CS4140 QR Operations Manual (2017-2018)

Ioannis Protonotarios

Embedded Software group
Faculty of Electrical Engineering, Mathematics, and Computer Science
Delft University of Technology

April 2018

1 Introduction

This document describes how to operate the **Quadruple** Quad-Rotor Aerial Vehicle, used in the MSc course CS4140 Embedded Systems Laboratory[1]. The last section contains a list of very important safety guidelines that must be obeyed at all times.

2 QR Overview

The QR's architecture (Figure 1) allows it to run in both tethered and wireless flying mode, as explained later. It features an ARM Cortex M0 based SoC, the nRF51822 (hereafter to be called 'nRF SoC') from Nordic Semiconductor, running at 16 MHz. The nRF SoC acts as an interface between the 4 motors, 3-axis accelerometer, 3-axis gyroscope, and the barometer. The motors are the Sunnysky X2212-13 980KV II brushless, each individually controlled by a FlyColor Raptor F330 20A Electronic Speed Controller (ESC). The 4 ESCs receive their RPM setpoints from the nRF SoC. The sensor module used is the GY-86, that features a MPU-6050 9-axis MEMS by Invensense (3-axis accelerometer, gyroscope, and magnetometer) and the MS5611 barometer by MEAS. The MPU-6050 incorporates a Digital Motion Processor, which processes complex 6-axis MotionFusion algorithms that provide fused sensor data (i.e., angles).
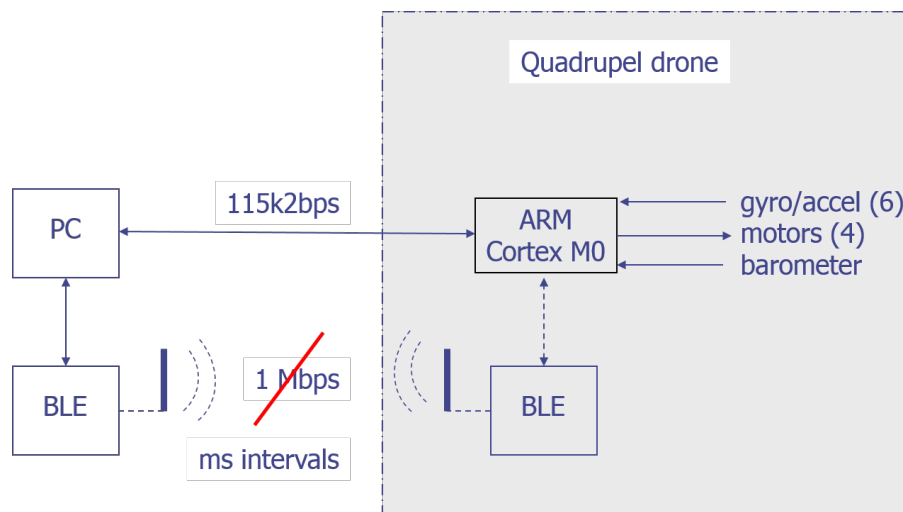

Figure 1: QR Hardware Architecture.

The nRF SoC communicates with the PC through either a wired duplex serial link (115.2 kbps, 8n1) or the Nordic UART Service (NUS), that emulates a serial port over Bluetooth Low Energy (BLE). The communication interface chosen depends on the QR flying mode (wired link in tethered mode, wireless link in wireless mode). Because of the intermittent operation of the wireless link, uploading and testing is performed in tethered mode. Only when the control design is fully validated, the QR can be flown in wireless mode.
The Flight Controller Board (FCB) of the Quadrupel features the following switches and connectors:

---

[1]CS4140 Course Web Site: `http://www.st.ewi.tudelft.nl/~koen/cs4140/`.

- Main power switch

- Reset button for the nRF SoC

- USB-B port to the FT230X UART-to-USB bridge

- 4 motor connectors (servo/Futaba style)

- 6-pin micro-JST connector to the Power Distribution Board (PDB) that provides one $5\,V$ power supply rail ($1\,A$), as well as voltage and current measurements from the battery

- SDA, SCL (I^2C) and 4 GPIO pins for interfacing additional hardware/debugging

- 3v3, $5\,V$, GND signals

- SWD pogo-pin style pads for programming the nRF SoC Flash directly

The FCB also accommodates 4 LEDS (Green, Blue, Red and Yellow) for diagnostic purposes, as well as 2 LEDs (Tx-Rx) for the FT230X bridge.

