Directed Research Individual Report

Sujita Shah

May 17, 2017

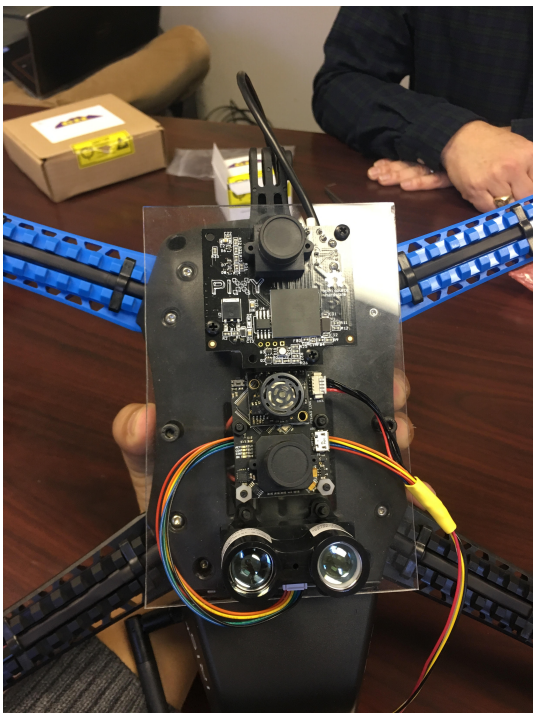
1 Purpose

My decision to participate in research and gain real world experience motivated me to participate in this directed study. I worked with hardware and automation during my summer internship and gained good experience in it. The project description provided us through email was very similar to what I was interested. I wanted to learn more about automation, application development, and gain experience while working individually with least instructions provided.

2 Contribution

2.1 Setup SMACCM-SiK firmware on 3DR Radio and Flashed 3DR Radio

After flashing APM firmware, we needed to flash all three 3DR Radios. While doing so we were getting errors related to Pexpect module of python. There were some errors related to setting of miniterm. We tried to remove Df-13 cable and tried to flash the radio again but radio did not respond. I switched to python version 2.7 and installed everything again including pexpect module. The radio flashed successfully and responded with “‘ AT OK “‘ message. After that I entered more commands for channel number and other setup.



2.2 Ensured IO coprocessor on pixhawk is updated

Before the final configuration of all the parts for arducopter, we needed to perform some steps on our side. I made sure that IO coprocessor on the pixhawk is up-to date.

2.3 Tried to Get GCS software building on lab machines

We also need to get GCS software building on the lab machines but due to storage issue we could not get it done on lab machine. I tried to set it up on my external hard disk after installing Ubuntu on it.

2.4 Ran valgrind script to install all the dependencies

NASA team provided us some documents on setting up AOS on Linux machine. We decided to use Debian VM for the development. One of my team member wrote a valgrind script which includes all the dependencies required to setup the AOS. I ran the script to install all the required software and debugged the errors while doing so.

2.5 Setup SIL - ArduPilot Software in the Loop on Lab Machine

There are three documents we received from the NASA team to set up AOS. The third document is about SIL setup. I followed all the instructions mentioned in this document and setup SIL on lab machine. I also debugged various errors I was getting to while doing so. Unfortunately, I need to repeat all these instructions three times due to different types of errors we are getting while running the test scenarios mentioned in the document.

2.6 Setup SITL simulator on my computer for further testing

Because we were getting errors related to SIL setup on the lab machine, I decided to setup SITL simulator with instructions provided on http://ardupilot.org/dev/docs/setting-up-sitl-on-linux.html page.

3 Record of All the errors in the setup

3.1 flashing the fmu apm firmware and working with Pixhawk

- **Error**: While configuring Pixhawk we were getting error related to Boot loader. The FMU processor in boot loader was steady. We got below message on the terminal while flashing it.

```
sujita@sujita-HP-ENVY-TS-m6-Sleekbook:~/Desktop/resea/smaccmpilot-hardware-prep/
    fmu_apm_firmware$ python px_uploader.py --port /dev/ttyACM0 px4fmu-v2_APM.px4
Loaded firmware for 9,0, waiting for the bootloader...
```

- **Intermediate Efforts** : We looked in the following things before proceeding.

