AMS Test Design and Automation

Major Project Report

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology

in

Electronics & Communication Engineering

(Embedded Systems)

By

Kanika Sharma

(14MECE07)

तमसो मा ज्योतिर्गमय

NIRMA
UNIVERSITY

INSTITUTE OF TECHNOLOGY

NAAC ACCREDITED ‘A’ GRADE

Electronics & Communication Engineering Department
Institute of Technology-Nirma University
Ahmedabad-382 481
May 2017

AMS Test Design and Automation

Major Project Report

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology

in

Electronics & Communication Engineering

(Embedded Systems)

By

Kanika Sharma
(14MECE07)

Under the guidance of

External Project Guide:

Chetan Hooli
Director, Validation and Automation,
Atonarp Microsystems India Pvt. Ltd.,
Bangalore.

Internal Project Guide:

Prof. Dhaval Shah
Assistant Professor, EC Department,
Institute of Technology,
Nirma University, Ahmedabad.

Electronics & Communication Engineering Department
Institute of Technology-Nirma University, Ahmedabad-382 481
May 2017

Declaration

This is to certify that

a. The thesis comprises my original work towards the degree of Master of Technology in Embedded Systems at Nirma University and has not been submitted elsewhere for a degree.

b. Due acknowledgment has been made in the text to all other material used.

- Kanika Sharma
14MECE07

Disclaimer

"The content of this paper does not represent the technology,opinions,beliefs, or positions of Atonarp Microsystems India Pvt. Ltd.,its employees,vendors, customers, or associates."

Certificate

This is to certify that the Major Project entitled **"AMS Test Design and Automation"** submitted by **Kanika Sharma (14MECE07)**, towards the partial fulfillment of the requirements for the degree of Master of Technology in Embedded Systems, Nirma University, Ahmedabad is the record of work carried out by her under our supervision and guidance. In our opinion, the submitted work has reached a level required for being accepted for examination.The results embodied in this major project, to the best of our knowledge,haven't been submitted to any other university or institution for award of any degree or diploma.

Date: Place: Ahmedabad

Prof. Dhaval Shah

Internal Guide

Program Coordinator

Dr. D.K.Kothari

Department Head, EC

Dr. Alka Mahajan

Director, IT

Certificate

This is to certify that the Major Project (Phase- I) entitled **"AMS Test Design and Automation"** submitted by **Kanika Sharma(14MECE07)**, towards the partial fulfillment of the requirements for the degree of Master of Technology in Embedded Systems, Nirma University, Ahmedabad is the record of work carried out by her under our supervision and guidance. In our opinion, the submitted work has reached a level required for being accepted for examination.

Chetan Hooli
Director, Validation and Automation
Atonarp Microsystems India Pvt. Ltd.
Bangalore

Acknowledgements

I would like to express my gratitude and sincere thanks to **Dr. Alka Mahajan**, Head of Electrical Engineering Department, and **Dr. N.P.Gajjar**, Professor and PG Coordinator of M.Tech Embedded Systems program for allowing me to undertake this thesis work and for his guidelines during the review process.

I take this opportunity to express my profound gratitude and deep regards to **Prof. Dhaval Shah**, guide of my major project for his exemplary guidance, monitoring and constant encouragement throughout the course of this thesis. The blessing, help and guidance given by him time to time shall carry me a long way in the journey of life on which I am about to embark.

I would take this opportunity to express a deep sense of gratitude to **Ms.Juhi Pabuwal**, Automation and Tool Engineer, Atonarp Microsystems Pvt. Ltd. for her cordial support, constant supervision as well as for providing valuable information regarding the project and guidance, which helped me in completing this task through various stages.I would also thank to **Mr. Chetan Hooli**,my Project Mentor for always helping,giving me good suggestions, solving my doubts and guide me to complete my project in better way.

I am obliged to **V. Balaji** Validation engineer of VAlidation and Automation team, Atonarp Microsystems India Pvt. Ltd. for the valuable information provided by him in respective fields. I am grateful for him cooperation during the period of my assignment.

Lastly, I thank almighty, my parents, brother and friends for their constant encouragement without which this assignment would not be possible.

- Kanika Sharma
14MECE07

Abstract

The history of Mass Spectroscopy can be traced back to the quest to study nature's physical and chemical components.A technique to chemically ionize an input sample and separate its constituents based on their respective mass to charge ratios forms the basis of Mass Spectroscopy. Whether it is quantitatively measuring the components of natural gas, achieving reliable gas compositions for fabrication industries, or monitoring the evaporation of water vapour during the freeze drying process in pharmaceutical industries, AMS is a smart device to handle all.

Training consisted of an in depth analysis of AMS features. This was followed by developing test cases for new features added and also manually testing them. Automating these test cases using python to reduce human intervention and effort in future provided a learning opportunity. Finally the process of Gas Calibration was done to increase the accuracy of the instrument.

AMS Validation Suite, a standalone tool is developed on python. the logs generated can be saved in the same directory or different one as per requirement. It checks the functionality of components, functionality of circuits consisting of the validated components, and the functionality of the board itself. It has test-cases to deploy firmwares onto micro-controller, run the functionality tests of components and circuits and board. After automation the focus was moved onto gas calibrating the instrument and its Mass Axis Correction features.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Changing technology landscape facilitates newer and better solutions to scientific problems. A client based, application specific mass spectrometer that provides highly accurate and precise measurements requires equally accurate control electronics and data algorithms. To function smoothly without any errors it has to be tested rigorously under development. To keep human intervention and its consequent errors at a minimal level, automation at the testing stage is required.

1.2 Objective

The main objective behind this project is to make testing and debugging more time efficient and error free.

1.3 Scope

This project is being used to automate the task of testing and validating new AMS software releases, which governs the functioning of Atonarp Mass Spectrometer (AMS).

1.4 Requirements

Completion of this project at Atonarp Microsystems, requires knowledge of linux/windows platforms, in depth hardware knowledge about ADCs/DACs, OPAMPS, basic of the physical and chemical nature of gases and their ionization. Automating the tests requires in-depth knowledge of Python language.

Chapter 2

Literature review

2.1 Spectrometers

Spectrometers are instruments that help in recording a spectrum. this spectrum can be linear or non-liner or log-linear variation of intensity with mass, energy, frequency or wavelength.

2.2 Types of Spectrometers

[1] Spectrometers can be classified into two broad divisions:

- Based on the quantity being measured
- Based on whether the ions are trapped or passed

2.2.1 Based on the physical quantity measured

- Optical spectrometer: In this, light intensity is shown as a function of frequency or wavelength by passing it through a prism or diffraction grating to obtain deflection.

- Mass spectrometers: In this the concentration of gas phase ions is measured at their respective mass-charge ratios. It is used for chemical analysis of a given sample.
- Time-of-flight spectrometers: In this two detectors are used and by measuring the time of flight of particles of known mass between these detectors their energy spectrum is developed.
- Magnetic Spectrometers: Based upon the whether alpha or beta particles are being measured there can two separate spectrometers for each of them.

2.2.2 Based on whether ions are trapped or passed through

Based on this fact the Spectrometers are further classified as Quadupole Mass filters and Quadrupole Ion Traps: [1]

- Quadrupole Mass Filters: These filters use use oscillating electrical and magnetic fields to influence the ion trajectories, passing through quadrupole field created between 4 parallel rods created by the application of RF voltages.Depending upon the voltages applied to the rods, ions of specific mass to charge ratio or within a specific range of mass to charge ratio can be selectively allowed to pass through and get collected on a collecting plate or cup.
- Quadrupole Ion Traps: Quadrupole ion trap is closely related to the quadrupole mass analyzer, and acts as a mass-selective filter. It does so by passing the untrapped ions. For this reason it is also referred to as a Transmission quadrupole.

2.2.3 Cylindrical Ion Trap

They differ from QIT only because their electrodes are flat shaped and not hyperbolic in shape. This helps in the size reduction of the device. As the size of a trap is reduced, in the region where the ions are trapped, the shape of the electric field near the center of the trap, is similar to hyperbolic trap electric field shape. [2]

2.2.4 Linear quadrupole ion trap

This is also similar to QIT and differs only in the fact that it traps ions in a two-dimensional quadrupole field.[2]

2.3 Using Python for Automation

- Human Readability matters. Code is read much more often than it is written. The coding conventions and standards of Python help write much human readable codes[3].
- Faster implementation You dont have to cook things in Python, things are ready to use and thats where we save the learning, design and implementation time[3].
- From logging classes to utilities to exception classes, things have already been defined[3].
- Standard OOP support[3].

2.4 Why OOP?

- Helps define abstractions, entities and relationships.
- Helps draw the entity-relationship models easily.
- Helps Reusability and Portability.
- Helps Easy Refactoring and Maintenance.

Chapter 3

Mass Spectrometers

3.1 Introduction

Mass Spectrometers are big and complex, mostly in research and developmental institutions for analytic purposes.Mass spectrometer is an apparatus which is used for measuring the masses of isotopes, molecules, and molecular fragments by ionizing them and determining their trajectories in electric and magnetic fields.

