false, false, 0);

TUeApi.pg_dacPutValue((short) 0, (short) 0, (short) dacValue);
```

### 4.2.5 *Cut-off Valve Control*

The cut-off valve AVG 040 PA requires only a signal of 220 Volt to open and since there is already available a relay system in the controlling board it is only required to have a digital signal to commute the relay.

The PigoGiant has one 16-bit input-output digital header (IO), so, in a similar fashion to how it was done with the DACs, one must configure the IO and then ask for what is needed, in this case, to commute the pin 8 in order to open or close the valve.

This is done by using the pg_dioInit to initiallize the module, followed by pg_dioSetOutputTriggerMode and pg_dioSetOutputBits to configure and finally the pg_dioOutputData function to determine the state of the output pins.

The IO control is therefore achieved by the following lines (after the preamble of the initial section):

```
TUeApi.pg_dioInit(zero, (short) 0x05); //Simple IO mode
TUeApi.pg_dioSetOutputTriggerMode(zero, false); // No Triggers
TUeApi.pg_dioSetOutputBits(zero , (short) 0xFFFF); //All pins are set to output
TUeApi.pg_dioOutputData(zero, (short) 0x100); //The Cut-off valve pin is setted to high (close
    )
```

Note that the valve works in inverted logic since it's of a "normally open" type (O is open and 1 is closed).

### 4.2.6 *Communication with Pressure Gauge*

To measure the pressure inside the chamber we use the Pressure Gauges described in the last chapter. However these require an extra unit to control them, the TPG 262 from Pfeiffer, which can communicate via RS232 to the computer. Since java has a library to communicate via RS232, the RXTXcomm needs to be integrated into our "Data Producer" with the correct instruction set to ask and decode the pressure.

To ask the pressure we must first send a command to the probe to configure it in data output mode, after that it can be queried to get pressure. The answer must be decoded, the following functions do that:

```
    public int configureGauge(){

        sendToOutStream (out,
        String.format("PR1%c%c", 13, 10).toCharArray());
        //Sends configuration

    try {Thread.sleep(100);}
    catch (InterruptedException e1) {}
        //waits for answer

    StringBuffer inbuffer =
        new StringBuffer(getFromInStream (in));
        inbuffer.trimToSize();
        //writes answer in a buffer

        if(String.format("%c%c%c", 6, 13, 10).contentEquals(
                  inbuffer.toString())){return 0;}
        else{return -1;}
      //if it is properly configured
```

```
      //the probe sends and acknowledgement

    }
```

```
    public double getValuefromGauge(){
    sendToOutStream (out, String.format("%c", 5).toCharArray());
        //Asks for pressure

    try {Thread.sleep(300);}
    catch (InterruptedException e1) {}
    //waits for answer

    StringBuffer inbuffer = new StringBuffer(getFromInStream (in));
        inbuffer.trimToSize();
        //writes answer in a buffer

        if(Double.valueOf(inbuffer.toString().split(",")[0])==0){
      return Double.valueOf(inbuffer.toString().split(",")[1]);
        }
        else{return -1;}
        //decodes the value and returns it

    }
```

### 4.2.7 *Pressure Control*

To control the pressure inside the vessel a combination of the two previous sections is used. In this case there can't be an permanent pumping of the vessel as it would introduce stress of the turbo molecular pump. Therefore a simpler but slower control is done. In this case the vessel has the connection to the vacuum pump always opened, so the pressure should always be low. On the beginning of the experiment the pressure is checked, if it is higher than the one that we want to perform the experiment in, it is pumped down. Then the connection to the gas feed is opened allowing for a slow increase in pressure which is constantly monitored until the desired pressure is achieved. After the pressure is set the gas feed is closed and the voltage ramp begins.

```
double pressure_inside=0;
pressure_inside = serialgauge.getValuefromGauge();

if(pressure_inside> press_set){
//If pressure is higher than the requested, pumps down
  while(pressure_inside> press_set){
    pressure_inside = serialgauge.getValuefromGauge();
    //Get value from gauge
    Thread.sleep(1000);
```

```
      //Wait 1s
    }
  }

  TUeApi.pg_dioOutputData(zero, (short) 0x100);
  //Close connection to vacuum pump

  if(pressure_inside<= press_set){
  //Opens valve until desired pressure is reached
    while(pressure_inside<= press_set){
      TUeApi.pg_dacPutValue(zero, zero, (short) 30000);
      //Open valve
      Thread.sleep(10);
      //Wait 10 ms
      pressure_inside = serialgauge.getValuefromGauge();
      //Get value from gauge
      i1++; //Increase Counter
      if(i1==10){//Every 100ms print data to user
        i1=0;//Reset Counter
        timestamp++;//Increase time stamp
        value = new PhysicsValue[NUM_CHANNELS];
        //Create values to send to client

        value[0] = new PhysicsValue(PhysicsValFactory.fromFloat(0), getAcquisitionHeader().
            getChannelsConfig(0).getSelectedScale().getDefaultErrorValue(), getAcquisitionHeader
            ().getChannelsConfig(0).getSelectedScale().getMultiplier());

        value[1] = new PhysicsValue(PhysicsValFactory.fromFloat(0), getAcquisitionHeader().
            getChannelsConfig(1).getSelectedScale().getDefaultErrorValue(), getAcquisitionHeader
            ().getChannelsConfig(1).getSelectedScale().getMultiplier());

        value[2] = new PhysicsValue(PhysicsValFactory.fromFloat((float) pressure_inside),
            getAcquisitionHeader().getChannelsConfig(2).getSelectedScale().getDefaultErrorValue
            (), getAcquisitionHeader().getChannelsConfig(2).getSelectedScale().getMultiplier());

        value[3] = new PhysicsValue(PhysicsValFactory.fromFloat((float) timestamp),
            getAcquisitionHeader().getChannelsConfig(3).getSelectedScale().getDefaultErrorValue
            (), getAcquisitionHeader().getChannelsConfig(3).getSelectedScale().getMultiplier());

        addDataRow(value);//Send data to client
      }
    }
    TUeApi.pg_dacPutValue(zero, zero, (short) 0);
    //Closes Valve
    Thread.sleep(100);
    //Wait 100ms for stability
  }
```

By the end of the experiment, or in case of emergency stop, the vessel is again completely evacuated. In case of normal operation the values of pressure going down are also printed to the user.

```
TUeApi.pg_dioOutputData(zero, (short) 0);//Open cut-off valve

double pressure_inside = serialgauge.getValuefromGauge();
while(pressure_inside>= 0.05){
//Wait until pressure is low enough (0.05mBar)

  pressure_inside = serialgauge.getValuefromGauge();
  //Check pressure
  Thread.sleep(1000); //Wait 1s
  timestamp++;//Increase time stamp
  value = new PhysicsValue[NUM_CHANNELS];
  //Create values to send to client

  value[0] = new PhysicsValue(PhysicsValFactory.fromFloat(0), getAcquisitionHeader().
      getChannelsConfig(0).getSelectedScale().getDefaultErrorValue(), getAcquisitionHeader().
      getChannelsConfig(0).getSelectedScale().getMultiplier());

  value[1] = new PhysicsValue(PhysicsValFactory.fromFloat(0), getAcquisitionHeader().
      getChannelsConfig(1).getSelectedScale().getDefaultErrorValue(), getAcquisitionHeader().
      getChannelsConfig(1).getSelectedScale().getMultiplier());

  value[2] = new PhysicsValue(PhysicsValFactory.fromFloat((float) pressure_inside),
      getAcquisitionHeader().getChannelsConfig(2).getSelectedScale().getDefaultErrorValue(),
      getAcquisitionHeader().getChannelsConfig(2).getSelectedScale().getMultiplier());

  value[3] = new PhysicsValue(PhysicsValFactory.fromFloat((float) timestamp),
      getAcquisitionHeader().getChannelsConfig(3).getSelectedScale().getDefaultErrorValue(),
      getAcquisitionHeader().getChannelsConfig(3).getSelectedScale().getMultiplier());

  addDataRow(value);//Send data to client
}
```

### 4.2.8 *Voltage & Current Measurement*

The PigoGiant has two 12-bit analog-to-digital converters (ADC), so, in a similar fashion to how it was done with the DACs, one must configure the ADC and then ask for its value whenever is needed.

This is done by using the `pg_adcSetModeLv` and then `pg_adcGetValue` functions. Again, since we are interested in using the most simple mode of operation most of the advanced features of the ADC are deactivated.

The ADC control is therefore achieved by the following lines (after the preamble of the initial section):

```
TUeApi.pg_adcSetModeLv((short) 0, false, false, false, false,
                (short) 0, (short) 0, false, false);