  1. We were using xhci_hcd (usb 3.0) and Rockwell Collins team was using ehci-pci (usb2.0). We tried plugging the pixhawk into a different port on the laptop.
  2. Rockwell Collins product was "PX4 BL FMU v2.x" and ours was "PX4 FMU v2.x", so the BL stands for bootloader in it. For some reason we were not going into proper bootloader mode. We followed above instructions but nothing works out.

- **Final Solution**
  We figured that we needed super user permissions. After that we did the IO coprocessor upgrade and the 3DR radio upgrade to the onboard 3DR radio (the silver one).

3.2 update the smaccm-sik settings and flashing 3DR radios

- **Error:** For the radio, we were not able to update the smaccm-sik settings. We were using python.serial.miniterm as a serial terminal and the radio was enumerating on ttyUSB0. We were using a baud rate of 57600. We were able to connect to the radio using miniterm but we could not enter AT mode or really do anything at all within miniterm. We were also getting errors within the file sik_uploader.py file. It is complaining about module fdpexpect. We tried to change this code with from pexpect import fdpexpect but then it gave error saying serial object has no attribute setBaudrate.

```
from pexpect import fdpexpect
Traceback (most recent call last):
  File "sik_uploader.py", line 279, in <module>
    if not up.check():
  File "sik_uploader.py", line 238, in check
```

```
    self.autosync()
  File "sik_uploader.py", line 201, in autosync
    self.port.setBaudrate(self.atbaudrate)
AttributeError: 'Serial' object has no attribute 'setBaudrate'
```

- **Solution:** We switched back to python 2.7 version and installed everything again including pexpect module.

3.3 Different types of errors during SIL setup

- Lack of expected messages We cloned the AOS repo on VM running on lab machine. While setting SIL using the instructions mentioned in NASA document we found following error. I tried to build SIL and I was at the stage where after installing everything, I ran sim_vehicle.sh -w but I did't get any information after this message "Waitng for heartbeat from tcp:127.0.0.1:5760." We have been working and we could not seem to get the SIL working. We kept getting the error message "APM: PreArm: RC not calibrated." Also, the documentation says that we should at some point see "Received 412 Parameters" and "saved 412 parameters to mav.parm" and we were not able to see that at all. We were running into this issue when we tried everything from the initial ArduCopter build up to the test scenarios.

- Possible solution which we tried
  - **Solution 1:** Tried the command param set ARMING_CHECK 0 at the mavlink prompt to check if there is PreArm issue.
  - **Result of Solution 1:** After trying the command, we got "Unable to find parameter 'ARMING_CHECK'". We have followed the instructions in that file before given from NASA team to fix the issue. We also tried to uninstall and rebuild the entire AOS but nothing worked out.
  - **Solution 2:** We pulled the updated code from AOS repo and tried to make install everything.
  - **Result of Solution 2:** We got errors related to libraries the setup requires.

```
lf -S -I0 --home -35.363261,149.165230,584,353 -w --model + --speedup=1
Connect tcp:127.0.0.1:5760 source_system=255
libdc1394 error: Failed to initialize libdc1394
set FRAME to 0.000000
timeout setting FRAME to 0.000000
set MAG_ENABLE to 1.000000
```