3 Operating the QR

The FCB is powered via two independent power sources, the PDB and/or the USB host. The main power switch toggles power on/off. The attached 3-cell 4000 mAh LiPo (Lithium Polymer) battery powers the ESCs and the FCB through the PDB. The ESCs boot-up immediately when the battery is connected to the PDB as there is no switch between the two. After making the motors beep 3 times, indicating that they detect a 3-cell battery being connected, the ESCs will wait for an input signal.

As soon as the FCB is powered, the bootloader is executed, with the sole purpose of accepting a new binary to upload and execute. After this, the QR is ready to be operated.

3.1 Power

The LiPo is the only source of energy to the motors, which consume more than $99\,\%$ of the total power spent. A LiPo is a potentially explosive device when overloaded and needs to be handled with care not to ignite! When a LiPo has been overcharged or undercharged it will be irreparably damaged, and can often become internally pressurized. If the battery feels puffy or starts to balloon, immediately disconnect the battery from the QR and store it at a safe location for handling by the TAs!

In order to prevent the LiPo from being too much depleted you are required to always check its voltage and/or current draw using the ADC of the SoC as instructed by the TAs. Failure to do so will be penalized.

3.2 Motors

The motors are brushless motors. Unlike ordinary motors that receive their electrical power through brushes, the brushless motors virtually don't wear, and provide much more efficiency. However, there is a trade-off. Controlling brushless motors is less straightforward than ordinary motors, and requires careful timing of the currents that flow into the various coils within the motor. Motor control is performed by the dedicated micro-controllers and nFETs on the ESCs. The firmware running on the micro-controller of the ESCs is BLHeli version 14.4, with DampedLight and Demag compensation enabled. The first feature will add loss to the motor for faster retardation, which will in turn make the controller more responsive. The latter protects the motor from damage from stalling by detecting it and restricting power to the motor. However, to prevent stalling at low rotor speeds, which is possible but cannot remedied by lowering the power, the user is advised to always exercise a minimum RPM setting.

4 Interfaces Specification

As mentioned earlier, the nRF SoC is interfaced with the GY-86 sensor board, the voltage and current sensor of the PDB, as well as the four brushless motors. This section provides more information on these interfaces.

4.1 MPU-6050

The MPU-6050 is interfaced using the I^{2}C protocol (mentioned as TWI in Nordic's Software Development Kit or SDK) at 400 kHz. Details about the driver implementation (blocking) can be found in the 'drivers' folder, TWI.c. There are two modes of operation for the MPU-6050, namely the *RAW* and *DMP* modes. In both modes, the sensors used are the accelerometer and the gyroscope. In addition, an auto-calibration feature is enabled, that removes the gyro bias whenever the quadcopter is stationary for more than 8 seconds.

All data that is to be read, is pushed in an on-board FIFO at the user-indicated rate. The level of an interrupt pin on the GY-86 sensor board is raised whenever a new sample is added to the FIFO. If the FIFO is not read at that rate, a FIFO overflow will occur, which will result in a FIFO reset and probably a crash!

The differences between the two modes are:

RAW In raw mode, the sensors can be read at a frequency of 32 Hz to 8 kHz (user selectable). In particular, 3 axis for accelerometer and 3 axis for gyro are pushed into the FIFO, as well as the number of unread samples. There is no hardware filter running in raw mode.

DMP In dmp mode, the on-board DMP is running an algorithm that fuses the raw sensor readings into quaternions, that describe the attitude of the quadcopter in 3-D. These quaternions are then transformed to Euler angles on the nRF SoC using a slow floating-point library. The raw accelerometer and gyroscope data is also pushed into the FIFO as in raw mode, but this time a hardware low-pass filter with a cutoff frequency of 10 Hz is applied beforehand. In dmp mode, the FIFO is updated at a fixed rate of 100 Hz.

Both raw data as well as the fused Euler angles are stored in signed 16-bit integers.

4.2 MS5611

The same driver as with the MPU-6050 is used to interface the MS5611 barometer. A new sample is available every approximately 20 ms, as there is an on-board filter that averages many pressure samples. Both temperature and pressure readings are stored in signed 32-bit integers. Values are scaled with a factor of 100, so an integer reading of XXXX represents the fractional number XX.XX.

4.3 ADC

The internal ADC of the nRF SoC is used to sample the voltage divider on the PDB as well as the current sensor. The ADC output is stored in an unsigned 16-bit integer. Values are scaled with a factor of 100 as for the MS5611 readings. Recall that the voltage of the LiPo battery should never go below 10.5 V.

