H01Q6: probleemoplossen en ontwerpen EAGLE: The hardware platform

Y. Murillo, B. Van den Bergh, W. Devriese, S. Pollin, R. Vanlaer

September 2016

This document describes the EAGLE hardware, giving an overview of the different hardware modules that are needed to build a smart drone. Programming, software and algorithms are covered in other documents.

We start with introducing the bare frame of the drone in Section 1. The ZyBo board contains the brains and control of the EAGLE, as introduced in Section 2. Then, we subsequently cover the IMU (Section 3), the camera (Section 4), the remote control (Section 5), the speed control (Section 6) and finally the power distribution (Section 7). This should give you enough information about the basic hardware infrastructure.

Contents

1 The EAGLE frame

A broad range of possibilities exist when considering drone frames, which vary in weight, size, building materials etc. For EAGLE, we chose this type of frame for the drone: Hobbyking X650F frame, see Fig. 1. From [1] we find following information:

> The Hobbyking X650F is a high quality folding-arm quadcopter frame that offers great looks and performance at a price that can't be beat! Built from light weight yet rigid glass fiber and aluminum, the X650F offers a great combination of weight savings and strength.
>
> This quad features folding arms which make it easy to transport and store. The arms are also two different colors for orientation, gold and black, so it's easy to keep your quad flying in the right direction. The glass fiber universal motor mounts will accommodate a wide range of motors and double as a vented ESC mount providing excellent cooling.
>
> This quad frame sports extended landing skids which give you the ground clearance needed for carrying most cameras. It features a dual rail camera mount attachment. (Please see related items below for optional camera mount)
>
> We offer a full line of multi-rotor electronics such as multi-rotor control board, ESC, motors and more, so it has never been easier or more affordable to get a quad in the air!
>
> All necessary hardware is included, just add your own electronics.
>
> Specs: Weight: 598g (frame only) Width: 550mm Height: 265mm Ground clearance: 155mm (bottom of frame), 185mm (camera mount rails) Motor Bolt Holes: 16 30mm
>
> Required: Your own 4 channel transmitter and receiver Multi-Rotor control board 20 40A brushless ESC x 4 3S 2200 3000mAh lipoly battery 28/35 900-1100kv brushless motor x 4 9 11 inch propeller x 4 (2 standard/2 reverse rotation)

It is basically a bare frame on which we can build and add all are own parts. The choice was made for a large drone frame with a flexible landing gear, this to be able to carry all the equipment we want to add and to have some protection against crashes.
Additional protection may be added, but the weight is always a limiting factor when building a drone.
Also, small central plates in plexi glass have been made so all hardware can be screwed on the drone, making the whole system more solid and reducing the risk of hardware damage due to crashes.

2 The Zybo board

The brain of the EAGLE consists of a Zync FPGA, with has two large ARM Cores. This is quite a powerful FPGA, available on a broad range of development boards. We will work with the Zybo development board, with lot's of processing power and a good set of IO options.

Here you find a complete list of the ZyBo specs (quite complete list from [2], feel free to search around for more info):

[1] $http://www.hobbyking.com/hobbyking/store/uh_viewitem.asp?idproduct = 29600$, September 2016
[2] $http://store.digilentinc.com/zybo - zynq - 7000 - arm - fpga - soc - trainer - board/$ September 2016

Figure 1: The Hobbyking frame is a good trade-off between cost, size and cargo power.

- Xilinx Zynq-7000 (XC7Z010-1CLG400C)
- 28,000 logic cells
- 240 KB Block RAM
- 80 DSP slices
- On-chip dual channel, 12-bit, 1 MSPS analog-to-digital converter (XADC)
- 650 MHz dual-core Cortex^{TM}-A9 processor
- On-board JTAG programming and UART to USB converter
- DDR3 memory controller with 8 DMA channels
- 512 MB x32 DDR3 w/ 1050Mbps bandwidth
- 128 Mb Serial Flash w/ QSPI interface
- microSD slot (supports Linux file system)
- High-bandwidth peripheral controllers: 1G Ethernet, USB 2.0, SDIO
- Low-bandwidth peripheral controller: SPI, UART, I2C
- Dual-role (Source/Sink) HDMI port
- 16-bits per pixel VGA output port

