IMAV2018-44
http://www.imavs.org/pdf/imav.2018.44

Enabling Intelligent and Autonomous Drones Using Embedded Linux

K. Amancio*, G. Pinheiro, L. Teixeira and C. Duarte
Federal University of Itajubá, Av. B P S 1303, Itajubá

ABSTRACT

In the current scenario, drones became more than simple toys for hobbyists and started to be tools with many applications. Among those, there are natural disaster prevention and combat, surveillance and monitoring, environmental and civil inspection, public security, rescue and salvage, among others. In many of those applications it is desirable that the aircraft operates autonomously, aiming at greater efficiency in the execution of those tasks. In this article, it will be shown how the technology of embedded computers running a Linux distribution as operating system can make possible to achieve a level of autonomy and intelligence within these aircrafts.

1 INTRODUCTION

This article aim to demonstrate how it is possible to achieve intelligent and autonomous drones using the technology of embedded computers running a Linux distribution as operating system and communicating with a flight controller board on a quadcopter aircraft.

2 ESTABLISHING COMMUNICATION BETWEEN FLIGHT CONTROLLER AND COMPANION COMPUTER

For the purpose of having intelligent and autonomous vehicles for various tasks, first is necessary to integrate a so called companion computer on the drone. These devices often are single board computers, which are credit card size boards with some computational resources like a microprocessor running an Operating System, memory, graphic unit and input/output (I/O).

In this article, we adopted the DragonBoard™ 410c as the companion computer, due to his features as listed below.

- Quad-core ARM® Cortex® A53 at up to 1.2 GHz per core with both 32-bit and 64-bit support
- 1GB LPDDR3 533MHz / 8GB eMMC
- Qualcomm Adreno 306 GPU with support for advanced APIs,
- 1080p@30fps HD video playback and capture
- Wi-Fi 802.11 b/g/n 2.4GHz, integrated digital core
- Bluetooth 4.1, integrated digital core
- On-board Wi-Fi, BT and GPS antenna
- Two USB 2.0 and a micro SD card slot
- One 40-pin low speed expansion connector: UART, SPI, I2S, I2C x2, GPIO x12 and DC power.

On the vehicle, we have a PixHawk Flight Controller Board running a Flight Stack, software and firmware responsible for the stability of the aircraft.

2.1 Physical Connection

To make PixHawk and Dragonboard communicate, we use the UART protocol, available through the

*contact email: kelvin013@gmail.com

TELEM1 port on PixHawk and through UART0 pins on Dragonboard.

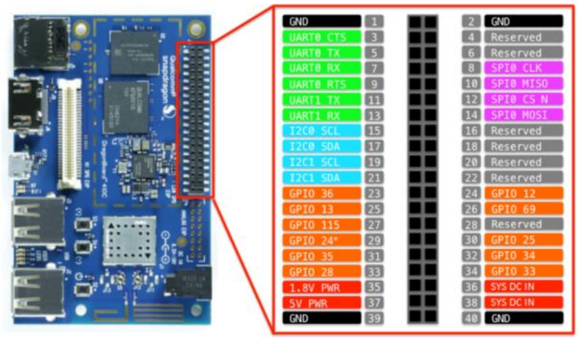

Figure 1 – Dragonboard 410c Pinout

TELEM1, TELEM2 ports

Pin	Signal	Volt
1 (red)	VCC	+5V
2 (blk)	TX (OUT)	+3.3V
3 (blk)	RX (IN)	+3.3V
4 (blk)	CTS	+3.3V
5 (blk)	RTS	+3.3V
6 (blk)	GND	GND

Figure 2 – Telemetry Ports on PixHawk

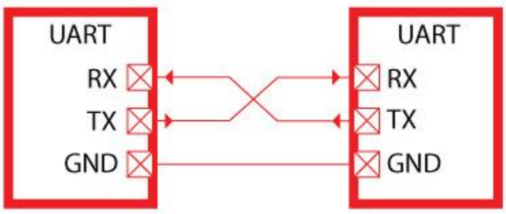

Figure 3 – Connection between Devices Using UART Ports

2.1 Logical Connection

Furthermore, these two devices also need one protocol to establish a connection. This is achieved with MAVLink (Micro Air Vehicle Communication Protocol) that runs on both systems.

As soon as the two devices are properly configured with the correct requirements and libraries and the two boards are physicaly connected, the connection can then be established.

```
SERIAL2_PROTOCOL = 1

SERIAL2_BAUD = 921

LOG_BACKEND_TYPE
```

Figure 4 – Parameter that Need to be Modified in the PixHawk Settings

```
$ sudo apt update

$ sudo apt install screen python-wxgtk3.0
python-matplotlib python-opencv
python-pip python-numpy
python-dev libxml2-dev libxslt-dev
python-lxml

$ sudo pip install future

$ sudo pip install pymavlink

$ sudo pip install mavproxy
```

Figure 5 – Linux Commands to Install Necessary Libraries and Dependencies on Dragonboard

```
$ mavproxy.py --master=/dev/ttyMSM1
--baudrate=57600
```

Figure 6 – Linux Commands to Test the Connection between Dragonboard and PixHawk

If the connection is successfully established, then it is now possible to run algorithms developed in a programming language such as python to command the aircraft to perform a diversity of tasks, such as avoid collision, detect a target, identify and follow an object, among many others.