4.4 ESCs

The interface to the ESCs, and subsequently to the motors, is based on the OneShot125 protocol. This dictates that the RPM setting for each motor (or its throttle) is set by the duration of a positive pulse between 125μs and 250μs. Thus the update rate is close to 4 kHz, allowing for very short control loops. As soon as the ESCs detect a signal like this, even with a short positive pulse indicating a zero RPM setpoint, they will arm, while making a high-pitch sound followed by a low-pitch tone. Before this, any RPM setpoint will be ignored.

5 Programming the nRF SoC

The nRF SoC is pre-flashed with Nordic's bluetooth software stack called *Softdevice*, and a bootloader. The Softdevice's API is used to implement all BLE-related tasks (advertising, connection, the UART service) that in the end are transparent to the user of the Quadrupel. Each FCB advertises the NUS service using the name 'Quadrupel #' with the number being on a sticker on the bottom side of the FCBs handed out or on the forward boom of the fully assembled Quadrupels. Using the script provided, a pseudo terminal or PTY is created that emulates a serial device on the system, like the one used to connect to the serial communication over the wired link.

The bootloader is the first piece of software that the nRF SoC will execute. After powering on the FCB, or after a reset, a constant blue and green LED indicate that the bootloader is running. At this stage, the `serial_dfu` python script provided (`in4073/serial_dfu`) can be used to upload the binary to the FCB. After verification, the bootloader will execute the just uploaded binary. As a safety measure, the uploaded binary will never be executed again, unless it is re-uploaded. You can find usage examples in the project Makefile.

The bootloader, as well as the sample program provided, has been implemented using only a few parts of the 'nrf5 SDK', version 10.0 by Nordic. However, the software pack provides all the software parts of the SDK, so there is no need to download a new copy. The same applies for the arm gcc cross-compiler, which is provided to make sure that there will be no compatibility issues in the future. In each version, Linux or Windows, tailored instructions are provided within the pack's README file.

6   Safety Guidelines

Operating the QR implies that the user has read, understood, and accepts the following quad rotor safety guidelines.

The Quadrupel is a complicated piece of equipment integrating engineering materials, electronics, mechanics, aerodynamics, high frequency radio and embedded software. Correct installation and operation are a must in order to prevent accidents from taking place. Operation of the quad rotor should be performed in a safe and responsible manner. Improper operation may result in serious bodily or property damage.

General safety guidelines:

- Keep the QR away from obstacles and people. Quadrotors are highly agile and can move at considerable speed. Therefore it is best to keep them as far away from other people as possible. Also, keep clear from obstacles that are either expensive or accident-prone (such as high-tension lines).

- Keep the QR away from humidity. Humidity will affect the quad rotor's electronics and result in unpredictable behavior or a crash.

- Keep the QR away from heat sources. Heat may damage the quad rotor and impair subsequent use.

- For initial testing hold the QR in one hand, strap it with a velcro, and use the other hand for operating the PC (joystick); never have another student pilot the QR. Also, make sure the QR is held whenever your code is running on it to avoid it taking off on its own.

- If not an experienced model pilot, always fly under direct supervision of someone that is experienced: operating a quad rotor is difficult. Maiden flights should only be attempted under the direct supervision of a TA.

- Operate the QR with concentration. Do not operate a quad rotor if you are tired or in any other sense less acute than normally. Loss of attention can result in crashes and consequential damage.

- Always check the quad rotor before flight. During flight, parts vibrate and may loosen, such as little screws, the battery connection, or the propeller-holding nuts. Verify before the flight that there are no loose parts, in order to prevent erratic behavior.

Also pay special attention to the following:

- The rotating blades: keep them away from other people and do not try to touch them as they can cause deep cuts. Wear safety goggles to prevent damage to your eyes.

- The LiPo batteries: if used wrongly they can ignite and even explode, for instance when overcharged. Do not use them outside the range of -20 °C to 60 °C, the voltage should never go below 3 V per battery cell neither should it go above 4.2 V per cell during charging. Avoid serious impact on the batteries and do not use sharp objects on it. Do not short circuit the battery's wires by, e.g., cutting them with scissors. Store the batteries at room temperature (19 °C - 25 °C).

- Charge and discharge the LiPo battery with caution, adhering to the guidelines for 3-cell LiPo operation.