Acknowledgement

My internship was a great success thanks to **Mr. C Rajendra** Sir, Research and Development Manager of Sertel Electronics; **Mr. Anand Sir** and **Mr. Thanigai Sir**, R&D Department staff. Throughout the course of the internship all the above-mentioned people helped me through various aspects of the work which includes conceptual issues, hardware issues and software issues. Their expertise and experience helped me open my knowledge to new boundaries, learn work ethics and hierarchy, look at things in a new way to solve work problems.

Abstract

This report highlights three main projects I did during the duration of internship.

First was a GPS tracker using a microcontroller called FiPy from Pycom and Pytrack an expansion board that uploads latitude and longitude data to adafruit.io cloud service using MQTT protocol and then making a receiver end GUI using Tkinter (Python library) that will get the value of the latitude and longitude coordinates previously uploaded from cloud and display it for user. Along with the GUI a web browser automation software was made with selenium which when run would automatically open our PC default web browser and display the location of GPS tracker and on termination of tracker would plot the entire course of the tracker.

Second was Face detection using Raspberry Pi 3, Open CV (Python Library) and Raspcam. Here an OS compatible with Raspberry Pi 3 was installed using bootable pen drive. Then using Raspcam, live video feed was got, on which processing was done to mark faces with a rectangular box in real time and display on a monitor.

Third was a GUI for ground-based sensor device using JAVA FX. Here various java libraries were used to get serial input from USB, then process the data got from the sensors through the serial port on the computer and display these values on GUI. This GUI was a part of their ongoing project related to weather forecast systems where they wanted to cross reference their ground sensor unit data with the data's got from the small sensor units sent high up the atmosphere using balloons. The GUI I made was to display this ground unit data.

Table of contents

1. Overview of the Organization

1.1 ABOUT US:

Sertel Electronics is an equipment manufacturing company based in UK (Birmingham) and India (Chennai). This is a technology-oriented company that focus on R&D and High-tech Industrial Applications. They are the world's leading manufacturers of highly precise time stamping technologies, solutions for industrial instrumentations, etc. Products include GPS based Time Synchronization System, Alarm Annunciators, Time Code Displays, Converters, Wireless, IoT Solutions for Smart Grids

1.2 HISTORY:

Started in 1977 by a team of Professional Electronic / Electrical Engineers to service Electronics Equipment's on Commercial and Industrial devices. They also have the distinct honor of being the first Indian company to manufacture GPS based Time management systems. They have grown as a unit ever since then to become the market leaders in India.

1.3 PRODUCTS AND PARTNERS:

They tailor their own solutions for industry verticals such as power sectors, chemical industries, Oil & Gas, Telecom, Defense sector, civilian networks etc. They support customers with solutions in different parts of the world to monitor, generate, distribute and apply time. Sertel strictly caters products to customer's industry requirements. Technology partners include SACO controls, Industrial Television, Quartz Lock, VJAS.

2. Plan of internship program

In Sertel Electronics, I was assigned to the Research and Development Department under the esteemed guidance of Mr. C Rajendra Sir.

About R&D department:

They currently have State-of-Art facility and wonderful R&D environment from niche developments at the premises. They have Invested on plant and machinery over the years for R&D activity. Their skill sets range from Embedded ARM processors, Wi-Fi Wireless technology, Microcontroller technologies, Protocol Stack development. Hardware & software development, prototyping to production design & testing. They also enable Extended Services such as Production automation & Research for Cost-cutting of existing product prototype designs & enable low cost production.

Start Date of Internship: 25-05-2019
End Date of Internship: 08-07-2019
Working days include Monday, Tuesday, Wednesday, Thursday, Friday and odd Saturdays
Time: 9.30 AM to 6.30 AM.

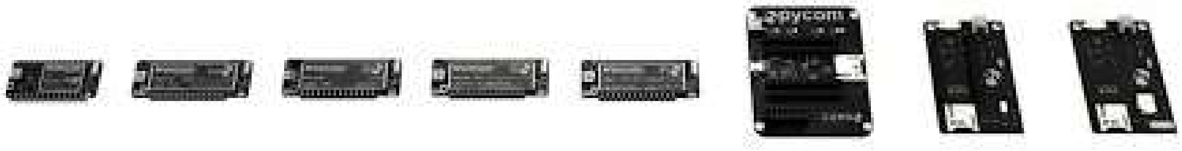
3. Training program

3.1 Duties and responsibilities performed:

From the start of my internship till the end I was kept busy with three projects covering IoT, Embedded Systems, Software Programming, etc.