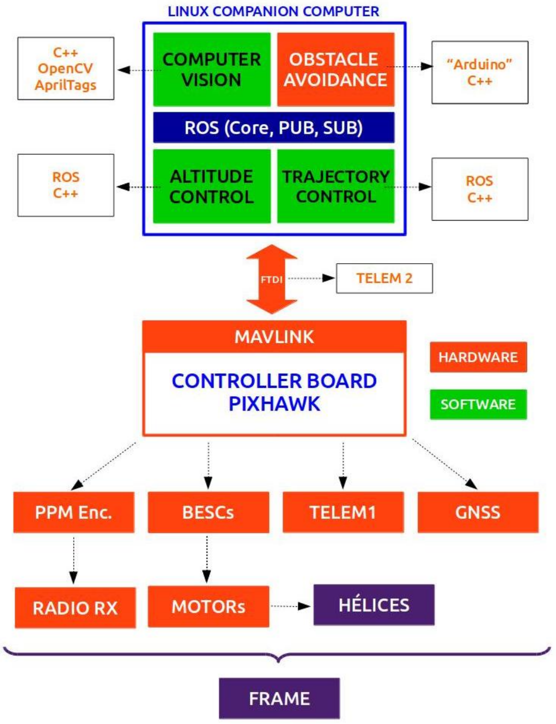

Figure 7 – Drone Schematic Representing the Devices on the Aircraft and Applications that can be Executed on the Companion Computer

3 ROS AND COMPUTER VISION

On the Black Bee Drones team, we have a division called Software, in that division, we develop the artificial intelligence that is applied to our drones. We do that using ROS (Robot Operating System) and Computer Vision Algorithms to recognize the environment, understand the data obtained and make decisions based on that.

Our primary tool that makes all of this autonomy possible is ROS (Robotic Operating System). ROS is a meta-operating system that allows us to work with a heterogeneous group of computers to control our drone from an onboard computer or a ground station where we apply our computer vision algorithms so that the environment around the drone is understood and a decision can be made.

The decision-making process is divided into two steps, first, all the data from the drone's camera is read, this data is processed by the algorithms developed by our team with a strong focus on specific tasks, like finding an object or pattern.

The second step is when those algorithms are used to recognize answer if something where found or not. The control part of the decision-making is then activated; this is when the drone executes an action based on the results obtained from the data processing. For example, an algorithm designed to find and follow an object; the drone would get the coordinates of the object on relation to itself through the camera and follow the movement based on the variation of where the object is on the image obtained.

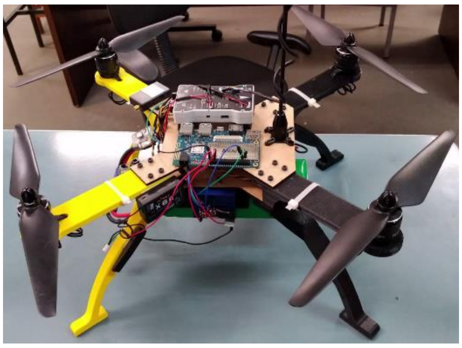

Figure 8 – Drone with PixHawk and Dragonboard Communicating Through UART Connection

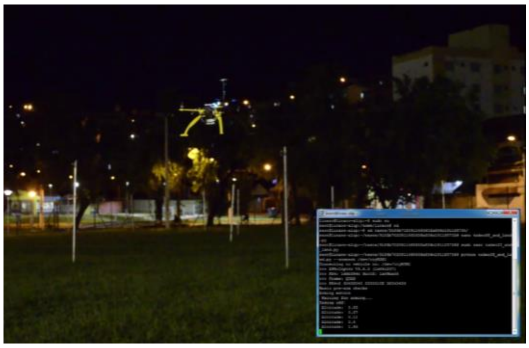

Figure 9 – Take Off, Rover and Land Test with a Python Algorithm Running on Dragonboard

4 CONCLUSION

After all the procedure described in this article, it is demonstrated that is possible to obtain aircraft with the capability of execute autonomous tasks, such as avoid collision, target detection, identification and following an object. This improves the efficiency, flight autonomy and security in comparison with a manual controlled aircraft and the development of this technology is the goal of Black Bee Drones team.

ACKNOWLEDGEMENTS

We would like to express our deep gratitude to Professor Dr. Alexandre Carlos Brandão Ramos, our project supervisor, and all of ours Black Bee Drones colleagues.

REFERENCES

FLOREANO, D.; WOOD, R. J. Science, technology and the future of small autonomous drones. Nature, Nature Research, v. 521, n. 7553, p. 460–466, 2015.

MEIER, L. et al. Pixhawk: A system for autonomous flight using onboard computer vision. p. 2992–2997, 2011.

Saif Aldeen Saad Obayes Al-Kadhim and Imam Alkadhim Collage. Communicating with raspberry pi via mavlink. 2016.