Unmanned Aerial Vehicle Project (UAV)

Set up and User Manual

Anastasis Georgiou

Email: ageorg39@ucy.ac.cy

Department of Electrical and Computer Engineering
University of Cyprus
December 29, 2017

Contents

Chapter 1

Introduction

1.1 Background

UNTIL recently, unmanned air vehicles UAVs had been largely connected in the public mind with military uses, such as armed attacks. However, academic and industrial researchers has started to change the above perception and used remotely piloted aircraft commercially for agriculture, infrastructure inspection, rescue and etc.
Particularly, at KIOS Research and Innovation Center of Excellence we are interested to develop a solid package, which combine DJI drone with embedded systems in order to improve and sustain critical infrastructure in developing countries.

1.2 Aims and Objectives

The current drone set-up is under the Innovation Hub project funded by EAC and aims to inspect electrical infrastructure. The primary objective is to design onboard controller, which has the ability to complete a static task considering dynamic environment. The initial goal of the controller is to keep the drone in a constant distance above ground and obstacles.

1.3 Contributions

The contribution of our research can be included into the following categories:

- Control Systems Design,
- Collision Avoidance Systems,
- Intelligent vehicles,
- Embedded Systems.

Control Systems Design: It means the use of direct control methods in order to obtain desired vehicle control inputs such as motor rotation speed orientation.

Intelligent Vehicles: Research around air vehicle dynamic is made in order to derive system model for prediction the next state of the drone. The estimation of the upcoming state of the UAV is used to prevent unsafe circumstances.

Embedded Systems: Various control unit and sensors are interconnected with the UAVs in order to improve drone capabilities for perception, planning and navigation control .

1.4 Structure

The following chapter describe the software and hardware requirments setup in onder to replicate the project. The third chapter summarise to application which implemented at the KIOS laboratory. All the code for this project is open and located at:
`https://github.com/ageorg39/KiosDJIProject`

Chapter 2

Development Kit

This guide aims to support anyone who might continue the project to set up the hardware, and software, as well as, understand the structure of the provided code easily. It further gives a rough overview of the DJI drone, with all the embedded system connected with it at its current state. The current set-up is still in development so some of the details given here might change in the future.

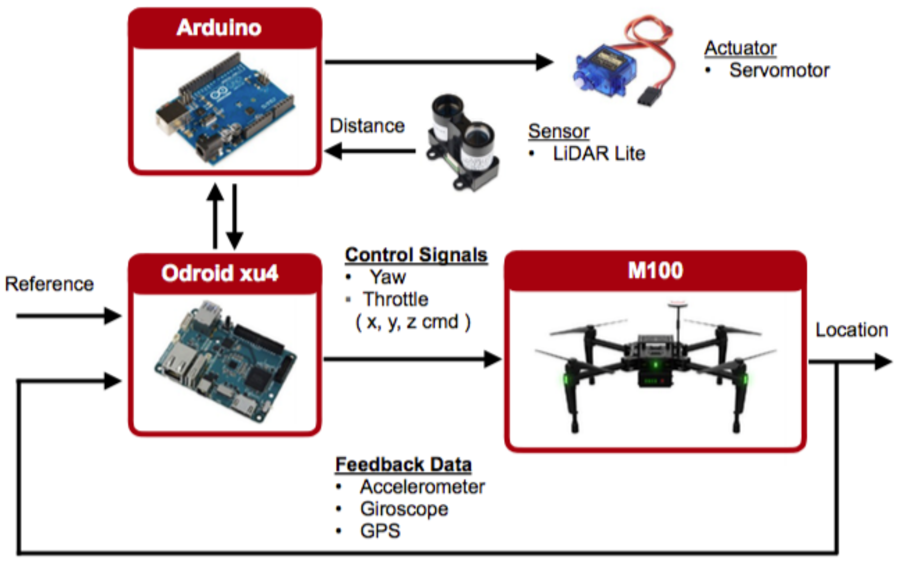

Figure 2.1: Airial vehicle system set-up

2.1 Software Environment and Setup Guide

This guide details the software environment needed to work with the Onboard SDK.

2.1.1 Ubuntu Linux

Toolchain

To build standalone Linux applications based on the OSDK, you need:

- A supported C++ compiler - currently only GCC (Tested with gcc 4.8.1/5.3.1)
- A bash shell
- CMake >= 2.8
- A modern Linux distribution (Ubuntu MATE 15.10)

2.1.2 SDK and Required Tools

- Download the onboard SDK (3.3) repository from Github.
  `https://github.com/dji-sdk/Onboard-SDK`
- Download the DJI PC Assistant 2 software for Windows/Mac.
  `https://www.dji.com/matrice600/info#downloads`
- Download the DJI GO App to your mobile device.
  `https://www.dji.com/goapp`

2.1.3 Update Firmware

- Connect your computer to the Micro-USB port on the M100/600.
- Update your aircraft/flight controller with the latest released firmware. Please visit the Compatibility Matrix to find out which SDK version your firmware supports.