On the day my internship started, based on the projects I had mentioned on my resume, I was first asked to make an IoT project with Node MCU that has an ESP32 processor and Wi-Fi capabilities for internet access. The project was very simple, to simply send five digital and one analog data got from the control unit of their very high temperature monitoring system to ThingSpeak server and store these data on their cloud services. I was able to finish the given project within a days' time. The project included working code written in C, with pin mapping, connecting the Node MCU to Wi-Fi credentials provided, connect to ThingSpeak cloud using API or MQTT protocol and transmitting data. This project was given to me to check if I knew the basics.

Then next I was given FiPy, a microcontroller manufactured by Pycom to work on. This microcontroller also had an ESP32 processor which includes – 4MB RAM, 8MB flash, 32 bit system, 600DMIPS processor speed and MicroPython enabled; has up to 22 GPIO pins and 2 x UART, 2 x SPI, I2C, micro SD card interface. But the major difference between the Node MCU and FiPy was that FiPy not only supported Wi-Fi network but also supported various other networks like dual LTE-M (CAT M1 and NB1), Bluetooth, LoRa and Sigfox. Along with the microcontroller, I was given three development boards namely Expansion board 2.0, Pytrack and Pysense. All these boards have their own use. The expansion board is useful to interface physical sensors or motors with the board as the board provides additional points of contact. The board also helps provide easy UART communication via USB bus of PC. The board has onboard LED to indicate the status of the board and jumpers to control the mode of operation of FiPy. The Pytrack mainly has two sensors attached to it, an accelerometer sensor – LIS2HH12 from ST microelectronics and a Quectel GNSS L76-L GPS chip embedded to it. The Pysense has the same accelerometer chip as mentioned above but comes with additional sensors like the Ambient Light sensor – LTR-329ALS-01, Barometer sensor – MPL3115A2 and the Humidity and Temperature sensor – Si7006-A20.



After I was given all these boards, the first thing I had to do was update the firmware to the latest version as older firmware versions did not support many functionalities of the board and had many bugs. Due to poor documentation by Pycom as such, it was very hard to update firmware of FiPy and the other expansion boards. References from core electronics, a website helped a lot with updating of firmware.

Then I started with MicroPython. MicroPython is a programming language derived from normal python but has been made light weight specially for low power microprocessors and microcontrollers. After updating the firmware, I took things slow initially by first learning ways to download MicroPython libraries from Github, how the structure of a basic MicroPython project must be for the code to be runnable for FiPy etc. A basic MicroPython project consists of a lib folder where all downloaded libraries must be copied to so that the code can access them. Then in the parent project folder there must be a boot.py file that contains the microcontroller basic attributes like type of interface for communication, baudrate of interface and the main.py file to run. Since the boot.py file is run by default, in the absence of this file the system throws and error.

Now coming to the IDE, most preferred IDE's are either Atom or Visual Studio. I used Atom. In Atom we had to first install a Pymakr library that supports direct UART communication of PC with FiPy and viceversa.

Now after setting up the work environment for the FiPy, it was time to code the main program. Initially I was told to connect the FiPy to the Wi-Fi and then use a Ublox CAM-M8Q GPS chip to get GGA packets using UART that contain latitude-longitude information. However, due to varied operation voltage and frequencies, interfacing Ublox Using UART was hard. So alternate was to use Pytrack with inbuilt GPS sensor. With FiPy and Pytrack, I started to make my first GPS tracker. My first approach to the problem was to first connect to Wi-Fi and establish internet connection, second receive the Latitude and Longitude coordinates using Pytrack GPS sensor once the sensor gets a lock onto the location and then finally upload these values to the cloud. The first two divisions of the approach were easy to code by referring documentations. The only problem was to decide which cloud service to use. First, I used ThingSpeak server. The problem was low data report rates to the cloud for free users. Then I switched to Wia cloud, which had good report rates for free users but here the problem was the difficulty to access the information stored on the cloud for receiver end purposes. So finally, I used adafruit's io cloud services which not only allowed to report 30 data in a minute to the cloud but also allowed ease of access to the data once stored onto the cloud. With this my sender side code was completed. Next was to use LTE to get internet access. But the only problem was than LTE-M, a low power LTE mode for IoT application is still not available in India. So, all I was able to write a code based on documentation. Practical testing was not possible in India.