Memory adcValue = new Memory(2);  // allocating space
TUeApi.pg_adcGetValue((short) 0, (short) 0, adcValue);
```

In this case the return value in this function is given using a pointer which java doesn't have, so JNA offers a "Memory" object that can be used in place of it.

#### 4.2.9 *Experimental Protocol Routine*

The final routine combines all those above so that the experiment can be performed. It takes as input sweeping values for the voltage (initial, final and step values) and the desired pressure in the chamber. After taking these values, the chamber is pumped down to the lowest pressure. Then it sets the pressure inside the chamber and once it's stable, the flow valve is closed. Then the voltage sweep process starts and the ADCs are activated. Since this process is running in parallel to the Hardware Server the data is directly given to it.

### 4.3 GRAPHICAL USER INTERFACE

On the graphical user side this experiment also follows the typical structure of e-lab, as described in [14]. This allows the configuration of the parameters that define the experiment. For dynamic quantities the user can define their sweeping range, by setting the initial and final values as well as the step by which they are increased. In case of case o constant parameters their static value throughout the experiment is chosen.

In this particular case, the user can select the sweeping range for the voltage source and the opening of the gas valve, as seen in figure 4.4. The configuration of the voltage sweep is such that it can either have a large but coarse range or a smaller but finer range, thus allowing for a quick experiment or a precise measurement. In other words, there is a maximum limit to the number of points, such that the experiment doesn't take too long. Again the recommendation is emphasised that despite the fact that a default configuration is available a user should always explore the experiment by himself.

Unfortunately, although the distance between plates is also a parameter for the experiment, the setup doesn't support automated motion of the plates (as they can only be moved manually) so this distance is fixed. It is therefore requested to the users in the manual mode that the leave the apparatus with a 4.5mm electrode gap.

Since both the gas injection and the vacuum pumping takes a long time a similar approach was taken to the one on the Langmuir Probe that is a constant updates in pressure such that the user can follow what is happening inside the chamber.

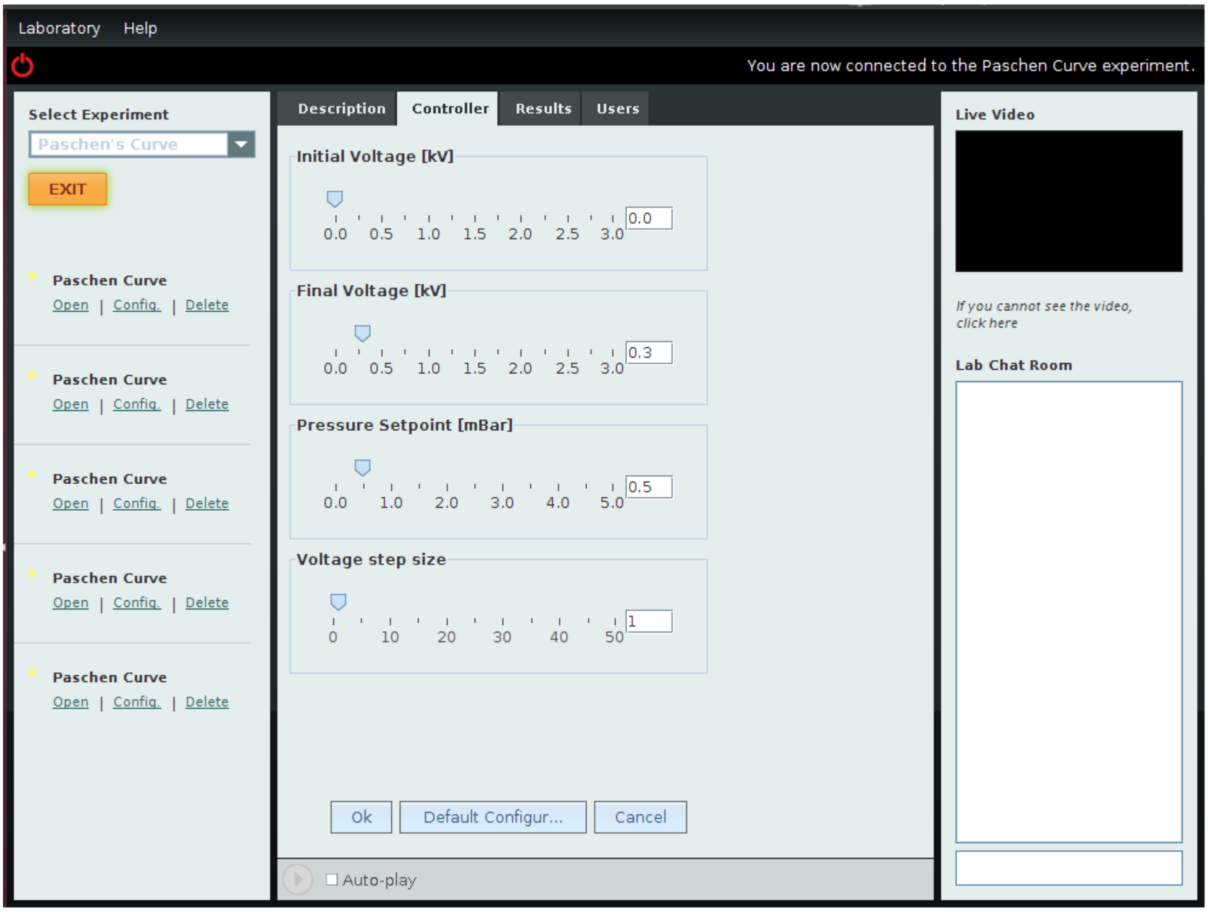



Figure 4.4.: e-lab user interface in configuration mode for the Paschen Curve experiment. In this screen the user can configure the voltage sweep and the pressure in the chamber for the experiment

While the experiment is running, the client displays data from the experiment in order for the user to be able to analyse it, similar to what can be seen in figure 4.5. For this experiment the user gets the current and voltage given by the power supply and the pressure measured by the pressure gauge. After the pressure reaches the setted value and it becomes constant the ramp in the voltage starts. If breakdown is reached a clear transition from zero to maximum current can be seen in the current graph. Also in the webcam the plasma can be seen.

Finally, after the voltage sweep, both current and voltage are set to zero and the gas is pumped from the chamber until it reaches a value lower than 0.5 $mBar$ at which point the experiment is over.

Again, as with all the experiments on e-lab, this data can be seen in form of a table which can be saved and exported for further analysis, or in form of graphs in which each quantity is plotted against time. The user can also see the video feed of the webcam in the top right corner of the user interface or in full-screen mode.

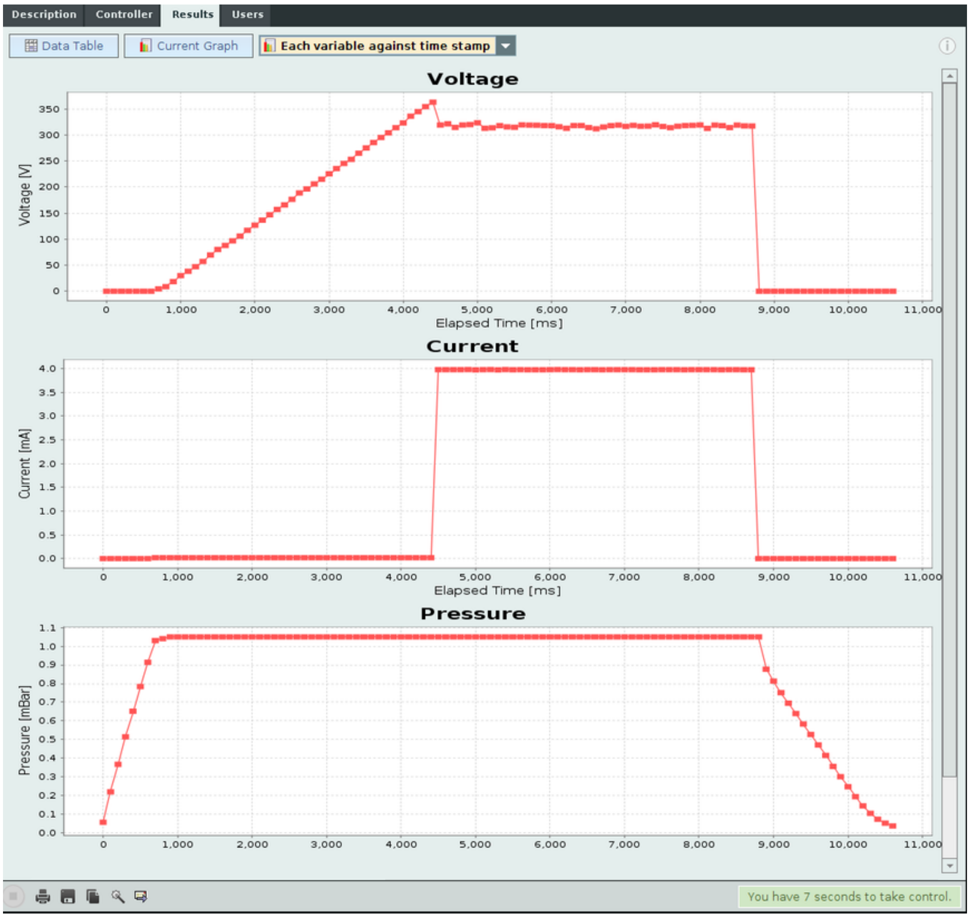



Figure 4.5.: e-lab user interface in result mode showing the graphics of the temporal evolution of the quantities measured from the experiment

## 4.4 CUSTOM DRIVER

The communication between the Hardware Server and the PicoGiant board could not be made using the ReC Generic Driver, as there is no way to incorporate its state machine into the device. Therefore in an attempt to maintain as much as possible the already available protocol a custom piece of code was written. This class, designated "Data Producer", was featured in the previous section. Because of that the implementation of the protocol still consists in two parts. One is written in Java that integrates within the hardware server and was developed by Linkare [16] and the other is the "Data Producer" which takes the place of the hardware driver. Although this class was well documented and done in such a way that any future contributor might use it to add a new experiment it does not done feature a full compatability with the XML configuration files. So not only this class talks with the PicoGiant but also talks with another independent hardware, the TPG 262 pressure gauge controller which talks via RS232 and cannot be connected to the board as was done with the other experiment.

Nonetheless there are still parameters and channels defined in configuration XML file of this experiment, which in this case are only for data validation and device protection.

In this experiment there are 4 parameters coming from the GUI: the voltage ramp (i) minimum, (ii) maximum and (iii) increment and (iv) the operating pressure. There are also 3 values that come from the hardware and that need to be transferred to the GUI, in the results screen: (i) the voltage and the (ii) current across the electrodes as well as the (iii) pressure inside the vessel. The XML files are identical to the one featured in the previous chapter.

# 5

# EXPERIMENT OPERATION

## 5.1 LANGMUIR PROBE

### 5.1.1 *Introduction*

Plasmas have different characteristics from other states of matter, and in order to measure them many diagnostic tools have been developed. This experiment allows the user to measure some of these characteristics using one of the most simple methods, the Langmuir Probe [3]. This probe consists of a thin filament made of conductive material [18] , placed inside the plasma, which either attracts or repels the electrons in the plasma according to its biasing. Measuring the probe I-V characteristic, that is, the relationship between the biasing voltage and the respective current going trough the probe, one can extrapolate the electron temperature and density of the plasma [19], [20], [21], [22].

When the probe is electrically isolated (floating), a plasma sheath is formed in the interface between the plasma and the probe. To compensate for the higher mobility of the electrons, the probe will attain a floating potential,$V_f$, negative with respect to the plasma potential, $V_p$. The density at the sheath entrance is roughly half of the density away from the probe [23].