2.1.4 Onboard SDK Application Registration

You must register as a developer with DJI and create an OSDK application ID and Key pair. Please go to `https://developer.dji.com/register/` to complete registration. Each new vehicle or flight controller must be activated the first time it is used with an OSDK application. The OSDK provides APIs for this activation, and all OSDK samples implement the activation.

2.1.5 Arduino Uno

In order to pass data from LiDAR lite to Ondroid the arduino has to be load with the code that is uploaded to the following link:

`https://github.com/ageorg39/KiosDJIProject/blob/master/LIDARLite_I2C_Library_GetDistance_ContinuousRead/LIDARLite_I2C_Library_GetDistance_ContinuousRead.ino`

2.2 Hardware Environment and Setup Guide

2.2.1 Components List

The following subsection briefly describe the key components which are connected to form the the system.

Matrice 100

The DJI Matrice 100 is a fully customizable and programmable flight platform that lets you to turn your ideas and dreams into reality. Equipped with DJIs easy-to-fly technology and optimized for easy programming through the DJI SDK, the Matrice 100 is ready to carry any sensors, devices that you want to put into the sky.

Figure 2.2: DJI Matrice 100 drone

Documentation webside:
`https://developer.dji.com/onboard-sdk/documentation/introduction/homepage.html`
Were to buy:
`https://github.com/ethz-asl/dji_onboard_sdk_ros/wiki/Parts-List`

Odroid XU4

ODROID-XU4 is a new generation of computing device with more powerful, more energy-efficient hardware and a smaller form factor. Offering open source support, the board can run various flavors of Linux, including the latest Ubuntu 16.04 and Android 4.4 KitKat, 5.0 Lollipop and 7.1 Nougat. By implementing the eMMC 5.0, USB 3.0 and Gigabit Ethernet interfaces, the ODROID-XU4 boasts amazing data transfer speeds, a feature that is increasingly required to support advanced processing power on ARM devices. This allows users to truly experience an upgrade in computing, especially with faster booting, web browsing, networking, and 3D games.

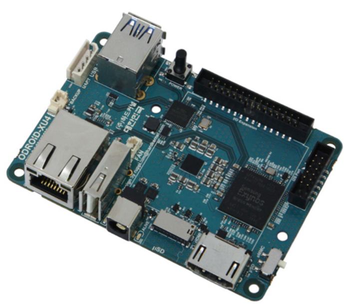

Figure 2.3: Odroid XU4

Documentation [pdf]:

`http://www.me.umn.edu/courses/me2011/arduino/arduinoGuide.pdf`

Were to buy:

`http://www.hardkernel.com/main/products/prdt_info.php?g_code=G143452239825`

Ardoino Uno

Arduino is an open source computer hardware and software company, project, and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical world. The project's products are distributed as open-source hardware and software, which are licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL),[1] permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially in preassembled form, or as do-it-yourself (DIY) kits.

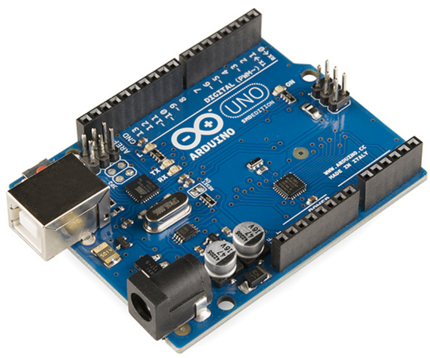

Figure 2.4: Ardoino Uno

Documentation webside:

`http://magazine.odroid.com/odroid-xu4/`

Were to buy:

`https://store.arduino.cc/usa/arduino-uno-rev3`

LiDAR Lite v1

This is the LIDAR Lite, a compact high performance optical distance measurement sensor from PulsedLight. The LIDAR Lite is ideal when used in drone, robot, or unmanned vehicle situations where you need a reliable and powerful proximity sensor but dont possess a lot of space. All you need to communicate with this sensor is a standard I2C or PWM interface and the LIDAR Lite, with its range of up to 40 meters, will be yours to command.
Each LIDAR Lite features an edge emitting, 905nm (75um, 1 watt, 4 mrad, 14mm optic), single stripe laser transmitter and a surface mount PIN, 3 FOV with 14mm optics receiver. The LIDAR Lite operates between 4.7 - 5.5VDC with a max of 6V DC and has a current consumption rate of ¡100mA at continuous operation. On top of everything else, the LIDAR Lite has an acquisition time of only 0.02 seconds or less and can be interfaced via I2C or PWM.