Next, I was told to make a receiver GUI for this GPS tracker. I made a GUI using Tkinter, a python library and then integrated it with web browser automation - Selenium to be able to display the location of the GPS tracker on Google Maps and then plot complete course of travel of the GPS tracker after completion of course.

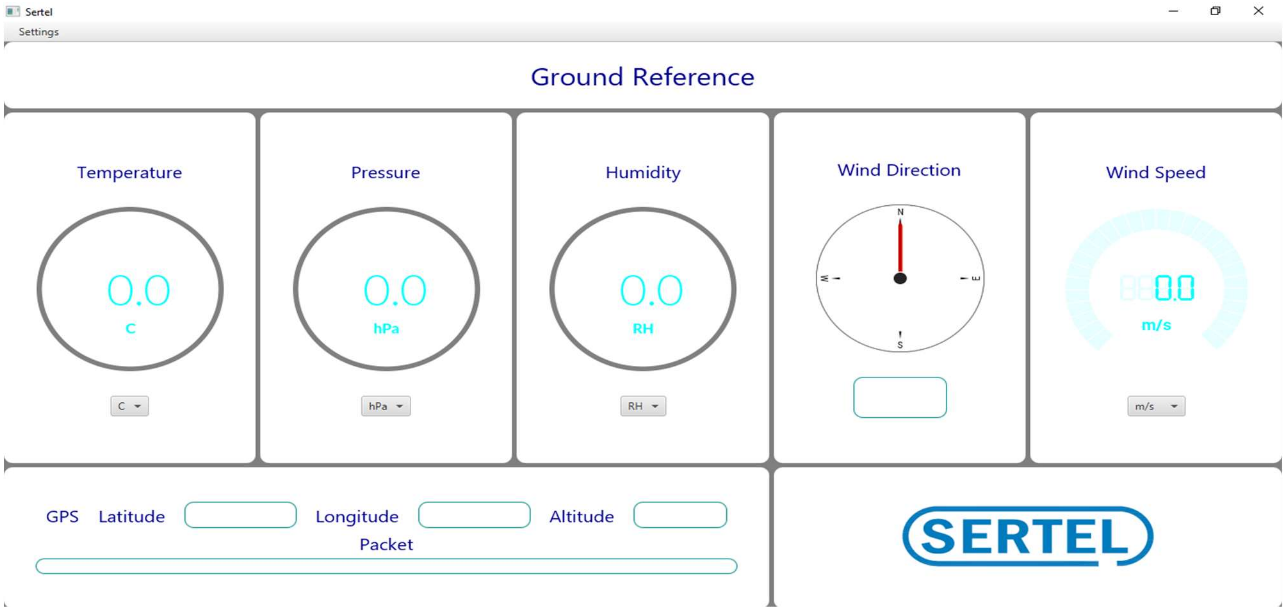
After this I used all the other boards to learn their basics like accessing sensor data using the libraries made for the individual sensors etc.

Next after this I started off with my second project that is face detection using Raspberry Pi 3, a SoC capable of running full fledge python codes. First the Raspberry Pi did not have an OS installed in its micro SD card. So, I downloaded the latest Raspberrian version and made a bootable SD card with Raspberrian on it using Rufus software. Next, I connected the Raspberry Pi 3 with ethernet cable and enable network sharing of my PC with Raspberry Pi. Since I did not have a separate monitor to attach to the HDMI port of the Raspberry Pi, I had to take the alternate route of using my own laptop screen for the Raspberry Pi. This involved first finding the IP address of the Raspberry Pi using Advances IP scanner. Once I got the IP address of the Raspberry Pi, only for the first time I had to use PuTTy software to open the SSH and bash of the Raspberry Pi and then install a remote desktop software(xrdp) inside the Raspberry Pi.