The probe voltage, $V_s$, can be changed with respect to the ground set by the winding filament using a variable voltage source. If the biasing of the probe, compared to the plasma is negative enough all the electrons will be repelled and the ion flux to the probe is independent of the potential applied. In a totally ionized plasma, this ion saturation current is described by the following expression:

$$i^+_{sat} = j^+_{sat} A_s \approx \frac{1}{2} e\, n\, c_s\, A_s \tag{5.1}$$

Where (i) $j^+_{sat}$ is the current density, (ii) $A_s$ is the contact surface of the probe, (iii) $e$ is the electron charge, (iv) $n$ is the ion density in the plasma, (v) $c_s$ is the ion sound speed.

If we bias the probe positively, the voltage drop in the sheath is reduced and electrons will be able to reach the probe. Taking a Maxwell distribution for the speed of the electrons, the relation between current and tension will become:

$$i = i^+_{sat} \left(1 - e^{\frac{e}{kT_e}(V_s - V_f)}\right) \tag{5.2}$$

Where $T_e$ is the electron temperature. This expression assumes that there is only one probe and that it is non-perturbative.

### 5.1.2 *Experimental Protocol*

Since there is the possibility to choose from different gases the first thing to do is to choose which gas the experiment will be performed with. After choosing the gas the user can pick two pressures. The first one is the "Pump Down Pressure". The connection to the vacuum pump will opened until the pressure inside the vessel decreases to this value. After this, the chamber will be filled with the chosen gas until the "Gas Pressure" is reached. For this reason the "Pump Down Pressure" must always be lower than the "Gas Pressure".

After, the user can choose the characteristics for the sweeping signal by selecting its "Amplitude" and "Period". Choosing a bigger amplitude will reveal details regarding the Langmuir probe's regimes while the period will have an effect on how pertubative is the probe. A big period is desired to allow the electrons to have time to reach equilibrium and to avoid AC coupling.

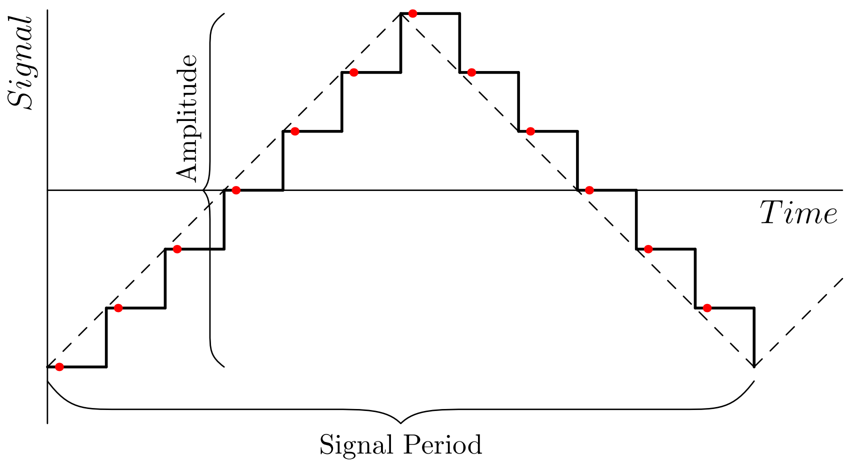



Figure 5.1.: Diagram illustrating the variables from the sweeping signal. The red dots represent data acquisitions. The full line represents the signal generated by the hardware before the final amplification, while the dashed line represents the signal after going through the transformer.

Finally the data acquisition can be configured by setting the "Number of samples per period" and the "Number of periods". "Number of samples per period" as the name suggests is the number of times, per sweep, that the ADCs perform an acquisition. However, this variable also influences the signal generation, since this is also the number of times the voltage is updated, as shown in figure 5.1. "Number of periods" is the number of triangular sweeps that the experiment runs for. Ideally, each triangular sweep is identical, since the data points are taken exactly at the same voltages, so this variable allows the user to get many similar data sets, which is necessary to achieve statistical relevance.

In a more advanced stage the user can combine more than one experiment and use the "Pump Down Pressure" and "Gas Pressure" to mix two or more different gases and thus exploring the influences of mixing gases.

### 5.1.3 *Data Analysis & Results*

As described in the previous section, when the user activates the experiment and the pressure is set, a triangular signal is established, and the ADC is used to get the signal at the probe. The software client delivers the values in the GUI of the voltage and current. It also prints the pressure before and after the experiment, during the experiment there is not enough band width to also get the pressure from the gauge, so the client prints 0, as can be seen in figure 3.11. After the selection of relevant data the plot of the points will look like the one shown in figure 5.2a. The client will also give the error to each value which is calculated based on the error propagation for the expressions that convert the measured value into the real physical value.

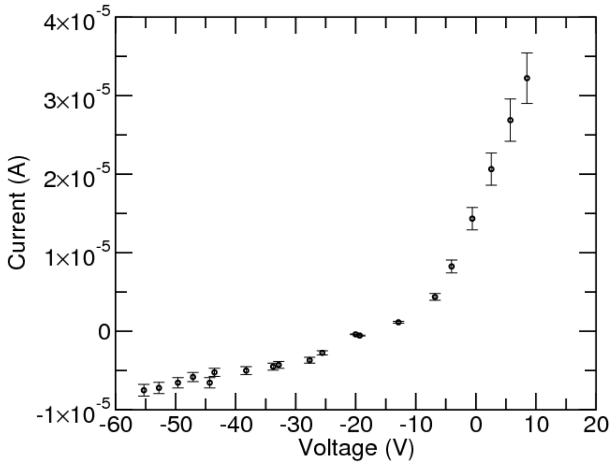
After that, one must check if the conditions of the experiment obey to the approximations made by the probe theory. The magnetic field is negligible. The ions are cold because, as mentioned, the plasma is generated by RF which mainly heats the electrons. Also due to the fact that this is not a very powerful source the ionization ratio is very small and therefore the collision mean free path is extremely big (later we will see that for these results its in the order of kilometre) so the non-collisionality of the plasma is also valid. However because of this low ionization ratio the Debye length of the system is very big (bigger than the probe diameter) meaning that this is not a one dimension system which in turn means that the equation 5.1 is only approximately true. With this one can conclude that the plasma will be perturbed by the probe.

From equation 5.2 it's possible to extract an estimate for the floating potential, $V_f$. This is done by taking the value at which the current characteristic crosses zero. With this method we obtain a floating potential estimate of $-20\,V$.

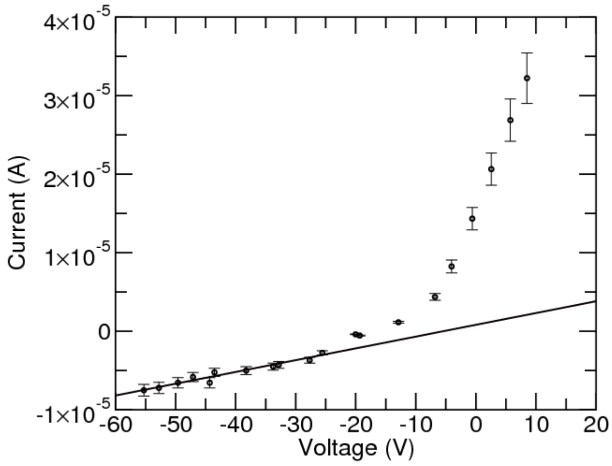
It will be easily seen in figure 5.2a that the data does not follow the regular characteristic on the ion saturation side. It should be constant instead of having a slope. This has to do with the fact that the sheath thickness expands with the applied voltage. To correct for this, a linear function is fitted on this side, as seen in 5.2b, and the slope is subtracted thus resulting in the data points of 5.2c.

However, we want to get the density, so we have to make another correction, which corresponds to add the value of current in point where we know the current is completely due to ions, namely to voltage values much lower than the floating potential (in this case we use two times the floating potential, hence the need for our initial estimate).

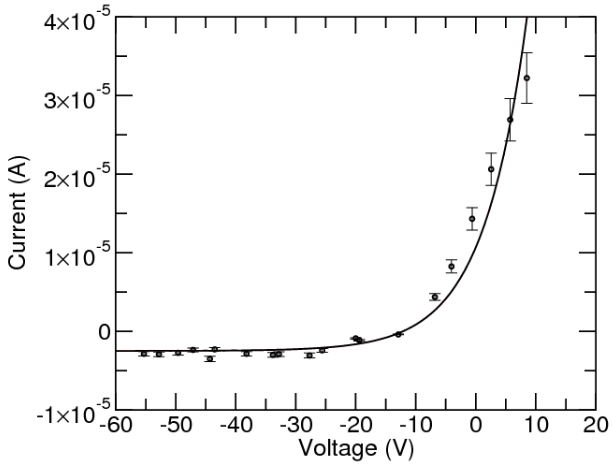
After that the experimental data has to be adjusted to the equation 5.2, as seen in figure 5.2c. From the fit $T_e$ can be extracted as well as the ion saturation current, $i^+_{sat}$ and the floating potential $V_f$ which should have a value close to the one we determined empirically.




(a) Raw data from the experiment




(b) Linear fit to the ion saturation region




(c) Fit to probe electrical characteristic, equation 5.2