  - **Solution 3:** I tried sudo ln /dev/null /dev/raw1394 to remove the error
  - **Result of Solution 3:** We got different error about timeout while setting the parameter and give RC not calibrated message.

```
lf -S -I0 --home -35.363261,149.165230,584,353 -w --model + --speedup=1
Connect tcp:127.0.0.1:5760 source_system=255
set FRAME to 0.000000
Init APM:Copter  V3.3.3 (fb148f75)
Free RAM: 4096
FW Ver: 120
--------------------------------------------
Firmware change: erasing EEPROM...
Requested parameter list
Log Directory:
Telemetry log: mav.tlog
STABILIZE> Waiting for heartbeat from tcp:127.0.0.1:5760
APM: PreArm: RC not calibrated
APM: PreArm: RC not calibrated
APM: PreArm: RC not calibrated
```

  - **Solution 4:** Using vagrant could be a problem because as we got suggestion from NASA team that they have prebuilt VMs on asodev. The one for apm3.3.3 is debian_mate64.ova, scp aosdev:/home/jpcastle/virtualmachines/ They suggested to have at least have a virtualbox version around for compatibility reasons. We tried the instructions for vagrant sitl as we did not have much time left in the end of the semester mentioned on http://ardupilot.org/dev/docs/setting-up-sitl-using-vagrant.html
  - **Result of Solution 4:** When we were in the Vagrant VM and we tried to run sim_vehicle.sh and we were getting the error "Failed to connect to tcp:127.0.0.1:5760 : [Errno 111] Connection refused". We did not any idea where this issue came from or how to proceed.

- **Solution 5:** Tried using the pre-built VM available NASA machine.
- **Result of Solution 5:** We tried this solution and ran into the same issues that we were having before.
- **Solution 6:** Build the online SITL on Linux using the instructions mentioned on this link http://ardupilot.org/dev/doc up-sitl-on-linux.html and then compare if there is problem in ardupilot setup or AOS setup itself.
- **Result of Solution 6:** I followed the instructions and could not get sim_vehicle.h file in Arduplane folder to proceed further.
- **Solution 7: Developing for AOS Without SIL** No matter what we tried, we did not get SIL working. We decided to proceed without SIL and tried to run some PLEXIL plans without it.
- **Result of Solution 7:** Finally we tried some PLEXIL plan after getting some instructions on it from NASA team. We ran a scenario using cfe by faking autopilot events through some convenience scripts that were provided in AOS. We worked on scenario number 60 and 61. We also made further efforts to get better understanding of it. We discussed with NASA team about the number scheme going on in aos/apps/Plex_mav/fsw/src/Plex_mav.c. There were number of different waypoint_t arrays with different numbers corresponding to waypoint names. We assumed we had to call mb_mis_reach N, where N is the number corresponding with the waypoint names from the scenarios. We tried running scen60 and scen61 from aos/apps/plexil/fsw/mission_inc/scenarios in this manner, but when we called mb_mis_reach 1 for KSJO nothing seemed to happen.

4 Learning Experience

- Gained experience working with hardware and became more familiar with application development and PLEXIL plans.
- Gained knowledge about the different parts of arducopter and its functionality through live setup demonstration at Rockwell Collins and diagrams.
- Familiarized with more Linux command and gained experience with Debian
- Got better understanding of SMACCMPilot system and AOS setup
- I learned how to work independently and debugged code through discussion with other team members.

5 Assessment

What I could have improved.

- Refer to more sources other than documents provided from NASA team and Dr. Rayadurgam.
- Communicate more often with team members and have detailed discussions with them.

What I suggest to Instructors

- Access to more documents to develop AOS applications
- Give us outline of Goals we need to achieve before end of the semester

6 References

- http://smaccmpilot.org/
- http://crisys.cs.umn.edu/smaccm.shtml
- https://ti.arc.nasa.gov/news/AOS-ATC-test/
- https://babelfish.arc.nasa.gov/confluence/display/AOS/SIL+-+ArduPilot+Software+in+the+Loop
- http://ardupilot.org/dev/docs/setting-up-sitl-using-vagrant.html
- Referred all the email to track the errors

Thank you for the opportunity Dr.Heimdahl and Dr. Rayadurgam.