3.2 Three basic concepts involved

3.2.1 Mass Number

The Mass Number is the actual number of protons and neutrons present in the nucleus, it is also known as the Atomic Mass Number or the nucleon number.

3.2.2 Mass to Charge ratio

Although being a dimensionless quantity it is widely used in the field of electrodynamics and ion optics. Commonly denoted as (m/z) where m stands for atomic mass number and z for the charge number of the ion. It eases interpretation of data in mass spectroscopy as it is directly related to the unified atomic mass unit.most of the time it is equal to the atomic mass as generally a singly charged ion is preferred during the process. This is because while creating ions in one of the stages within a mass spectrometer, energy equal to the first ionization potential is applied.

3.2.3 Quadrupole Mass Analyser (Filter)

A quadrupole is a set of four rods with a space down the middle. The ions enter this space. The rods are electrically connected to each other in opposite pairs. A constant (DC) voltage and an alternating (AC) voltage are applied to the two pairs of electrodes.[2]

A small ion will be dragged a large distance by the alternating field. It will quickly collide with an electrode and disappear. A very large ion will not be affected much by the alternating field, but will gradually drift in the constant part of the field (the DC part). The alternating field is not strong enough to drag it back as it wanders, so it also collides with an electrode, and is lost.

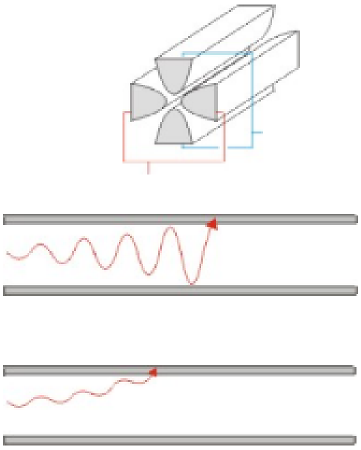
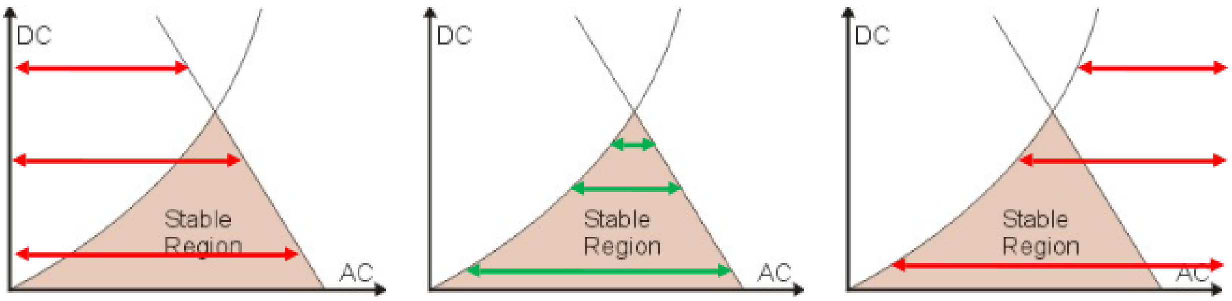


Figure 3.1: Deflection through the Quadrupoles

An ion that is the right size drifts slightly in the constant part of the field, but is always dragged back by the alternating part. The alternating part, however, is not quite strong enough to make it spiral out of control into an electrode. Thus an ion just the right size is stable in this quadrupole field and reaches the end, where it can be measured.

The stability of an ion in a quadrupole (its chance of making it through the quadrupole without wandering so far from the safe region in the middle that it hits an electrode and is lost) therefore depends on the sizes of the alternating and constant fields. It is possible to draw stability diagrams describing whether an ion is stable or not at any given pair of voltages, AC and DC.



Ion is stable in X for all DC, AC value pairs in this region

Ion is stable in both X and Y for all DC, AC value pairs in this region

Ion is stable in Y for all DC, AC value pairs in this region

Figure 3.2: Ion Stability Diagrams

In reality, the boundaries are not binary, and operation near the edges of the stability diagram results in lower ion transmission efficiency.

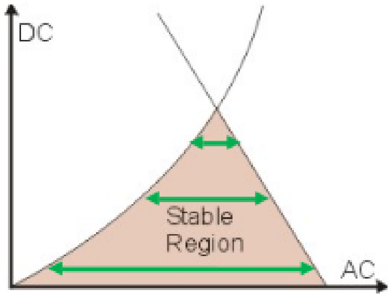


Figure 3.3: Actual stability region

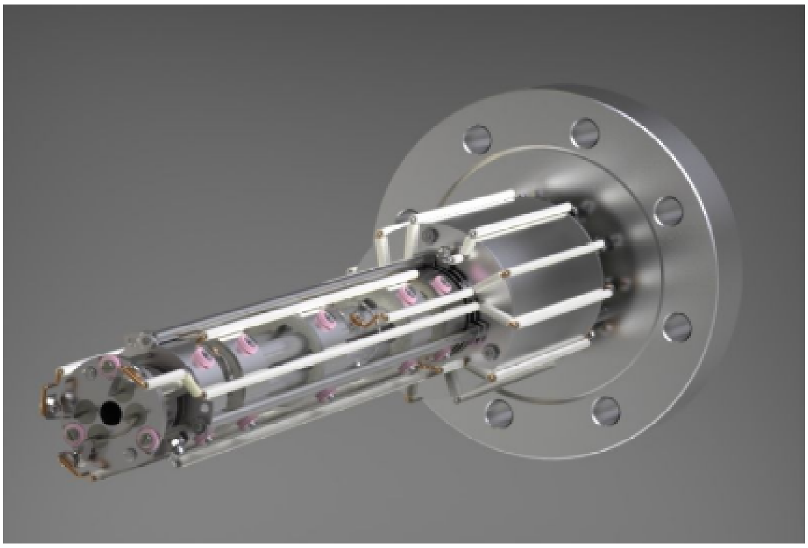


Figure 3.4: A Linear Quadrupole Filter

3.3 Working of a Mass Spectrometer

Atoms and molecules after being converted into an ion can be deflected by magnetic fields. Only charged particles are affected by a magnetic field, electrically neutral ones are not. Thus intuitively there are four steps involved in its working.

3.3.1 Ionization

Using an electron beam under vacuum conditions the atom or molecule is ionized by knocking off one or more electrons to give a positive ion. Generally the first ionization energy is utilized as the second would remove more than one electron, and modify the m/z ratio.

3.3.2 Acceleration

The ions are accelerated so that they all have the same kinetic energy and are able to enter the quadrupole field together.

3.3.3 Deflection

The ions are then deflected by a magnetic field according to their masses and the positive charges on them. The lighter the ion, the more it gets deflected. The more the ion is charged, the more it gets deflected.

3.3.4 Detection

The beam of ions passing through the machine is detected electrically. The ion detector senses the charge/voltage induced or the current produced when an ion grazes or deposits on a collector. Additionally, there is a need for a vacuum as the ions produced in the chamber should have a free run through the machine without hitting air molecules.The positive ions produced are further pushed out into the rest of machine by the ion repeller which is another metal plate carrying a slightly positive charge.Deflection of these positive charges is dependent almost entirely dependent upon the factor of mass to charge ratio having the unit Thompson (Th). Lighter ions need a lesser magnitude of magnetic field to be deflected heavier ones need a larger magnetic field. On varying the magnetic field each ion stream can be brought to the ion detector to produce a current that is proportional to the number of ions arriving.

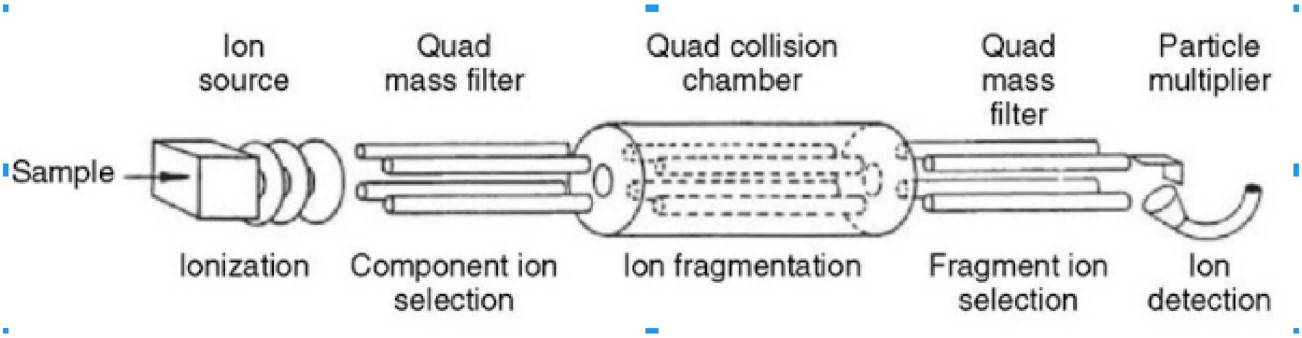


Figure 3.5: Working of mass spectrometer

3.4 Atonarp Mass Spectrometer

AMS being a smart device has few major differences with the conventional types:

- Reduced form factor.
- Customized for different users
- This product has the application, middleware and the algorithm software running on the SOC.
- The SOC then communicates with the microcontroller which commands the low level mass spectrometer hardware.