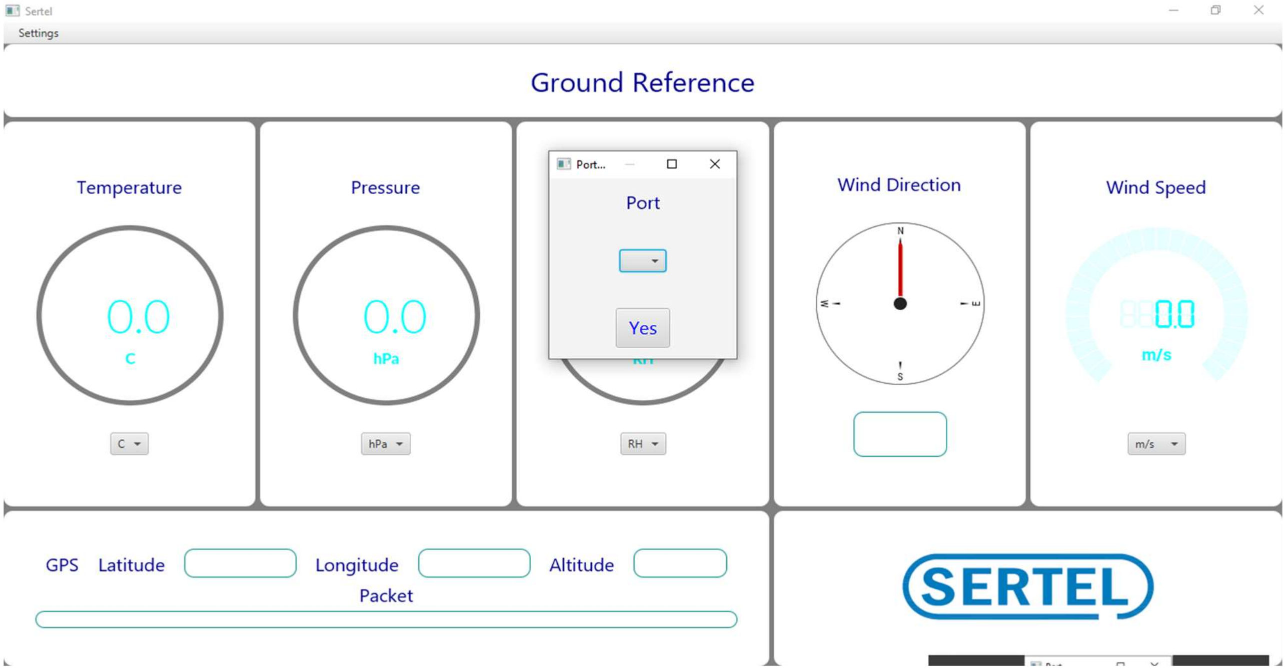
Now after this is installed, all I had to do was input the IP address in our windows default Remote Desktop Software that will link our Raspberry Pi CPU on our monitor. After this I got Raspcam, a camera made specially to interface with Raspberry Pi. A separate ribbon cable port is given for Raspcam on Raspberry Pi. Now after interfacing this camera, I wrote an Open CV code for facial detection by first getting the image from the attached Raspcam and then performing xml analysis to detect the face. Once this ran smoothly, I extended this code to detect faces in real time like for surveillance purposes.

I was finally given a project to make a GUI using JAVAFX for one of their ongoing projects. The ongoing project was related to weather forecasting. Here there are two sensor devices involved. One, a mobile sensor device that transmits data through RF and is sent deep into the sky using Helium balloon to get various parameters in the upper atmosphere. Another is a ground sensor device that also gives info about the various parameters at ground level. Now the data from both these devices is synced and comparisons are made within the data to make predictions. The mobile sensor device is lost after the Helium balloon blast in the upper atmosphere due to low pressure. The GUI I made is used for the ground sensor device.

There are many OOP methods in JAVA to make GUI with most common method form making GUI is SWING in JAVA. JAVAFX is a newer method added into JAVA that allows CSS styling thereby providing more flexibility in making vary attractive GUI. Here is a picture of the GUI I made using JAVAFX.