Figure 5.2.: Data acquired and treated for 0.45 mBar in the Langmuir Probe experiment

For the data presented in figure 5.2c, which corresponds to mixture of Helium and Neon at 0.45 $mBar$, the fit gives a plasma temperature of $T = 7.29 \pm 0.2\, eV$, which is in the same magnitude as the ionization energy for both Helium and Neon.

By knowing the area of the probe and the plasma speed of sound, one can determine the electron density in the plasma. Since $c_s = \sqrt{\frac{kT_e}{M}} \approx 6012 ms^{-1}$ , and knowing the fact that the probe is $10mm$ [24] and has a diameter of $0.2mm$ we can use the equation 5.1 to determine the density.

For the particular case above the ion saturation current is $i^+_{sat} = -2.5 \pm 0.08\, \mu A$ and the floating potential $V_f = -12.1 \pm 0.1V$, this value for floating potential is close to the one we determined empirically thus confirming our initial guess. Finally, the density is determined giving a value of $n = 8.3 \times 10^{14} \pm 0.3 \times 10^{14} m^{-3}$. To make a estimative of the ionization ration one can assume the gas inside the vessel is at room temperature of $298K$. Then the gas has a density of about $1.1 \times 10^{22} m^{-3}$. Given this value one can argue that the gas is very poorly ionized since the ratio between the plasma and the gas density is $7.6 \times 10^{-8}$. Furthermore one can confirm the comments made earlier as the collision mean free path [3] gives more than $2km$ and the Debye length, $\lambda_D = \sqrt{\frac{\epsilon_0 k T_e}{ne^2}}$ [3], yields a value of $0.69mm$. This second value is bigger than the radius of the probe.

## 5.2 PASCHEN CURVE

### 5.2.1 *Introduction*

The state of matter transition from gas into plasma is investigated in this experiment, allowing the study of the Paschen's law. This is done by means of an apparatus similar to that used by Paschen in which a voltage is applied to two parallel electrodes surrounded by low pressure gas. The breakdown voltage is determined as a function of pressure and distance separating the two plates.

The breakdown phenomenon can be interpreted as a chain reaction where one charged particle collides with a neutral and generates an extra charged particle. If this collision process has a net gain, then there will be a discharge across the electrodes, otherwise the process will decay and the ionization will stop. It is therefore required that energy of the impacting particle exceeds the ionization energy of the neutral particle as well as enough such that there is no recombination and this is both a function of the gas pressure and the plates distance.

In order to intuitively understand the phenomena, one can break it down into two scenarios which illustrate the dependencies of the two variables of the experiment. This is done by keeping one of the variables constant while the other varies and dividing the curve in two regions left and right to its minimum.

Maintaining a constant distance between plates and going from low to a high pressure, the voltage necessary to arc decreases up to a point, the minimum, as the pressure is reduced. This can be pictured as a situation in which there are not enough particles to carry the chain process since the mean free path of these particles is bigger than the distance between the two electrodes. The breakdown voltage then increases, greatly exceeding its original value. In the previous picture, we now have an abundance of particles and the process cannot progress any further because the distance between collisions is so small that the energy picked up by the electrons isn't enough to sustain the ionization process.

On the other hand, one can maintain a constant pressure. In which case it can also be found that the voltage needed to cause an arc reduces proportionality with a decrease in the gap size, but only up to a point. As the gap is reduced further, the required voltage begins to rise and again, exceeding its original value. The same intuitive picture can be used to describe these situations, a small gap doesn't allow the particle collide many times before it hits the electrode and a big gap might make it so that the energy picked up the particle's mean free path is not enough to ionize.

It can be shown [25], using the model for a chain reaction, that the breakdown voltage for given conditions is described by the equation:

$$V = \frac{a.pd}{\ln(pd) + b} \tag{5.3}$$

Where (i) $V$ is the breakdown voltage, (ii) $p$ is the pressure, and (iii) $d$ is the gap distance. The constants (iv) $a$ and (v) $b$ depend upon the composition of the gas.

### 5.2.2 *Experimental Protocol*

Taking into account the fact that there is no remote control over the distance between electrodes, the first step for the user should be to devise the range of pressure trough which the experiment will be performed. This will determine the number of times he will have to do the experiment. To configure this parameter one must use the "Gas Pressure" slider. While the experiment is not in use the vacuum pump will be connected to the main reaction chamber since the last time the experiment was performed until the moment the experiment was activated. Considering that by the end of the experiment the pump down is monitored as it goes down until it reaches a limit of $0.05 mBar$ the pressure inside the vessel before the experiment starts will always be at least that one. After experiment begins, the chamber will be filled with Argon gas until the "Gas Pressure" is reached.

After that, the user can choose the characteristics for the voltage sweeping selecting its "Maximum", "Minimum" and "Increase Step", which affect the signal as shown in figure 5.3. Choosing a bigger step might hide the details but will allow for a faster determination of the region at which the breakdown occurs while a smaller step will provide a higher detail on the determination of the data. It is recommended that the step value remains constant since the quickly varying signal of the steps helps facilitating the breakdown and if it changes throughout the experiment the data will not be usable.

Finally the data acquisition is also configured by setting the "Maximum", "Minimum" and "Increase Step", since the ADCs performs a data acquisition every time the voltage is increased. Therefore the number of data points is equal to the number of times the voltage has to update.

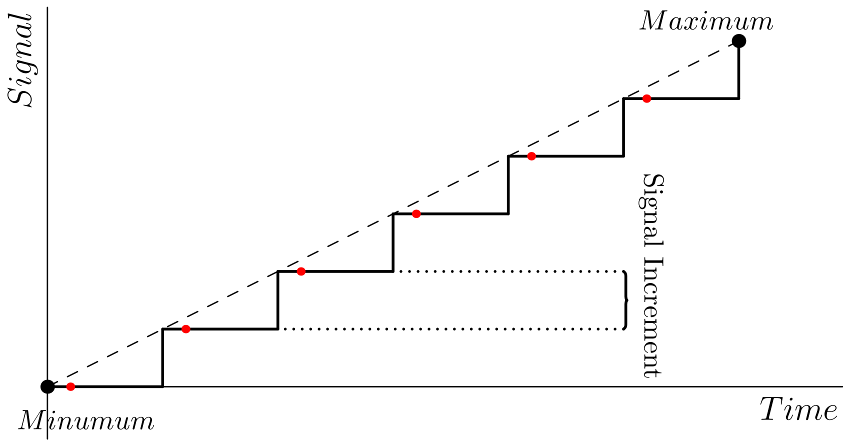



Figure 5.3.: Diagram illustrating the variables from the sweeping signal. The red dots represent data acquisitions. The full line represents the signal generated by the hardware, while the dashed line represents the ideal signal. The smaller the step values are the closer the signal become to the ideal.