3.4.1 Hardware

It comprises of a filament which is maintained at very low pressure. A box that houses the filament. A lens which is used to colimate the cation beam, that further moves towards the quadrupoles. Quadrupoles are the area where the cations get deposited

3.4.2 Bias Voltages

Depending upon the voltages applied to the quadrupoles, cations of a particular gas get deposited on them. Quadrupole is followed by a detector, which transports the cation current generated for further analysis. All these components, namely filament, box, lens are provided with bias voltages. These bias voltages ensure that each component works efficiently

3.4.3 Working

- Voltages applied to the Quadrupoles have peak to peak values reaching upto 780V.
- Thus to accomodate the circuitry needed to generate all the bias and RF voltages the AMS assembly consists of two major boards: the Ion Drive and the RF drive.
- The Ion drive is responsible for generating all the bias voltages and the RF drive generates the RF voltages.
- The RF voltage generated is sine wave given by the expression (v+ucos wt), where u=(m/z)*k. This value of u changes for every different cation produced of a gas, hence ensuring selectivity. Here k is a constant, a ratio of the RF to DC voltage

Wherever required the voltages and currents are set and read using DACs/ADCs, eg: box, lens, filament voltages and currents. For measuring the current detected by the ion-detector a capacitor is used. For measuring higher values of currents and its fluctuations two capacitors are used which can be switched on and off depending upon the amount of current being measured

Chapter 4

AMS software, Testing and Automation

4.1 AMS Software Layers

AMS software is implemented with a layered architecture in order to abstract out the purpose, thus keeping the scope of each layer intact and facilitating parallel development. While the bottom most layer is more dependent on HW interface needs and capabilities, the top most layer is driven by the customer needs[4].

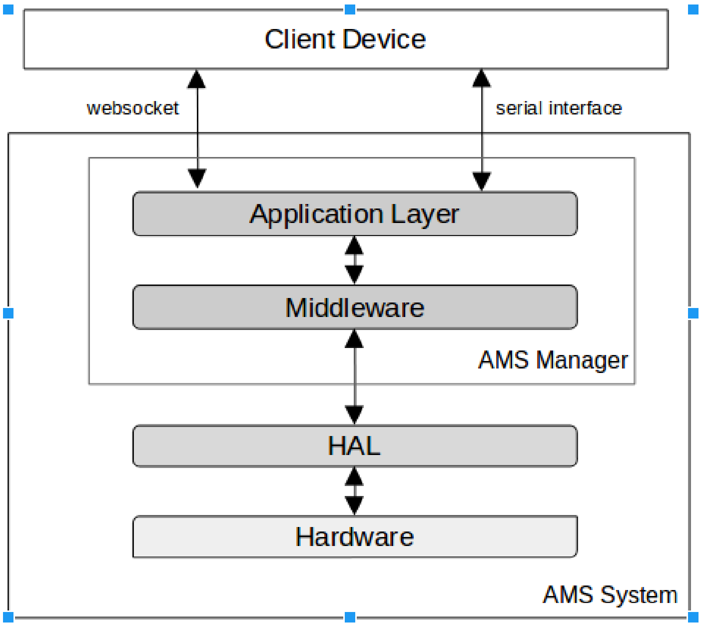


Figure 4.1: AMS System Layered Architecture

4.2 AMS Manager

AMS-Manager is the middleware and application layer entity that runs on SOC controlling and issuing commands to devices beneath it, and provides interface to applications above it.

It is also responsible for monitoring and regulating various device parameters and health of the system.So most of the higher level characterization can be performed through this layer and so does the validation[4]. It is a Tool used by Atonarp to run a scan over serial port to talk to the AMS-Hardware. The communication is performed between PC/SOC with the Micro-controller present on the AMS-Hardware.This communication is performed over a serial port, where commands are transmitted. These commands are later processed by various software layers which are flashed on micro-controller[4].

AMS-Manager requires an configuration file with ini format as input. Based on the input from the config file the AMS-Manager performs the actions. These inputs give information like the configuration of micro-controller, the range of m/zs to be scanned, or the rate at the scan data is required[4].

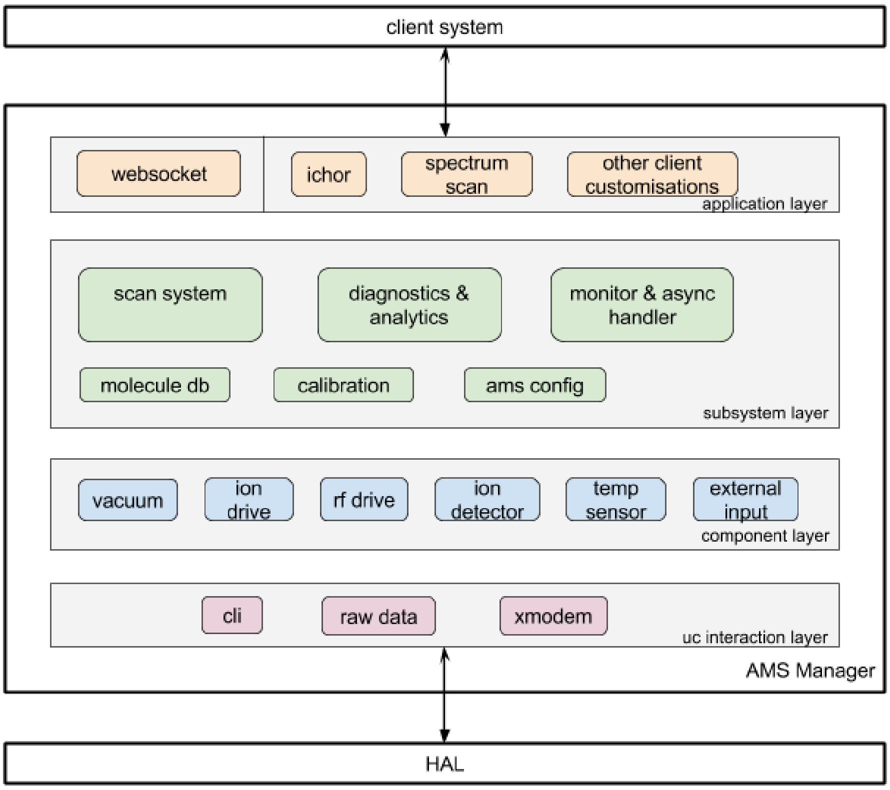


Figure 4.2: AMS System Layered Architecture

4.3 Software on Microcontroller

Software on Microcontroller has 2 layers[4]:

- System-Software
- HAL

System Software is like image flashed on the different partitions on a bootloader. This layer takes care of the drivers code required for driving the various components on AMS-Hardware. It even takes care of transmitting lower level notifications to upper layers like HAL or middle layer(AMS-Manager) HAL is a wrapper software around the system-software which takes care of higher level functionalities, like when executing a command given by AMS-Manager, it performs higher level functionalities by using the various underlying hardware. This layer is like a abstraction layer which lets only the required functionalities from drivers exposed to the the layers above

4.4 How the Atonarp Software Validation Suite works?

ASVS(Atonarp Software validation suite)[4] is a validation suite consisting of Automated test-case scripts to validate the Atonarp software. This validation suite checks the functionality of components, functionality of circuits consisting of the validated components, and check the functionality of the board itself. It has test-cases checking the bias voltages supplied on the boards, deploy firmwares onto micro-controller, run the functionality tests of components and circuits and board. The results are bundled up in a folder with csv, which contains test-data pertaining each tests and also the logs for each test.

The Atonarp Mass Spectrometer requires various boards to provide voltages to ionise the gas, drive the gas towards the quadrupole, to filter the ions passing through quadrupole, detect the ions which have passed through the quadrupole. We also have a board which puts together all of these boards and contains a microcontroller to communicate through SPI/I2C/UART protocols with components on the various boards. All of these boards get their voltages supplied from another board.

Features of ASVS:

- Easy to add test-cases with minimum number of entries in the framework
- Provides access all the python drivers for spi,i2c based components.

- The Test-cases can be easily configured through configuration files.
- The drivers for components and also signals from this components can be configured through excel based database.
- The test-cases which need run can be selected through configuration file.
- The order of execution of test cases can also be configured through configuration file
- The python code written is OS Agnostic as it can be run on windows, linux and other os which support python.
- Logs all sequences of the Validation suite, A master log contains logs of all the test-cases, and even logs for each test-case is also provide.