Figure 2.5: LiDAR Lite

Documentation webside:
`https://github.com/PulsedLight3D/LIDAR-Lite-Documentation/blob/master/Docs/LIDAR-Lite-v1-docs.pdf`
Were to buy: `https://www.sparkfun.com/products/retired/13167`

Micro usb to UART TTL Converter

Figure 2.6: Micro usb to UART TTL Converter

Were to buy:

http://www.uctronics.com/cp2102-micro-usb-to-uart-ttl-serial-converter-module-stc-
html

UART cable

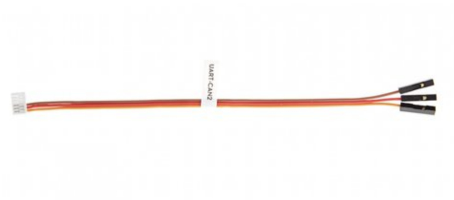

Figure 2.7: UART cable

Were to buy:

https://www.globe-flight.de/DJI-Matrice-100-M100-UART-Cable-PART30

2.2.2 Hardware Installation

1. Connect ondroid with M100 by micro USB to UART converted and UART cable.

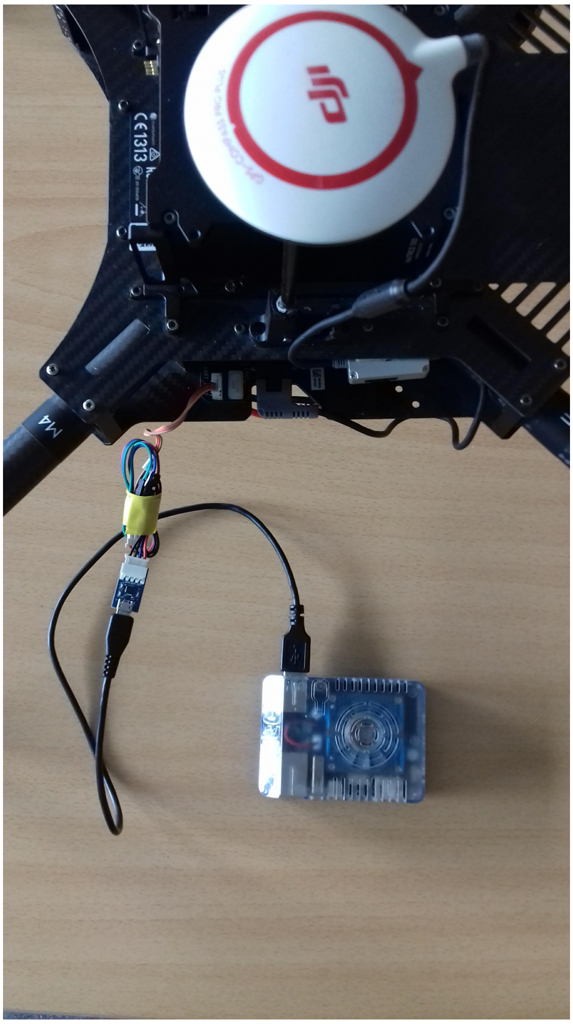

Figure 2.8: Micro USB to UART converted and UART cable connecting odroid with M100.

2. Connect Arduino Uno with ondroid by micro USB.
3. Connect LiDAR lite with Arduino (see figure).
4. Connect any other sensor to or acuator to Arduino.