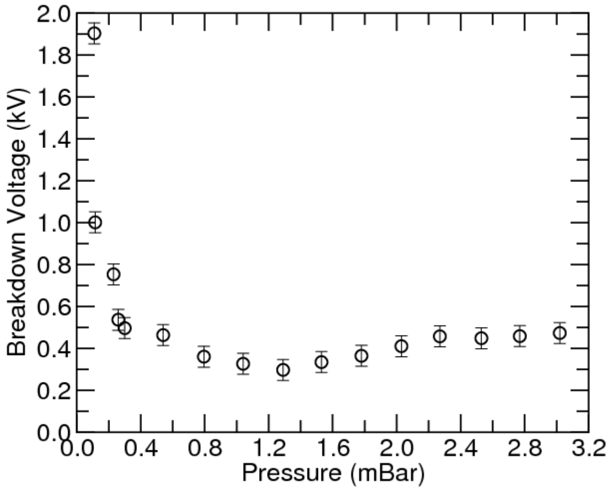
As said in the beginning, this setup performs the experiment under constant pressure so this will only result in one data point. To get the Paschen Curve the user should go trough a large range of pressures.

### 5.2.3 *Data Analysis & Results*

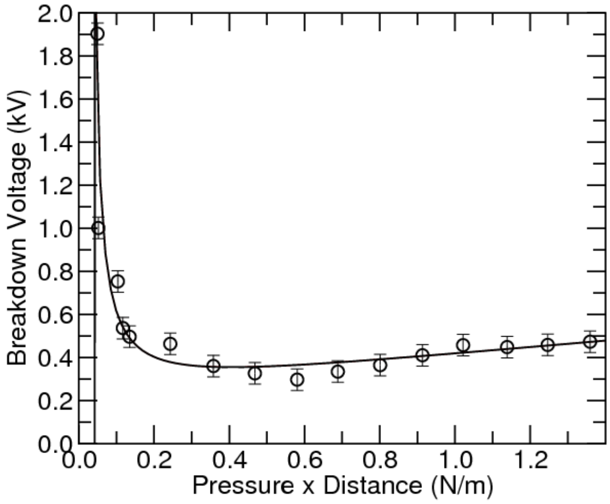
The data from one experiment gives only one point, so the experiment should be repeated multiple times for different pressures. As with the other e-lab experiment, when the user activates the experiment and the pressure is set, a voltage ramp starts sweeping the electrodes, and the ADC is used to get both current and voltage applied to the electrodes. The software client prints the values of the voltage, current and pressure in the interface during the whole experiment. After that, the user will be able to see a clear transition from 0 to saturation in the current graph. The corresponding point in the voltage is the breakdown for that pressure. After gathering the points for many different pressures the user will have a data set similar to the one displayed in figure 5.4a. Although there is a very high precision in the determination of the voltage values in the chamber, multiple runs of the experiment under the same conditions will show that often there is a range of about 50V under which the disruption can occur. Therefore this value was used for errors in the fit instead of the error with the ADC.

As it is the data is still in raw form. The values in pressure must be multiplied by the gap distance and only after this is the experimental data ready to be adjusted to the equation 5.3, as seen in figure 5.4b. In order to accommodate for a systematic error in the measurements of pressure or distance one should add a extra fitting parameter:

$$V = \frac{a.(pd + c)}{\ln(pd + c) + b} \tag{5.4}$$




(a) Raw data from the experiment




(b) Fit to the Paschen's law, equation 5.3

Figure 5.4.: Data acquired and treated for 45 mm gap

From the fit $a$ and $b$ can be extracted, which can allow the determination of the gas inside the chamber. By opposite taking the information regarding the gas inside the chamber, in this case Argon, the data points can be used to determine the distance between the plates by fitting the data to equation 5.3.

For the data presented in figure 4.5, which corresponds to Argon at 1.05 mBar, one can determine that the breakdown occurs at $370V$. Taking all the data points, as seen in figure 5.4b and the knowledge that the distance between plates is of $45\,mm$ the fit gives a $a = 590 \pm 30\,V/(Pa.m)$ and $b = 1.51 \pm 0.49$, which is within the same order of magnitude as the values in literature [26]. However the value is about a factor 3 off which might have to do with multiple reason namely gas contamination or poor isolation from the cosmic radiation that easily ionizes the gas. The fit also gives $c = 0.19 \pm 0.01\,Pa.m$ this means that there is a systematic error in the measurement of pressure or gap distance. This means that there is an error in the order of magnitude of a few tenths of micro-bars in pressure or millimetre in distance. In both cases this is corresponds to the least significant digit.

By differentiating the equation 5.4 one can determine the optimal conditions at which the transition from gas to plasma occurs. For the case above this happens at $pd = \exp(1-b) - c = 0.41\,Pa.m$ with a breakdown voltage of $V_b = 354\,V$.

# 6

# CONCLUSION

The remote laboratory e-lab hosts many experiments with different degrees of difficulty. In the work developed for this thesis two experiments, the Langmuir Probe and the Paschen Curve, were successfully added to the advanced laboratory on e-lab.

The Langmuir Probe experiment gives data regarding the sweeping voltage that changes the bias of the probe and it's respective current, thus allowing for the construction of the electrical characteristic of the probe. It also gives information regarding the pressure inside the vessel which in conjunction with a deep analysis of the probe's characteristic gives information on many plasma parameters, namely the electron temperature and the plasma density, both of which are essential to characterise the plasma. A system was devised to switch or mix 3 different gases inside the chamber allowing for a variety of configurations.

The Paschen Curve experiment gives data regarding pressure, voltage and current during the evolution of the experiment allowing the user to determine at which point breakdown is achieved. Successive experiments with different pressures allows the user to get the Paschen Curve, the relation between the breakdown voltage and, in this case, the pressure inside the vessel. The experiment was used in classroom environment, in which the students, after a lecture on the subject and on how to use the experiment were able to obtain successful results.

It became evident during the implementation of the Paschen Curve experiment that there is a disagreement regarding the concept of remotely controlled experiments namely in the degree of interactivity with the experiments. This has to do with the methodology under which e-lab operates which is based in stages (ie: there is a well defined underlying state-machine), the experiment is configured by the user, this configuration is verified and loaded to the experimental apparatus which then executes the experiment. Either in the end or during the execution the data is displayed. Assuming everything is correct all of this is done really quickly allowing the experiment to be repeated. However only after the end of the experiment can a user re-configure the apparatus. This impediment is usually seen as a feature as it suggests a structured way of performing the experiment which is the common methodology in large research centers. However a different methodology arises if the real-time concept is taken to the limit. In this case the configuration, validation, execution and the result's display are cycled trough very fast so that the user can immediately see the consequences of its configuration. Such an approach would bring the user's interaction closer to what is typical of a real school laboratory where there the actions have immediate reaction. There aren't, as of the time of writing this thesis, no implementations on the ReC for this methodology.

Due to financial constrains, there was no way to acquire a positioner for the Langmuir Probe, which would allow a density profiles study along the central axis. This would allow an advanced protocol where plasma density axial fading could be measured.