4.4.1 Main.py

Main module is starting point for the Atonarp Software Validation Suite, it does the following functions parse command line arguments Command line arguments used are –help/-h , this argument provides help functionality –logpath, path where logs need to be saved. –conf-files, path to the ASVS conf-files, The configuration files consists of Master config file which has information on all the test cases The other configuration file consists of Test Parameter config file which contains information on the parameters required for the test-cases.

- Log all the debug information to a file, creates a global logging functionality, such that all the debug information are a routed to a specified file in a specified location.
- Controls the BaseSuite module, The BaseSuite runs searches for all the test-cases provided in the configuration file.
- Initializes all the test cases to make sure everything is in place.

- Runs the test-cases one by one and de-initializes the test cases right after the run.

4.4.2 Base.py

The base.py contains all the base classes which can be derived later to provide any other required functionalities, the functions of base module is as follows.
Basesuite module controls all the test-cases which need to be run

- The BaseSuite runs searches for all the test-cases provided in the configuration file.
- Initializes all the test cases to make sure everything is in place.
- Runs the test-cases one by one and de-initializes the test cases right after the run.

BaseParam handles all the signals entering or exiting a component. This module takes the get or set param as an argument.

- Searches for the component which handles this get or set signal.
- Gets the required information to initialize the drivers for the components.
- When a get function is called it provides the signal value
- When a set function is called it sets the provided value.

BaseHardware handles all the drivers required for the ASVS Module

- The basehardware initializes the FT2232H drivers
- Initializes the Instrument drivers when required.
- Start the session for lan or uart connection to talk to device.

BaseTest is base class for all the test-cases of all the test-cases. It has basic test which are required for all the test-cases,like

- Test two array similarities
- Test a value is within a tolerance limit.
- Test a condition
- Test i2C device scan.

4.4.3 Config.py

config.py Config module is an input medium for the ASVS, It parses the config files and provide it as a machine understandable format. The config module contains a dictionary which contains the section and option names, its type and its default values in case the values are missing in the configuration file.

- The config module also creates dict for each configuration file and updates the value based on the user's input.
- The config modules is dependent on a python package config parser.
- Using the configparser python package the configuration files are parsed and its value are read.
- The read values are converted to their respective types, in case the type is not mentioned it uses string type.
- If the value is itself not present the default values are chosen to fill the dictionary.
- The config module is written in a scalable format so that in future in can be used for other Automation purposes.

4.4.4 Testresults.py

The test results module is an output for the ASVS, where the functionality to dump the result is found. Test results stores the results in the following format:

- Test-name: Name of the test-cases.
- Result: The test is a PASS/FAIL
- Measurement: The deviation from the expected data.
- Description: Brief description of the test-case
- Test-Data: The path where the test data is dumped for this test-case.

Test results also dumps the test-data which is in machine understandable format to a csv file which can later be opened in excel to see the test-data. This module also takes case of Displaying the final result on the screen. The information displayed are test-name, result, and Description.

The final result of all the test-cases is also logged in a csv format. So that the information the test-cases can also be seen or analysed easily.

4.4.5 Provider.py

- The provider contains functionality to wrap/abstract all the input output objects into one object.
- The provider module abstracts all the input/output module object in it.
- It also has functionality to run a command on the host machine.
- It provider module consists of Log class which is used to log all the events in one master log file.

- It Log class also as functionality to print the logs in color, and also to various files.

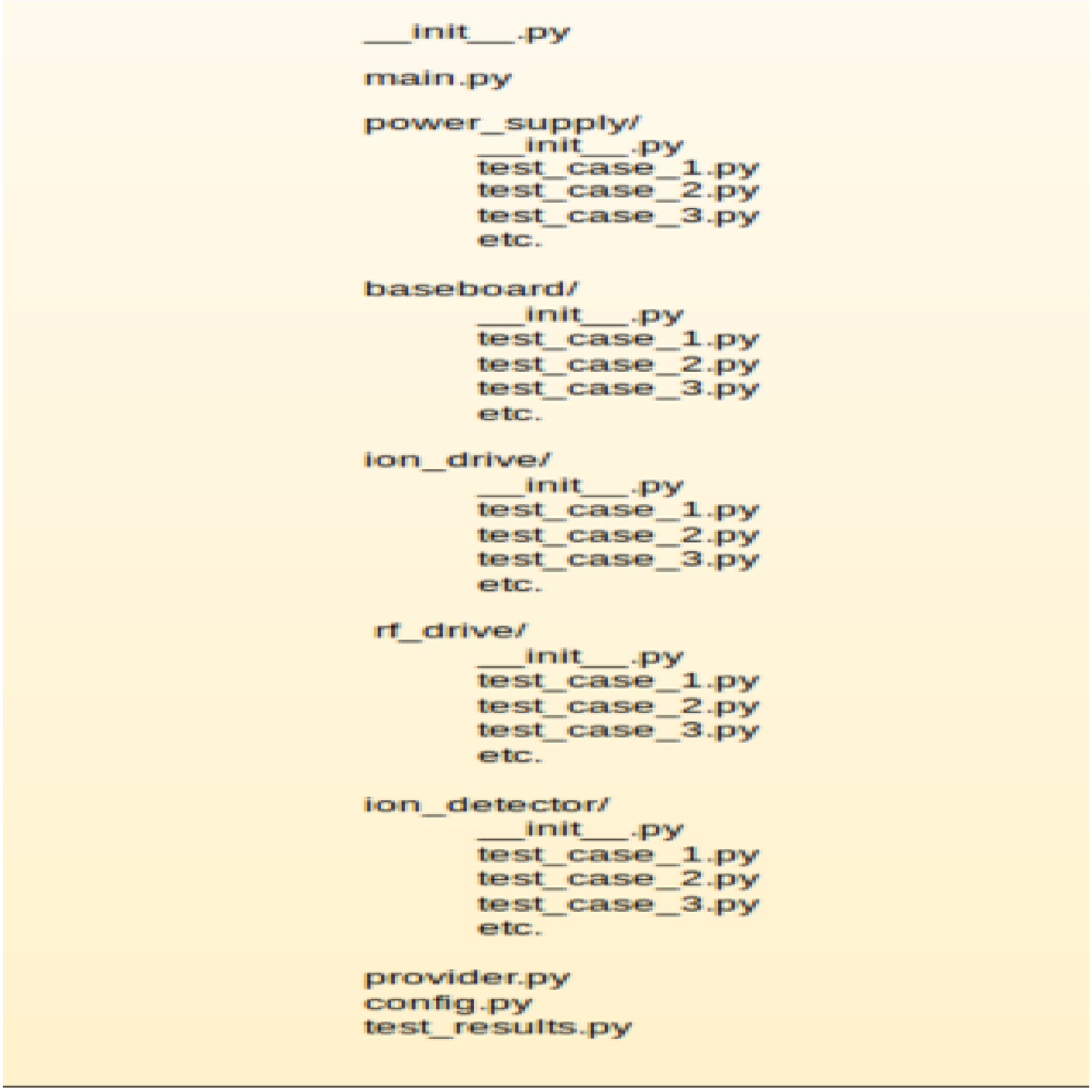
4.4.6 Board class

Each board has a class in its name in ASVS 'init' module which will is derived from the BaseTest class. This module consists of all the signals as its attributes, so that the signals can be used to set or get its value. Its also contains tests or functionality which is common to that specific board.

4.4.7 Test-case Class

The test-class is derived from its board class which was explained in the previous topic. The test-class thus has access to all the drivers, Input, Output, board specific functionalities. The test-case class consists of init function to initialise the test, run function to run the test and deinit function to deinit the test.

4.5 An overview of the Validation suite



```
__init__.py

main.py

power_supply/
          __init__.py
          test_case_1.py
          test_case_2.py
          test_case_3.py
          etc.

baseboard/
          __init__.py
          test_case_1.py
          test_case_2.py
          test_case_3.py
          etc.

ion_drive/
          __init__.py
          test_case_1.py
          test_case_2.py
          test_case_3.py
          etc.

 rf_drive/
          __init__.py
          test_case_1.py
          test_case_2.py
          test_case_3.py
          etc.

ion_detector/
          __init__.py
          test_case_1.py
          test_case_2.py
          test_case_3.py
          etc.

provider.py
config.py
test_results.py
```

Figure 4.3: Atonarp Software Validation Suite

Chapter 5

HAL state machine

Atonarp HAL is part of the system software bundle to provide a uniform view of various features provided by the AMS-P1000 hardware. It is the thin layer placed between various device drivers and the application. Device driver facilitates accessing various boards present in the system which includes ION detector, ION drive, Pressure sensor, RF drive and Power supply. HAL abstracts out various device drivers and provides uniform view of various features provided by these boards. HAL also plays a vital role to ensure these boards shall operate on wide temperature ranges.