This GUI as you can see above can display values coming from sensors like Temperature, Pressure, Humidity, Wind Direction, Wind Speed, GPS Location and Altitude as well. The Packet region is used to display the serial information coming from the sensor device on which string manipulation is done to get individual sensor values. The Baudrate of the sensor device is 115200 and when we click the settings button on the top left corner, a new window pops up which will list the port number of all the devices connected to this PC having the known Baudrate.



On selection of the port and clicking yes, the program checks if the incoming packet from the selected device is the wanted packet and coming in the format our sensor device sends using regex. Once verified, string manipulations are done to segregate the data of the sensors from the packet and then this data is respectively displayed in the GUI. The Temperature, Pressure etc are Gauge meters got from an open source library specifically designed for JAVAFX. A blue line of variable circumference appears when values are given to the gauge and the maximum and minimum of the gauge vary according to incoming value. The value automatically changes as new packets come replacing old packets.

4. Learning Experiences

4.1. **Knowledge acquired:** Learnt the following:

Update firmware of FiPy, Expansion board 2.0, Pytrack and Pysense.
Install new OS intro Raspberry Pi 3.
Code ESP32 microprocessors using MicroPython (Light Weight Python)
Use libraries of Python such as Open CV, Selenium, Webdriver, request, Tkinter etc.
Code in JAVA FX and use CSS styling.
Use Libraries of JAVA like JSSC for serial port reader, collections, etc

4.2. **Skills learned:** Skills learnt include:

Learn coding languages – MicroPython, JAVA.

Work under supervisor, how to state your point of view and improve overall professional communication skills.

Time management and self-motivation.

4.3. **Observed attitudes and gained values:** This includes:

Hard work, Dedication, Punctuality, Honesty, Confidence

4.4. **The most challenging task performed:**

I personally feel that the most difficult task given to me was towards the end of my internship where I had to make a GUI using JAVAFX within 10 days of time given I had no prior knowledge in JAVA. My dedication and hard work paid off where without those I wouldn't have been able to attain my target within the stipulated amount of time.

5. Problem Identification and Solution

Problems I dealt with during internship is as follows:

5.1. Pycom poor documentation:

Whenever we buy any new electronic product or device, it is advised to first read the documentation that comes along with it. But Pycom hasn't updated its documentation since the initial release which made updating latest firmware and accessing certain sensors after updated software using MicroPython hard.

Thanks to the efforts taken by core-electronics later whom I found on YouTube, who have been making alternate documentation with respect to the new release of firmware's, I was able to do the updates myself to the FiPy. And for the MicroPython code I had to refer several other websites and Pycom's Forum to get the latest changes made to the libraries and code in general.

5.2. Setting up remote desktop for Pi 3:

Since a monitor was not available to run the CPU of Raspberry Pi on, the only alternate was to use remote desktop. The problem faced though was the not a single source had elaborated the actual steps to be followed to setup a remote desktop and sources for the Pi 3 version were very less.

The solution was to learn how remote desktops are setup initially for other devices using PuTTy software and Advanced IP scanner and then late on make the necessary changes to incorporate it with the Pi 3. The download of xrdp onto Pi 3 by using system network sharing and terminal/bash of Raspberrian was the challenge which finally let me use default Windows Remote Desktop Software.

6. Conclusion

I conclude by thanking all the employees who supported me thought the course of this internship, for giving me such a great opportunity to work under their guidance and learn so many things and gain so many important skills that will be necessary for me in my future jobs. My efforts, hard work and dedication paid off where I was successfully able to complete three projects under a very short duration of time. The exposure to work environment of the company was new to me and educating. Overall my internship at Sertel electronics was a great success.

7. References and Sources Used

The main references used were:

Official Documentation:

- https://pycom.io
- https://docs.pycom.io
- https://python.org
- https://docs.python.org
- https://docs.oracle.com
- https://www.raspberrypi.org
- https://community.oracle.com/docs/

Cloud Services:

- https://adafruit.io
- https://thingspeak.com
- https://www.wia.io

Other Sources:

- https://forum.pycom.io
- https://stackoverflow.com
- https://github.com
- https://core-electronics.com.au
- https://geeksforgeeks.org

Report:

https://www.sertelelectronics.com.