Due to mechanical constrains, there is no way to automatically control the distance between the electrodes in the Paschen Curve. This could be surpassed by a stepper motor coupled with the central shaft allowing a complete 2D mapping of voltage and pressure ranges. Moreover it will introduce a reliable way to measure accurately the distance between electrodes.

Comparison should be drawn between the two systems used to control the experiments. On one hand there is the typical e-lab board, the dspicnode, which has as its "brain" the dsPIC30F4011. The microprocessor has its own program memory making the board a completely autonomous device, specially taking into account the large variety of peripherals it has and allowing the possibility for another level of input validation making the apparatus more secure. This means that a higher level device only has to give a command and the board can perform extensive and complicated tasks. However at the expense of speed due to the program execution order.

The PicoGiant on the other hand is based of a field programmable gate array which means very small independence but very fast compliance. Although this board is capable of interfacing with many devices it almost doesn't have any embedded on itself making it necessary to introduce all the components externally. In this case it means acquiring expensive components for the experimental apparatus compatible with the board. Also since it doesn't have a program memory it needs to be externally controlled, via USB to a computer, in order to perform complicated tasks. For the application at hand this device is extremely overpowered since controlling such an experiment doesn't require fast data acquisition or fast signal generation.

In the end both are devices suited for controlling remote experiments. However the dspicnode provides a less expensive and more versatile device while the PicoGiant provides higher speed but requiring a much more expensive setup (as it needs a computer to control it as well as components external to the board). Therefore the balance should be achieved between the requirements and the budget.

Currently, both the server and the computer cloud that runs the driver are in the same room. However the introduction of the Paschen Curve experiment, located in Eindhoven, has allowed the test of e-lab distributed capabilities. As a matter of fact, until now all of the e-lab apparatus and their hardware drivers were within the same intranet. But in this case the server is at IST in Portugal while the experiment is at the PlasmaLab at TU/e in the Netherlands. This opens the possibility for an experiment to be added from anywhere to e-lab without the need to set up a new server on-site, demonstrating the full capabilities of the e-lab framework.

# BIBLIOGRAPHY


[1] "e-lab" main website: `http://elab.ist.utl.pt` (10/09/2013).

[2] "Fusenet" main website: `http://www.fusenet.eu` (10/09/2013).

[3] F. Chen. *Introduction to plasma physics and controlled fusion*, volume 1. Springer, 1974. Electrostatic Probes.

[4] J. Ma and J.V. Nickerson. Hands-on, simulated, and remote laboratories: A comparative literature review. *ACM Computing Surveys*, 38(3): article 7, September 2006.

[5] D. G. Kasten. Integrating computerized data acquisition and analysis into an undergraduate electric machines laboratory. In *proceedings of the 30th ASEE/IEEE Frontiers in Education Conference*, Kansas City, MO., 2000.

[6] K. Torres and *et al*. Introducing 9-12 grade students to electrical engineering technology through hands-on laboratory experiences. In *proceedings of the 2001 31st Annual Frontiers in Education Conference*, Reno, NV., 2001.

[7] M. Cooper, A. Donnelly, and J. M. Freira. Remote controlled experiments for teaching over the internet: A comparison of approaches developed in the pearl project. In *proceedings of the ASCILITE Conference 2002*, Auckland, New Zealand, 2002. UNITEC Institution of Technology.

[8] H. W. Tzeng. The design of pedagogical agent for distance virtual experiment. In *proceedings of the 2001 31st Annual Frontiers in Education Conference*, Reno, NV., 2001.

[9] "labshare" main website: `http://www.labshare.edu.au` (25/08/2013).

[10] "iLab" main website: `http://openilabs.mit.edu` (25/08/2013).

[11] "UNED Labs" main website: `http://unedlabs.dia.uned.es` (25/08/2013).

[12] dspic30f family reference manual. Microchip `http://www.microchip.com` (10/09/2013), 2006.

[13] Pfeiffer vacuum protocol - interface rs 232, pm 800 488 bn/c (0309). Pfeiffer Vacuum `http://www.pfeiffer-vacuum.com` (10/09/2013).

[14] Creating a new experiment gui, version 1.0. Linkare `http://www.linkare.com` (10/09/2013).

[15] R. B. Henriques, H. Fernandes, and *et al*. Generic protocol for remotely controlled experiments @ e-lab. In *proceedings for the 1st Experiment@ International Conference*, 2011.

[16] Creating a new experiment driver, version 1.1. Linkare `http://www.linkare.com` (10/09/2013).

[17] "TUeDACs" main website: `http://www.tuedacs.nl` (10/09/2013).

[18] V. A. Godyak and V. I. Demidov. Probe measurements of electron-energy distributions in plasmas: what can we measure and how can we achieve reliable results? *Journal of Physics D: Appl. Phys.*, 44, 2011.

[19] V. A. Godyak, R. B. Piejak, and B. M. Alexandrovich. Measurements of electron energy distribution in low-pressure rf discharges. *Plasma Sources Science Technology*, 1:38–58, 1992.

[20] Tsv Popov, M Dimitrova, F. M. Dias, N. A. Tsaneva, V. N. Stelmashenko, M. G. Blamire, and Z. H. Barber. Second derivative langmuir probe diagnostics of gas discharge plasma at intermediate pressures. *Journal of Physics: Conference Series*, 44:60–69, 2006.

[21] R. R. Arslanbekov, N. A. Khromov, and A. A. Kudryavtsev. Probe measurements of electron energy distribution function at intermediate and high pressures and in a magnetic field. *Plasma Sources Science Technology*, 3:528–538, 1994.

[22] V. Guerra, F. Dias, J. Loureiro, P. Sá, P. Supiot, C. Dupret, and Tsv Popov. Time-dependence of the electron energy distribution function in the nitrogen afterglow. *IEEE Trans Plasma Sci*, 31:542–552, 2003.

[23] P. C. Stangeby. *The Plasma Boundary of Magnetic Fusion Devices*. Institute of Physics Publishing, 2000. 1D Plasma Flow along the simple SOL to a Surface.

[24] J. D. Swift. Effects of finite probe size in the determination of electron energy distribution functions. *Proc. Phys. Soc.*, 79:697, 1962.

[25] J. P. Freidberg. *Plasma Physics And Fusion Energy*. Cambridge University Press, 2007.

[26] Y. P. Raizer. *Gas Discharge Physics*. Springer-Verlag, 1991.

[27] A. Ellett and R.M. Zabel. The pirani gauge for the measurement of small changes of pressure. *Phys. Rev.*, 37(9):1102–1111, May 1931.

[28] H. M. Mott-Smith and I. Langmuir. The theory of collectors in gaseous discharges. *Phys. Rev.*, 28(4):727–763, 1926.