5.1 Architecture

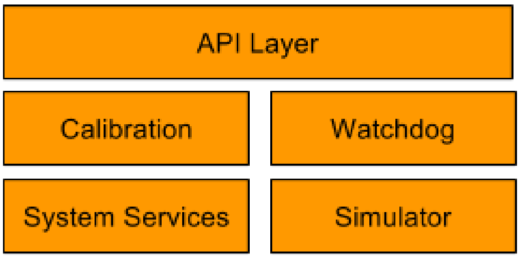


Figure 5.1: HAL Building Blocks

Figure above shows the basic building blocks of Atonarp HAL, these blocks cater both third party exposable and intellectual properties of AMS-P1000 system. API Layer provides set of interfaces to program the AMSP-1000 features to third party firmware developers. These are generic and at the same time powerful to use all features provided by the AMS-P1000 hardware. API layer intern makes use of all other blocks to realize the required functionalities.

Calibration layer is tightly linked with API layer. API Layer exposes voltages and currents as per the sensor/analyzer specification. It is the responsibility of the calibration layer to translate these to digital domain or vice versa. It also takes care of board diversity, temperature adjustments, data independence, transfer functions and integrity management.

Primary responsibility of System Services is to extend the functionalities to the CPU board. CPU board is equipped with powerful Qualcomm 801 hardware with high processing power and hardware security features. This enables one to develop very complex extensions without worrying about the computing power/security. Currently System Services include a command line interface over UART to perform various tasks including firmware upgrade.

5.2 API Layer

Figure below shows the list of APIs, its relationship with hardware, any restrictions and software only interfaces. APIs are broadly classified as non restricted and restricted. Non restricted APIs can be called anytime and at the same time restricted APIs are associated with AHAL state machine. State machines are covered in the Watchdog section. Currently majority of the APIs belong to non-restricted bucket which helps upper layer play around with APIs and define the most suitable sequence. Once the sequence is finalised the same should be moved to restricted

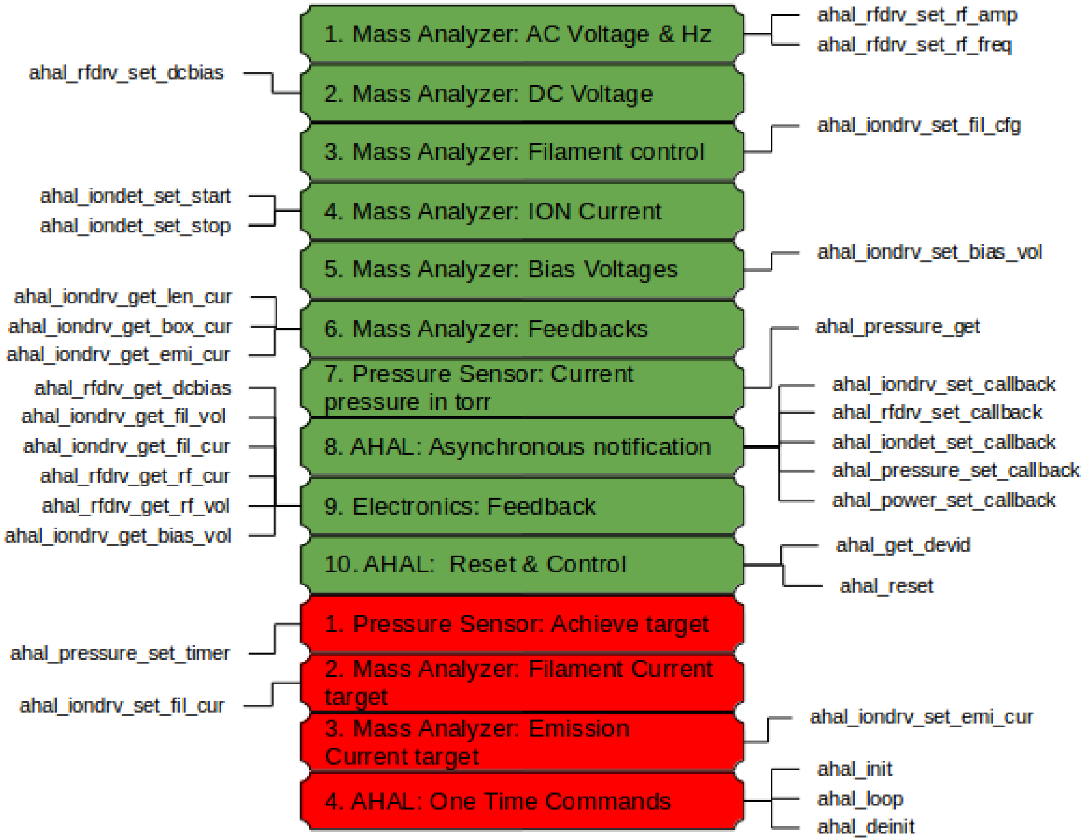


Figure 5.2: HAL APIs

bucket. Another important point is, all APIs associated with hardware shall follow the specification of the hardware. This greatly helps third party developers to build/customize the firmware as per their need without any additional documentation.

5.3 Calibration

Calibration block performs two major functionalities: integrate the sensors/analyzers with Atonarp electronics and bring precision and stability. Figure 5.3 shows major blocks in the calibration subsystem, it is splitted into two independent sub blocks: code (transfer functions) and data (calibration table). Code is part of the AHAL library and data is present in the respective board EEPROM, this enables seamless update of calibration data without touching the firmware.

Calibration data is present in three different locations: Respective board EEPROM, microcontroller EEPROM and LPC1769 internal flash memory (to speedup). During system startup AHAL checks for the integrity of calibration binary tables in the internal flash, any error results in recreation of cache that can be either from step-1 (First priority) or step-2 (Second priority). During this time an additional lookup is also created to map the board components to corresponding transfer functions mentioned in the calibration binary table. This brings additional flexibility to change the transfer function of a given component later point of time.

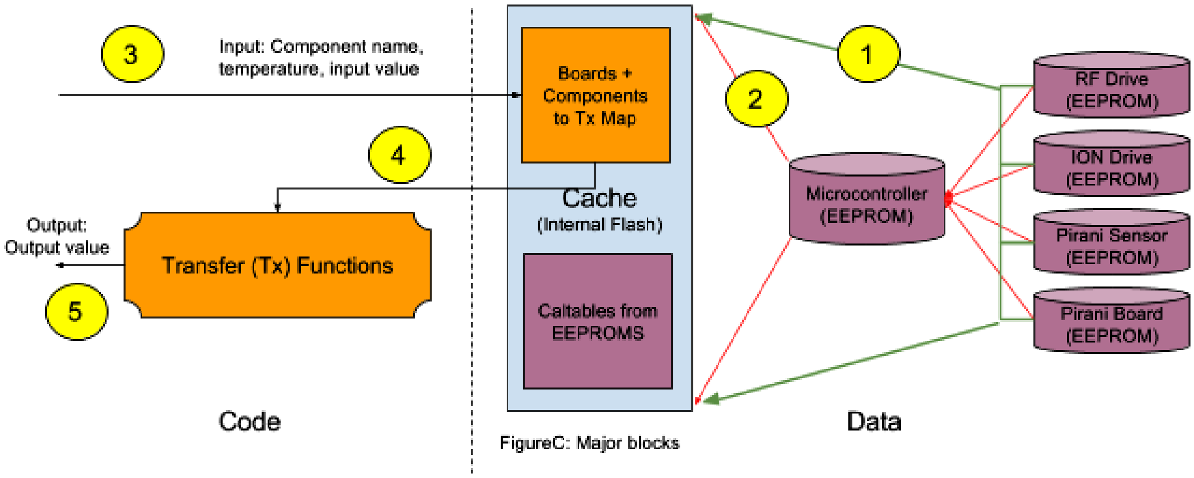


Figure 5.3: How it works

How it works, suppose AHAL wants to translate DC voltage to DAC voltage, perform step-3: first do lookup of the transfer function from the cache, step-4: execute the transfer function and step-5: transfer function computes the DAC value from the calb table and same is given back to AHAL for programming a DAC channel.

5.4 System Services

Primary responsibility of System Services is to extend the microcontroller features to the CPU board which has high processing power and cutting edge security. Currently three services are exposed, Firmware Upgrade, Environment Variables (set and get) and ION current estimation.

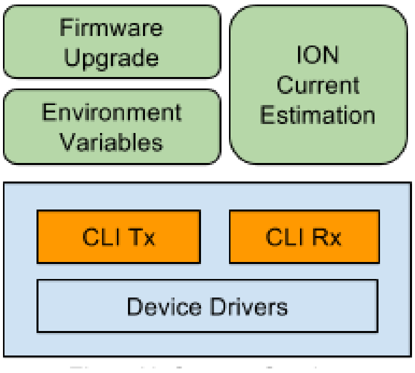


Figure 5.4: System Services

The software part of the CPU board makes use of these services to implement desired functionality. All these services are implemented on top of two independent execution sequences, CLI Tx and CLI Rx, which are responsible for sending and receiving the data.