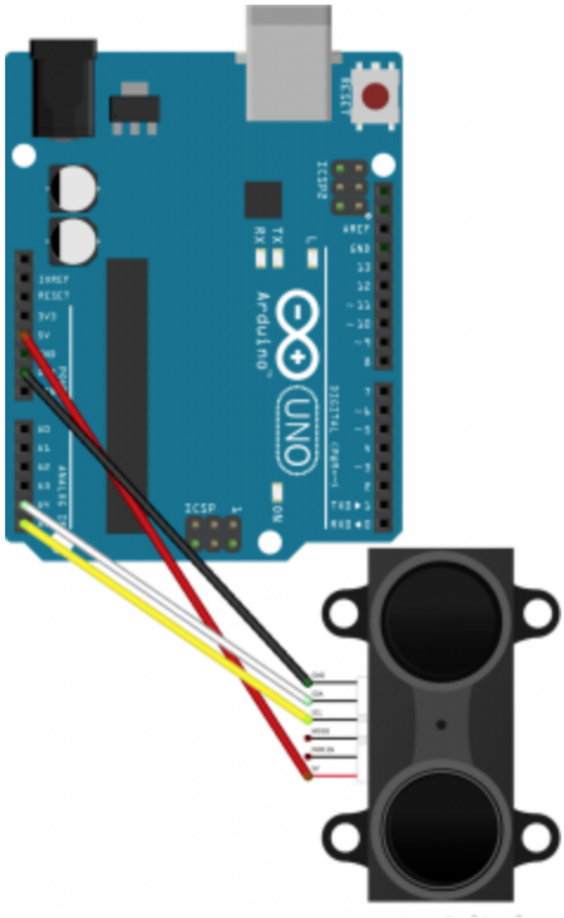

Figure 2.9: Connecting the LIDAR Lite to an Arduino

Chapter 3

Implemented Application

3.1 Flight Control

The flight control sample demonstrates sending control commands to the aircraft using the Position control commands, as well as, real-time sensor data from the aircraft and LiDAR sensor. The sample allows you to run operations such as:

- Monitored Take off
- Monitored Landing
- Get Status Data
- Law Level Position Control
- High Level Mission
- Print Data from LiDAR

To run the flight control example on DJI Assistant 2 you have to following the following steps:

1. Connect M100 with micro usb to the PC that simulator is installed.
2. Open DJI Assistant 2 on start simulator mode.
3. Run on terminal the commands:
   - "cd home/odroid/Documents/Anastasis/Onboard-SDK-3.3/ "
   - " ./compile.sh "
4. Open a new terminal window and type:
   - "cd home/odroid/Documents/Anastasis/Onboard-SDK-3.3/build/bin "
   - " ./djiosdk-flightcontrol-sample "

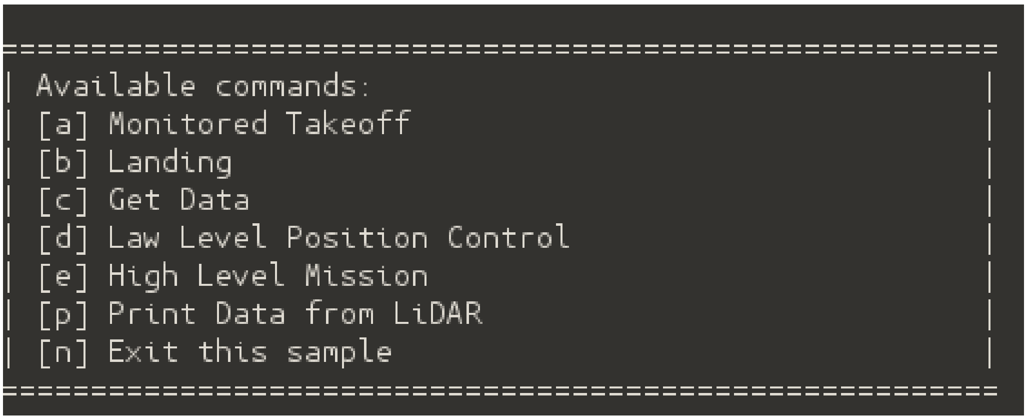

Figure 3.1: Screen-shot of menu appeared on the terminal at flight control sample.

3.2 Autonomous Flight using LiDAR

The Autonomous Flight using LiDAR sample take off the drone at 1.5 meters and then the drone move forward for 6 meters while keeps its distance to the ground constant at 2 meters. After compliting the mission the drone land.

To run the Autonomous Flight example on DJI Assistant 2 you have to following the following steps:

1. Connect M100 with micro usb to the PC that simulator is installed.
2. Open DJI Assistant 2 on start simulator mode.
3. Run on terminal the commands:
   - "cd home/odroid/Documents/Anastasis/Onboard-SDK-3.3/ "
   - " ./compile.sh "
4. Open a new terminal window and type:
   - "cd home/odroid/Documents/Anastasis/Onboard-SDK-3.3/build/bin "
   - " ./djiosdk-LiDAR-sample "

In order to run the Autonomous Flight example without simulator just edit the rc.local file at as it shown below.

```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

if [ -f /aafirstboot ]; then
    /aafirstboot start
fi


if [ -f /aafirstboot ]; then /aafirstboot start ; fi

/home/odroid/Documents/Anastasis/Onboard-SDK-3.3/AutoBoot.sh

exit 0
```

Figure 3.2: Screen-shot of rc.local file containt.