[29] R. L. Merlino. Understanding langmuir probe current-voltage characteristics. *Am. J. Phys.*, 75(12):1078–1085, December 2007.

[30] R. J. Goldston and P. H. Rutherford. *Introduction to Plasma Physics*. Institute of Physics Publishing, 1997.

[31] J.A.C. Cabral. Seminário sobre propagação e radiação de ondas electro-magnéticas, 1979.

[32] D. Bohm. *The Characteristics of Electrical Discharges in Magnetic Fields*. McGraw-Hill, 1949.

[33] J. S. Townsend. *The Theory of Ionization of Gases by Collision*. Constable and Company LTD, 1910.

# A

# DEMONSTRATIONS

## A.1 DERIVATION OF LANGMUIR PROBE CHARACTERISTIC

In the interface between a solid and a plasma, a thin net-charge layer called the Debye sheath develops spontaneously. This sheath mediates the flow of particles and energy out of the plasma to the solid surface. Because of that it is of major importance in setting the temperature, density and other properties of the plasma. The sheath allows a separation of the plasma into an upstream, unaffected region and a downstream, interface region.

The presence of a surface that acts as a sink causes a depression of the local plasma density. This leads to a pressure gradient to which plasma reacts by creating a pre-sheath field that retards the electrons. That leads to a force balance between the pressure pushing the electrons to the wall and the electric field force that retracts them. It can be shown [30] that because of this the electrons must obey the Boltzmann factor relation:

$$n = n_0 \exp\left(\frac{eV}{kT_e}\right) \tag{A.1}$$

With $k$ the Boltzmann constant, $T_e$ the electron temperature, $e$ the elementary charge and $V$ the voltage drop across the sheath.

By analysis of the plasma sheath interface on the upstream side taking the "isothermal fluid model with a particle source proportional to density" assumption it is shown [23] that the speed at the sheath edge $v_{se}$ cannot exceed the isothermal sound speed $c_s$:

$$c_s = \sqrt{(kT_e + \gamma kT_i)/m_i} \tag{A.2}$$

Where $T_i$ is the ion temperature, $m_i$ the ion mass and $\gamma$ flow constant. However throughout this demonstration the "cold ion" assumption is made,so $T_i \approx 0$ and therefore $c_s \approx \sqrt{kT_e/m_i}$. Furthermore its also assumed that its a uncollisional plasma, that the magnetic field is 0, that the system is one dimensional and that the Debye length is much smaller than the solid surface. This means that $v_{se} \leq c_s$

however if the analysis is done from the sheath side of the plasma-sheath interface it can be shown [32] that $v_{se} \geq c_s$. Therefore one can only conclude that:

$$v_{se} = c_s \tag{A.3}$$

Taking the momentum conservation equation for the plasma at the interface it can be shown [23] for the isothermal assumption that:

$$n(x) = \frac{n_0}{1 + (v(x)/c_s)^2} \tag{A.4}$$

And in the particular case of sheath edge this gives that $n_{se} = \frac{1}{2}n_0$. So for isothermal conditions the density drops only by a factor of 2 from upstream to the sheath. Note that this is still approximately true for small deviations on the isothermal assumption. Also the particle flux density at the sheath edge, $\Gamma_{se}$ is:

$$\Gamma_{se} = n_{se}c_s = \frac{1}{2}n_0c_s \tag{A.5}$$

Assuming a Maxwellian velocity distribution it is trivial to show that the average particle speed, or thermal speed, for a non-drifting Maxwellian is $\bar{c} = \left(\frac{8kT_e}{\pi m_e}\right)^{1/2}$. Note as well that $4c_s = \bar{c}$. Equally trivial is the one-way particle flux density for a Maxwellian speed distribution, $\Gamma^{max}$ is:

$$\Gamma^{max} = \frac{1}{4}n\bar{c} \tag{A.6}$$

We can now take into account a floating surface. In this case the contact between the plasma and the surface is non-existent due to the arising of the sheath. In this case $\Gamma^e_{se} = \Gamma^i_{se}$. Since the action of the electrostatic field leaves the velocity distribution Maxwellian, the particle flux density at the sheath is still given by equation A.6. To go from the sheath entrance to the wall one must use the Boltzmann relation given in equation A.1. Therefore we have:

$$\Gamma^e_w = \Gamma^i_w = n_w c_s = \frac{1}{4}n_{se}\overline{c_e}\exp\left(\frac{eV_w}{kT}\right) \tag{A.7}$$

where $V_w$ is the potential of the wall relative to the plasma potential at the sheath edge where $V = 0$. For floating conditions the wall is at a floating potential, $V_{sf}$.

Subsequently we analyse a non-floating surface, biased with external applied voltage, $V_{applied}$ and see how the fluxes react. Regardless of whether the wall is floating or not $\Gamma^i_w$ is still given by the equation A.7. In other words for the ions the flux remains the same as if $V_{applied} = 0$. Taking into account a symmetrical situation with floating surfaces in both ends to which an external bias, $V_{applied}$, is applied one can say that regarding the particle fluxes that:

$$\Gamma^e_{rw} = \frac{1}{4}n_{se}\overline{c_e}\exp\left(\frac{eV_r}{kT}\right) \tag{A.8}$$

$$\Gamma^i_{rw} = \frac{1}{4} n_{se} \overline{c_e} \exp\left(\frac{eV_{sf}}{kT}\right) \tag{A.9}$$

$$\Gamma^e_{lw} = \frac{1}{4} n_{se} \overline{c_e} \exp\left(\frac{eV_l}{kT}\right) \tag{A.10}$$

$$\Gamma^i_{lw} = \frac{1}{4} n_{se} \overline{c_e} \exp\left(\frac{eV_{sf}}{kT}\right) \tag{A.11}$$

Where the voltage drop across the right and left sheaths are respectively $V_r$ and $V_l$. If we take the flow balance for one of the sides one arrives to the conclusion that there is a net current:

$$\begin{aligned} j_r &= e\left(\Gamma^i_{rw} - \Gamma^e_{rw}\right) (= -j_l) \\ &= \frac{1}{4} e n_{se} \overline{c_e} \left(\exp\left(\frac{eV_{sf}}{kT}\right) - \exp\left(\frac{eV_r}{kT}\right)\right) \\ &= \frac{1}{4} e n_{se} \overline{c_e} \exp\left(\frac{eV_{sf}}{kT}\right)\left(1 - \exp\left(\frac{e(V_r - V_{sf})}{kT}\right)\right) \\ &= e n_{se} c_s \left(1 - \exp\left(\frac{e(V_r - V_{sf})}{kT}\right)\right) \end{aligned} \tag{A.12}$$

Furthermore the conservation of charge gives that:

$$\Gamma^e_{lw} + \Gamma^e_{rw} = \Gamma^i_{lw} + \Gamma^i_{rw} = 2 n_{se} c_s \tag{A.13}$$

We also have that:

$$V_r - V_l = V_{applied} \tag{A.14}$$

And if we combine A.10 and A.8 with A.13 and A.14 we get:

$$\begin{aligned} \Gamma^e_{lw} + \Gamma^e_{rw} &= 2 n_{se} c_s \\ \frac{1}{4} n_{se} \exp\left(\frac{eV_l}{kT}\right) + \frac{1}{4} n_{se} \exp\left(\frac{eV_r}{kT}\right) &= \frac{1}{2} n_{se} \exp\left(\frac{eV_{sf}}{kT}\right) \\ \exp\left(\frac{eV_l}{kT}\right) + \exp\left(\frac{eV_r}{kT}\right) &= 2 \exp\left(\frac{eV_{sf}}{kT}\right) \\ \exp\left(\frac{eV_l}{kT}\right)\left[1 + \exp\left(\frac{e(V_r - V_l)}{kT}\right)\right] &= 2 \exp\left(\frac{eV_{sf}}{kT}\right) \\ V_l &= \frac{kT_e}{e} ln \left[\frac{2 \exp\left(\frac{eV_{sf}}{kT}\right)}{1 + \exp\left(\frac{eV_{applied}}{kT}\right)}\right] \end{aligned} \tag{A.15}$$