Chapter 6

Gas Calibration

When a gas blend is introduced to the AMS, the gas molecules are first ionized, then filtered for specific mass-to-charge (m/z) ratios of the ions, and then detected. The primary output of the AMS is a spectrum that relates detected ion current intensity to m/z ratio for all ions that were filtered. By modeling the gases from which the ions are generated, and by modeling the instrument characteristics, the AMS analytical algorithm derives the gas blend composition from the mass spectrum. The AMS software user interface has been designed with the flexibility to configure many of the instrument properties. This allows an advanced user to tune the instrument to optimize its functionality and performance. We calibrate the instrument to identify optimal values for its configuration parameters. The core of the calibration effort is the modeling of the gas and instrument characteristics, represented as matrices, which are input into the AMS software to accurately estimate the composition of the gas blend introduced to the AMS. To perform the calibration, the AMS is introduced to various known gas blends and mass spectrums obtained. The spectrum is then fed to an algorithm that generates the calibration matrices. Each assembled AMS system is calibrated and tested with pure gases and gas blends that are not limited to but include the application's analytes of interest.

6.1 Gas delivery

The calibration effort requires various gas delivery plumbing setups depending on instrument accuracy requirements and the specific calibration step that needs to be performed. Frequently used physical setups are described and required parts identified.

6.1.1 Pure gas and pre-mixed precision blends

The gas vendors offer various grades of purity and blend accuracy. The selection of the grade is dependent on the accuracy required in the application where the AMS is to be deployed. Pre-mixed precision gas blends are required for calibration when instrument accuracy error requirement is less than 2 percent.Pure gases and pre-mixed blends are delivered in cylinders filled at high pressures of more than 500 Psi (34 Bar).The role of the delivery system is to drop the pressure from more than 500 Psi to a value desired in the AMS vacuum chamber, nominally 1 mtorr.The desired pure gas or precision blend from the cylinder is stepped down to 2 bar using a pressure regulator. With the turbo and roughing pump operating at maximum RPM, metering valve 1 is adjusted until the capacitance manometer indicates the pressure desired in the vacuum chamber. Note that metering valve 1 may need to be partially closed for the turbo pump to achieve the maximum RPM. A capacitance manometer is used to measure chamber pressure accurately. The ionization gauge is optionally used to zero the capacitance manometer since it is very sensitive to orientation in relationship to gravity.

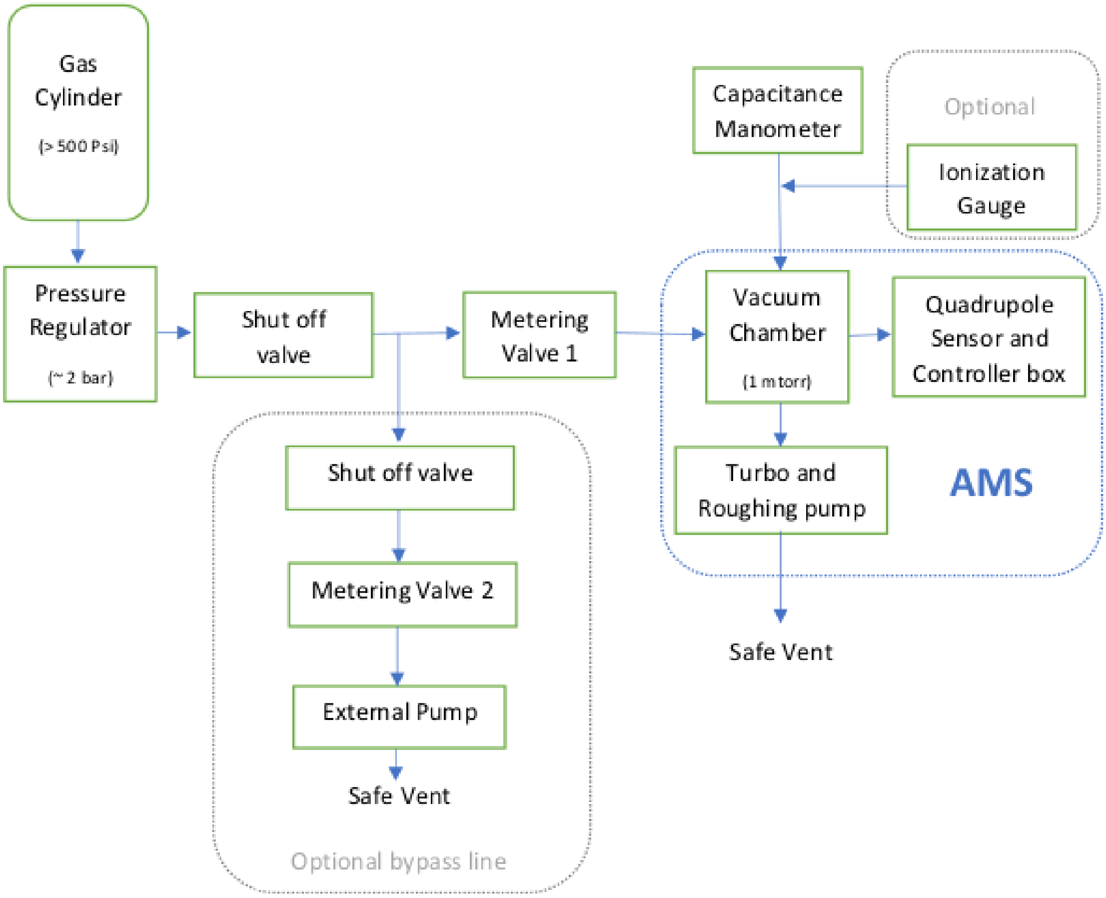


Figure 6.1: Pure Gas and Pre-mixed Blends

6.1.2 Gas blending with MFCs

A low-cost low-accuracy gas blending system can be setup in-house using mass flow controllers (MFC). This is useful to calibrate the AMS for low accuracy applications (more than 2 percent accuracy error) or for a quick and dirty calibration when high accuracy pre-mixed precision blends are not available. Three pure gases from three distinct cylinders are regulated and sent to an MFC. The outputs of the MFC are blended using a cross and then fed to the AMS via metering valve 1. The design can be scaled for more than three gases by connecting more cylinders and regulators via more tees and crosses. With the turbo and roughing pump operating at maximum

RPM, metering valve 1 is adjusted until the capacitance manometer indicates the pressure desired in the vacuum chamber. Note that metering valve 1 may need to be partially closed for the turbo pump to achieve the maximum RPM. If the metering valve at almost closed position is not able to achieve a satisfactory low pressure set point, the bypass line can be used. The external pump connected to the main gas line via a shut-off and metering valve can step down the pressure before the generated blend is introduced to the AMS. Metering valve 2 can be adjusted to control the extent of pressure step down. A shut off valve is used to close the bypass line if needed. This is essential for purging. A capacitance manometer is used to measure chamber pressure accurately. The ionization gauge is optionally used to zero the capacitance manometer since it is very sensitive to orientation in relationship to gravity. (Refer to the figure on Pg 36)

The blending principle is that for ideal gases, the ratio of volumetric flow rates of gases constituting a blend is equal to the ratio of the mole fractions of the gases in the blend. With the vacuum chamber heated to high temperature (120 degrees C) and operated at low pressure (1 mtorr), gases behave close to ideal. A desired composition can be obtained by setting the MFC flow rates in the same ratio as the desired composition of the constituent gases.

6.1.3 Pressure regulation

All calibration experiments are best performed with chamber pressure tightly controlled to a desired set point. A capacitance manometer is specifically chosen as a pressure gauge because its readings are unaffected by the nature of the gases in the blend, unlike a pirani or an ionization gauge. The capacitance manometer can also provide pressure measurement at much greater accuracy than other pressure gauge types. A pressure control loop is required that reads the pressure from the capacitance manometer and regulates pressure by throttling either the turbo pump RPM or by adjusting a proportional valve.

6.1.4 Purging

Whether calibrating/testing using pure gases, pre-mixed precision blends or generating blends with MFCs, it is essential that the blend introduced to the AMS is not contaminated with residual gas from the previous calibration or test.
Purge steps: (refer to the figure on Pg 37)

- Power off the AMS filament
- Close the bypass line shut-off valve
- Introduce desired gas blend into the AMS chamber with metering valve 1 fully open
- Set turbo pump RPM to 0 and wait for five minutes. This should flush the chamber with the new gas blend at high pressure. Meanwhile a vacuum forms in the bypass line between the external pump and the shut-off valve.
- Now open the bypass line and allow it to be flushed with the new gas blend of interest.
- After two minutes, increase turbo pump to 90,000 RPM and throttle metering valve 1 and 2 to achieve desired pressure in the AMS chamber.
- Power on the turbo pump pressure control loop and perform further calibration and test.

6.2 Calibration preparation

The effectiveness of calibration to model the gas and instrument characteristics and the subsequent effectiveness of the analytical algorithm to estimate composition is contingent on certain pre-requisites and preparation.

6.2.1 Resonant Frequency identification

The AMS user interface provides an RF resonant frequency search tool in its calibration tab. At an RF amplitude of 140 V, sweep between 10.8 and 11.2 MHz to identify the resonant frequency. This test is best performed with the sensor intended to be shipped with the instrument connected. Record the resonant frequency.

6.2.2 Nominal configuration settings and baseline scan

After the AMS system is first assembled, some of its properties are set via software to nominal values, eg.: Box bias voltage to 70V, lens biaas voltage to 55V, DC bias to 57.5, RF to DC ratio to 6.2 etc. Once configured, the AMS chamber pressure is set to 1 mtorr and air scan is done. The expectation is that the peaks corresponding to components of Air Nitrogen (14 and 28 Da/e), Oxygen (16 and 32 Da/e) and Argon (40 Da/e) are clearly visible in the spectrum. If the peaks have any unexpected artifacts, these configuration parameters are manually adjusted and scans rerun until distinct peaks are visible. If the m/z resolution is found to be insufficient, the RF to DC ratio may be adjusted. The end-point for this manual correction is subjective and is a starting point for subsequent calibration and fine optimization. Expected reference is thus: (Refer to the figure on Pg 38)

6.3 Ion current detection noise floor

The lowest constituent of a gas (lower detection limit) to be measured in a gas blend should produce an ion current greater than the inherent noise in the ion detection circuitry at the nominal operating pressure of the vacuum chamber. The inherent noise in the detection circuitry could be from the electronics or the quadrupole sensor. The quadrupole sensor used in the AMS defines a Base value that reflects the lowest ion current it can measure. Quadrupole sensor must be chosen such that its Base value meets the applications lower detection limit. The relationship between Base value of the sensor and the lower detection limit requires a characterization study. Since the ion current detected is a function of pressure, this characterization is best performed at the nominal operating vacuum chamber pressure.