It is very enlightening if we take these results to the case when $V_{applied} \rightarrow -\infty$. Then $V_l \rightarrow V_{sf} + \frac{kT_e}{e} ln(2)$, $V_r \rightarrow -\infty$ and $j_r \rightarrow en_{se}c_s$. As it can be seen almost all the applied voltage goes to the right, the "electron-repelling" side, while the left, "electron-attracting" side saturates both its sheath voltage drop and the current. Therefore one can define ion saturation current:

$$j^+_{sat} \equiv en_{se}c_s \approx \frac{1}{2}en_0c_s \tag{A.16}$$

Finally one can take the Langmuir Probe. Taking the most simple geometry of a central rod as the probe one can define $A_s$ as the contact surface of the probe. The current from the probe must close through the plasma and return to vessel walls. Since the return area and $A_s$ are very different, $A_s \ll A_{return}$ the case of the real Langmuir Probe is not as the one considered above. This has a very important consequence, the sheath potential drop on the return side wont change very much as as consequence of the change in the passing current. So we can consider this as a constant which means it is only a offset in the I-V characteristic designated floating potential, $V_f$ . Taking the current at the probe as $i = A_s j$ gives:

$$i = i^+_{sat}\left(1 - \exp\frac{e}{kT_e}(V_s - V_f)\right) \tag{A.17}$$

## A.2 DERIVATION OF PASCHEN'S LAW

Usually the plasma breakdown is the mechanism that leads to the transformation of a gas (non-conductive) into a plasma (conductive) by means of the application of sufficiently strong field. The underlying process that ultimately leads to the breakdown is the electron avalanche, which develops in the gas when a strong enough electrical field is applied to it. Such a process starts when a number of free "seed electrons" get accelerated by the field, collide the neutrals in the media and ionize the gas. These "seeds" can come about accidentally, usually from cosmic rays. If these electrons get an energy higher than the ionization energy for the background gas then the collision results in two slow electrons. In turn each one of these can gather again enough energy and then collide generating more electrons. This process continues indefinitely until the media turns into a plasma. The threshold value of the field at which this avalanche starts is designated the electrical breakdown field, $E_{breakdown}$.

In the study of this phenomena it is commonly introduced a parameter $\alpha$ which is the number of ionizations performed by an electron per length of path. The functional form of this parameter was empirically suggested by Townsend [33] as:

$$\alpha = Bp\exp\left(-\frac{Ap}{E}\right) \tag{A.18}$$

Where $p$ is the gas pressure, $E$ the electrical field, $A$ and $B$ are experimentally determined constants.

The electrical field is usually generated using two parallel plate electrodes separated by a distance $d$ to which a voltage $V$ is applied. Taking now the fact that this is the parallel plates capacitor geometry we also have that $E = V/d$. Substituting this into equation A.18 gives:

$$\alpha = Bp \exp\left(-\frac{Apd}{V}\right) \tag{A.19}$$

the definition of ignition [33] where the discharge must be self-sustaining, that is, to be capable of having a steady current without the need for external "seed" electrons should be taken into consideration. In such a case the electrons from the cathode must reach the anode and ionize at least one atom by means of collision in his way. Therefore one can write $\alpha d \geq 1$, that is the number of ionizations performed by an electron during its transverse of the gap must be bigger than one. For the limit situation where $\alpha d = 1$ the equation A.19 can be reorganized to get the Paschen's law:

$$\begin{aligned}
\frac{1}{d} &= Bp \exp\left(-\frac{Apd}{V_{breakdown}}\right) \\
1 &= Bpd \exp\left(-\frac{Apd}{V_{breakdown}}\right) \\
(Bpd)^{-1} &= -\frac{Apd}{V_{breakdown}} \\
V_{breakdown} &= Apd\left[ln\left(pdB\right)\right]^{-1} \\
V_{breakdown} &= \frac{a.pd}{\ln(pd)+b}
\end{aligned} \tag{A.20}$$

# B

# PINOUT TABLES

## B.1 DSPICNODE PINOUT

| Board pin | Connected to |
|---|---|
| AN2 (RB2) | Current Measurement |
| AN3 (RB3) | Voltage Measurement |
| OC3 (RD2) | Flow Valve |
| OC4 (RD3) | Sweeping Noise Generation |
| TX1 | TX Channel for RS232 Pressure Gauge |
| RX1 | RX Channel for RS232 Pressure Gauge |
| PWM3_ L (RE4) | Relay 1 - Gas 1 |
| RF6 | Relay 2 - Gas 2 |
| OC1 (RD0) | Relay 3 - Gas 3 |
| AN8 (RB8) | Relay 4 - Backup/Expansion |
| OC2 (RD1) | Relay 5 - High Voltage Generator |
| PWM3_ H (RE5) | Relay 6 - Vacuum Pump & Cut-valve |

Table B.1.: Pinout table for the dspicnode board

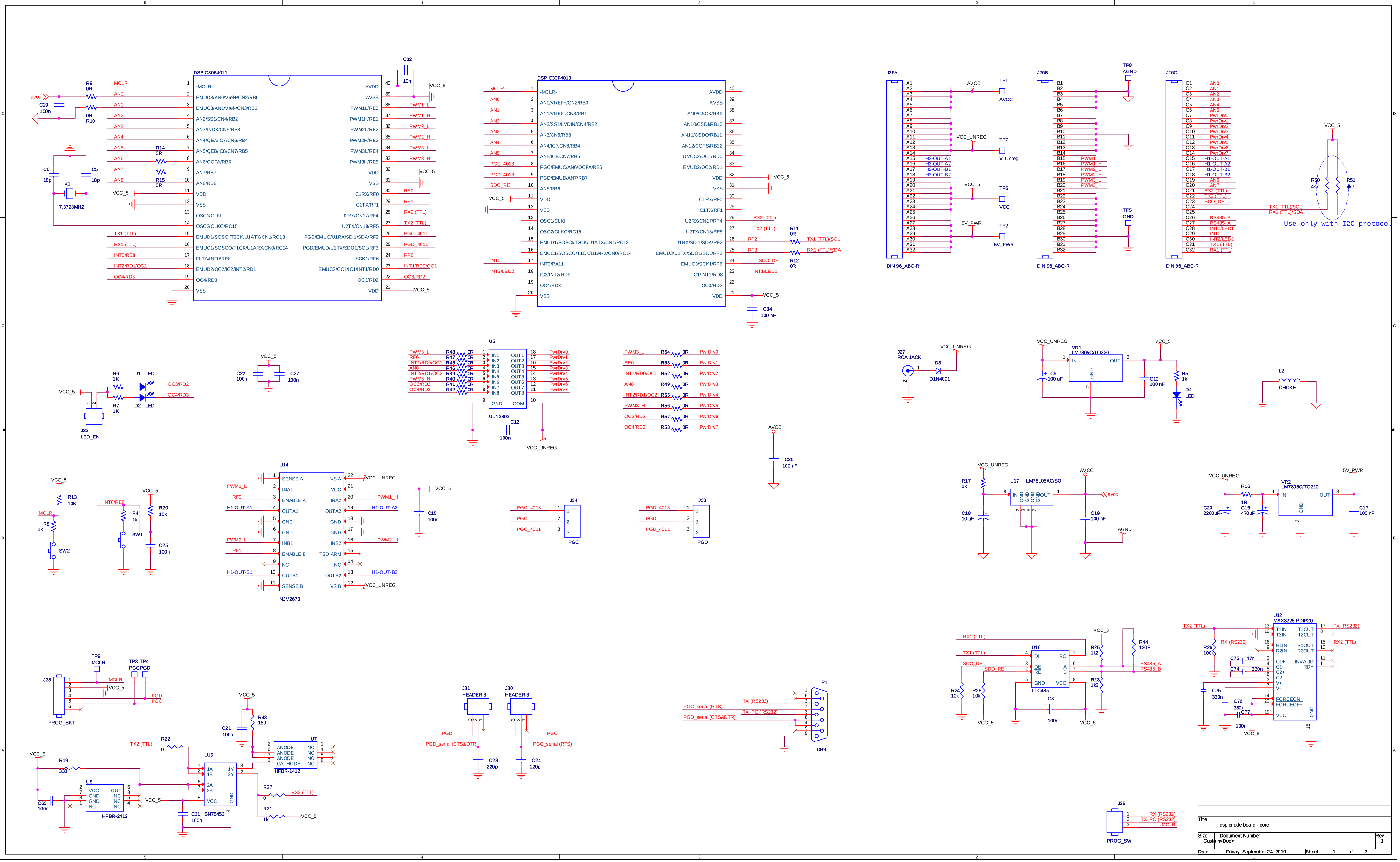
## B.2 PICOGIANT PINOUT

| Board pin | Connected to |
|---|---|
| DAC1 | Voltage Source Control |
| DAC2 | Flow Valve |
| ADC-IN1 | Voltage Monitorization |
| ADC-IN2 | Current Monitorization |
| DIO 8 | Switch Valve |

Table B.2.: Pinout table for the PicoGiant board

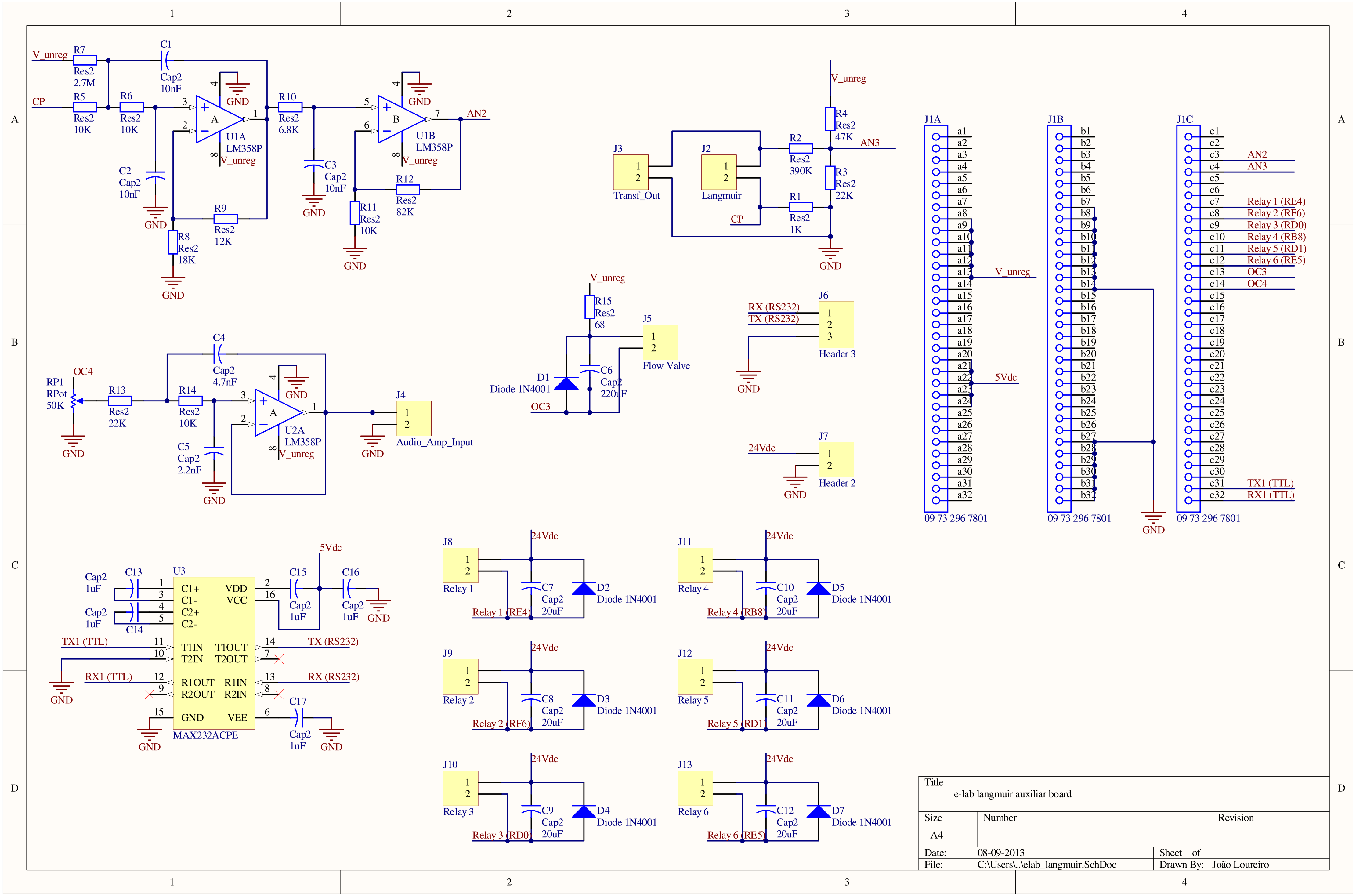
# C

# DSPICNODE SCHEMATIC


Use only with I2C protocol
DSPIC30F4011
DSPIC30F4013
DIN 96_ABC-R
ULN2803
NJM2670
LM7805C/TO220
LM78L05AC/SO
LTC485
MAX3225 PDIP20
SN75452
HFBR-1412
HFBR-2412
RCA JACK
PROG_SKT
PROG_SW
LED_EN
CHOKE
DB9
Title
dspicnode board - core
Size
Custom
Document Number
<Doc>
Rev
1
Date: Friday, September 24, 2010
Sheet 1 of 3

# D

# LANGMUIR AUXILIARY BOARD SCHEMATIC


V_unreg
CP
R7 Res2 2.7M
R5 Res2 10K
R6 Res2 10K
C1 Cap2 10nF
C2 Cap2 10nF
U1A LM358P
U1B LM358P
R8 Res2 18K
R9 Res2 12K
R10 Res2 6.8K
C3 Cap2 10nF
R11 Res2 10K
R12 Res2 82K
AN2
AN3
R4 Res2 47K
R2 Res2 390K
R3 Res2 22K
R1 Res2 1K
J3 Transf_Out
J2 Langmuir
GND
RP1 RPot 50K
OC4
R13 Res2 22K
R14 Res2 10K
C4 Cap2 4.7nF
C5 Cap2 2.2nF
U2A LM358P
J4 Audio_Amp_Input
R15 Res2 68
D1 Diode 1N4001
OC3
C6 Cap2 220uF
J5 Flow Valve
J6 Header 3
RX (RS232)
TX (RS232)
J7 Header 2
24Vdc
5Vdc
J1A 09 73 296 7801
J1B 09 73 296 7801
J1C 09 73 296 7801
Relay 1 (RE4)
Relay 2 (RF6)
Relay 3 (RD0)
Relay 4 (RB8)
Relay 5 (RD1)
Relay 6 (RE5)
TX1 (TTL)
RX1 (TTL)
U3 MAX232ACPE
C1+ C1- C2+ C2- T1IN T2IN R1OUT R2OUT GND VDD VCC T1OUT T2OUT R1IN R2IN VEE
C13 Cap2 1uF
C14 Cap2 1uF
C15 Cap2 1uF
C16 Cap2 1uF
C17 Cap2 1uF
J8 Relay 1
J9 Relay 2
J10 Relay 3
J11 Relay 4
J12 Relay 5
J13 Relay 6
C7 Cap2 20uF
C8 Cap2 20uF
C9 Cap2 20uF
C10 Cap2 20uF
C11 Cap2 20uF
C12 Cap2 20uF
D2 Diode 1N4001
D3 Diode 1N4001
D4 Diode 1N4001
D5 Diode 1N4001
D6 Diode 1N4001
D7 Diode 1N4001
Title
e-lab langmuir auxiliar board
Size A4
Number
Revision
Date: 08-09-2013
Sheet of
File: C:\Users\..\elab_langmuir.SchDoc
Drawn By: João Loureiro