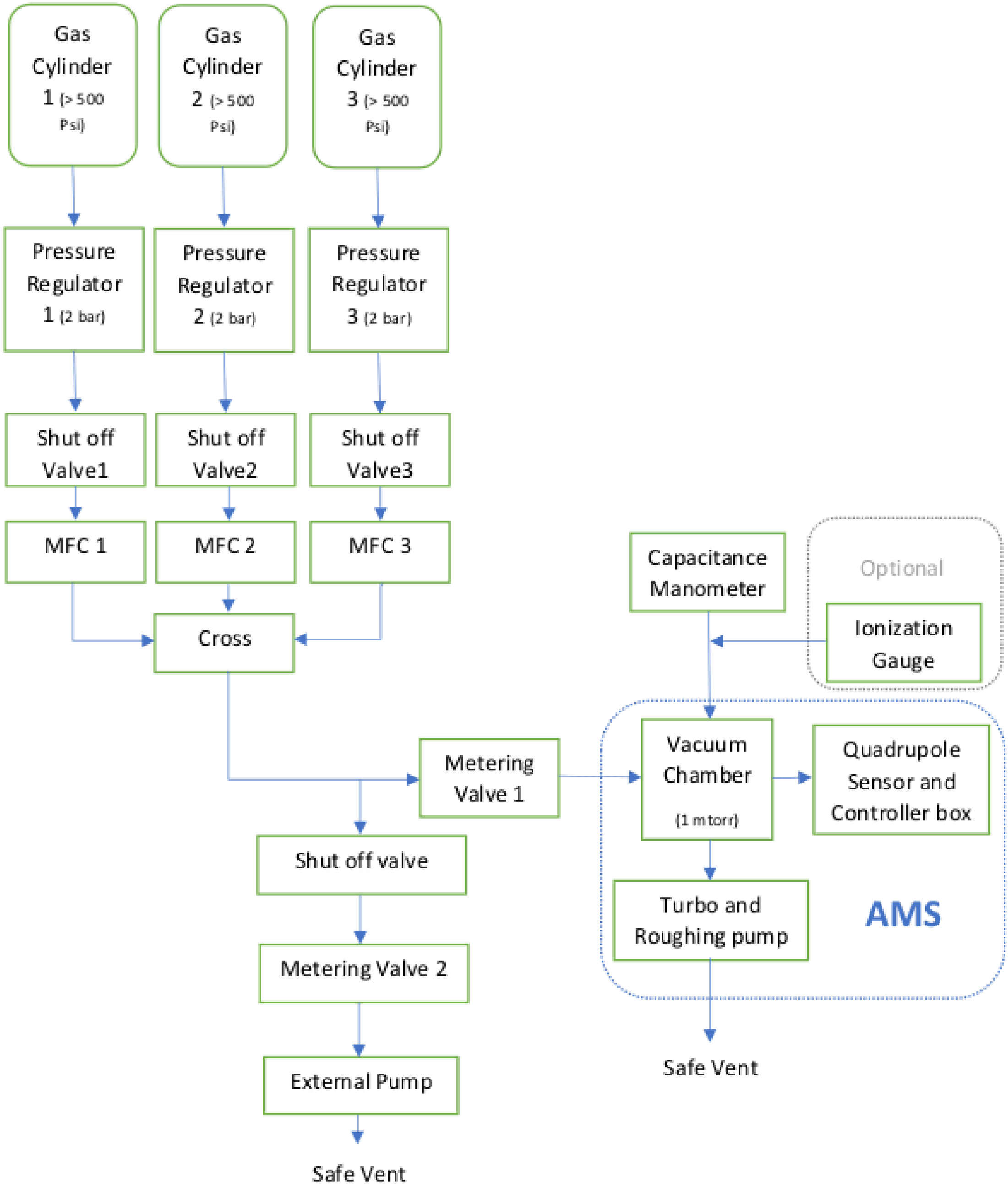


Figure 6.2: Gas Blending using MFCs

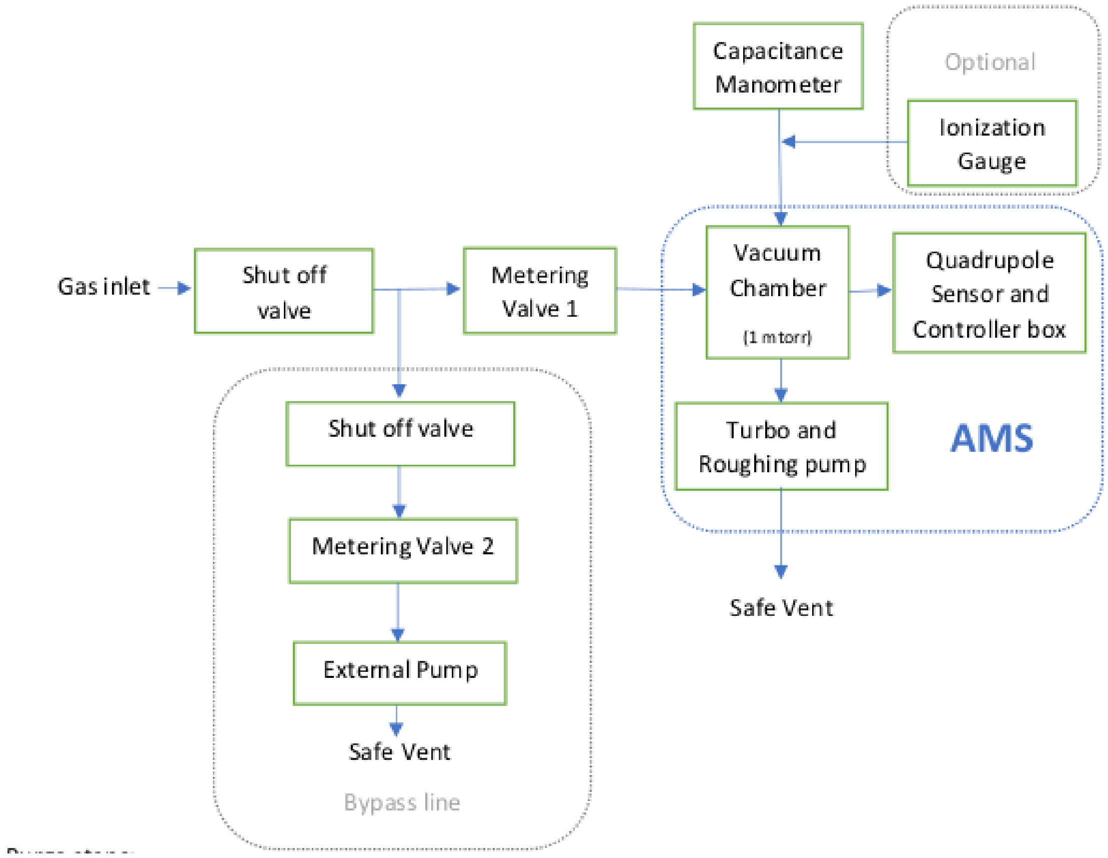


Figure 6.3: Purging Mechanism

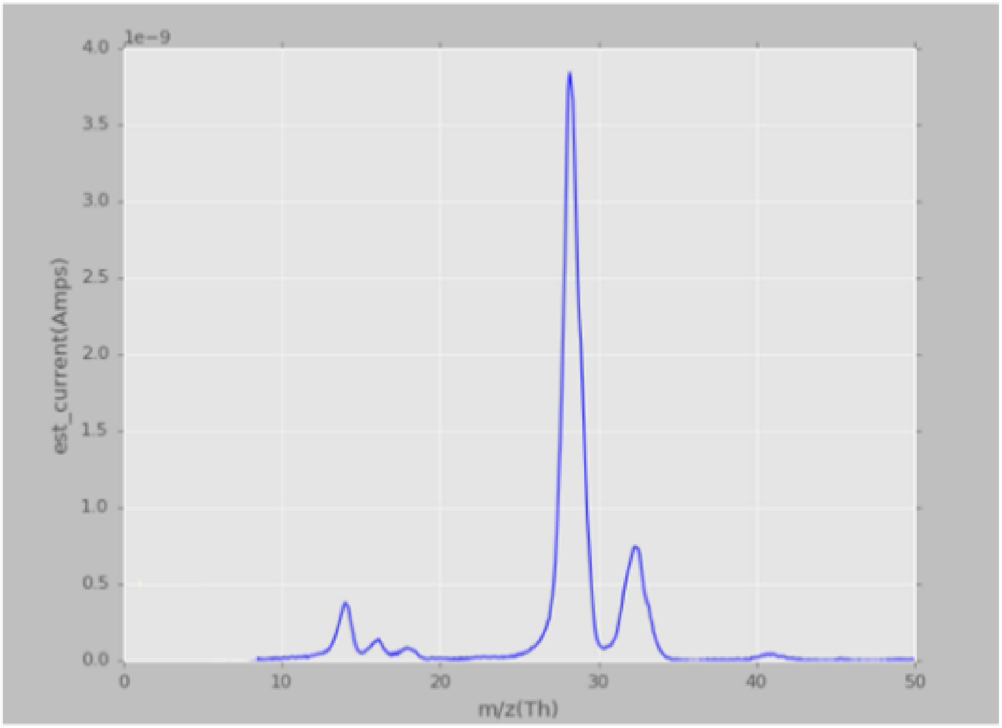


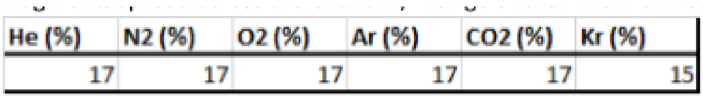
Figure 6.4: Expected scan graph

Chapter 7

Mass axis error and correction

All gas molecules have a well-defined mass and so a mass spectrum should show the ion current intensities at the expected mass-to-charge value for base peaks, molecule fragments, ion charge fragments and isotopic fragments. Due to the analog nature of the RF circuitry and limitations in hardware calibration, it is possible that the peaks in the mass spectrum are not centered at the expected m/z value. This can cause inaccuracies in composition estimation, particularly if there are ion peaks from different gas molecules close to one another on the m/z scale. This delta between where the ion current peak centers are expected and where they are observed is termed as mass-axis error. Mass axis errors can be coarse corrected either using the pre-scan mass axis correction algorithm (SW implementation in progress) or can be manually corrected by adjusting mass axis correction coefficients. These coefficients tweak the RF amplitude to force ion peak centers at desired m/z values.
The procedure for coarse correction is performed thus:

- Introduce a mass axis calibration gas to the Instrument. An ideal one is that which has many fragments spread across the entire m/z range of the AMS. For instance in the absence of a custom blend spread across the m/z range, air may also be used for mass axis calibration.

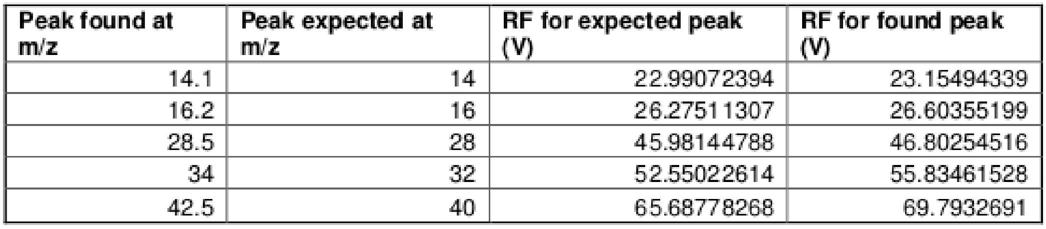


He (%)	N2 (%)	O2 (%)	Ar (%)	CO2 (%)	Kr (%)
17	17	17	17	17	15

Figure 7.1: Air scan mole fraction values

However, since components of air dont exceed a mass of 44 Da/e, any correction is unlikely to fix mass axis errors at higher m/zs when heavier gas molecules are introduced to the AMS

- Perform a scan of the mass axis calibrating gas and identify the m/zs at which peaks are observed. Identify where the peaks are expected. Calculate the RF amplitude corresponding to both the observed and the expected peaks using the formula:
  Vrf = K*(m/z), where K = (qx*f*f*r0*r0)/(4*e)
  Here,
  qx=0.706 * 1.660538782E-27
  f = RF frequency. This is ideally the resonant frequency, nominally 11 MHz converted to (radians/sec)
- Tabulate the m/z values and the RF amplitude.



Peak found at m/z	Peak expected at m/z	RF for expected peak (V)	RF for found peak (V)
14.1	14	22.99072394	23.15494339
16.2	16	26.27511307	26.60355199
28.5	28	45.98144788	46.80254516
34	32	52.55022614	55.83461528
42.5	40	65.68778268	69.7932691

Figure 7.2: Tabulated m/z and RF amplitude values

- Perform a scatter plot between the two RF amplitude columns, curve fit a polynomial that produces an RMS value close to 1.

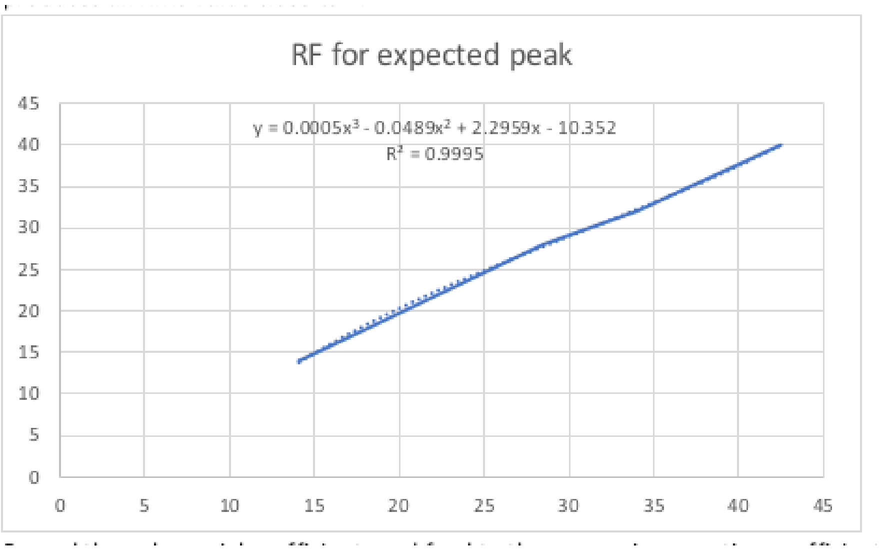


Figure 7.3: A scatter plot between two RF amplitude columns

- Record the polynomial coefficients and feed to the mass axis correction coefficient settings in the AMS UI ¿ AMS configuration ¿ Mass filter. Subsequent scans should show a spectrum with peak centers aligned correctly with expected mass axis values.

Chapter 8

Conclusion

For compacting an instrument as big and comprehensive as the mass spectrometer into a handheld size and to customize it for user needs, a highly integrated chipscale analytical instrumentation approach has to be adopted. Further to maintain the software layer abstraction, automation in python should be based on abstractions, entities and relationships. By automating the test steps for AMS software the manual testing which went on for a period of 5 days was done in 30 minutes. Also accuracy of results was increased by Gas Calibrating the instrument for a specific mixture of gases.

Chapter 9

Future Scope

Till now the aim of this project was to understand the basic functioning of the Atonarp Mass Spectrometer so that an appropriate testing framework can be developed. This testing framework was first manually tested and verified and then later on was automated to reduce human intervention and time. Manual testing and debugging the system that lasted for around a week if done manually is now being done within hours.

Future scope would comprise of running two parallel threads to capture debug logs from the system as well as to carry on testing simultaneously.

Bibliography

[1] E. MARCH RAYMOND and JOHN F. J. TODD, *QUADRUPOLE IONTRAP MASSSPECTROMETRY* The John Wiley and Sons, Inc.

[2] "LINEAR QUADRUPOLES IN MASS SPECTROMETRY ",D.J. Douglas Department of Chemistry, *UniversityofBritishColumbia*, 2036*MainMall*, *Vancouver*, *BC*, *CanadaV*6*T*1*Z*1

[3] Tutorialspoint *Python Programming language* Simple and easy learning

[4] Atonarp internal documents