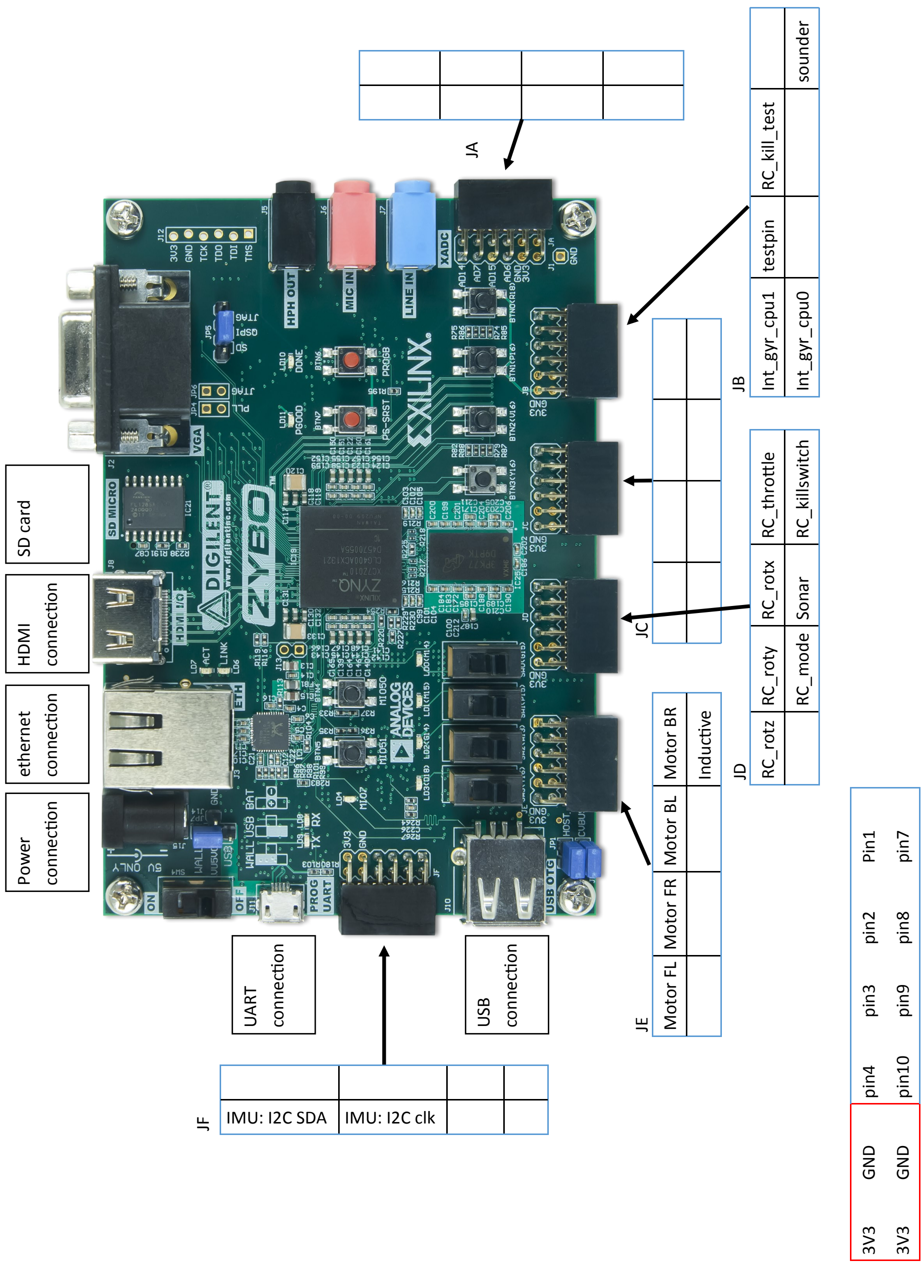

Figure 2: The Zynq FPGA is powerful enough the make the EAGLE smart

- Trimode (1Gbit/100Mbit/10Mbit) Ethernet PHY
- OTG USB 2.0 PHY (supports host and device)
- External EEPROM (programmed with 48-bit globally unique EUI-48/64™ compatible identifier)
- Audio codec with headphone out, microphone and line in jacks
- GPIO: 6 push-buttons, 4 slide switches, 5 LEDs
- Six Pmod ports (1 processor-dedicated, 1 dual analog/digital)

This processing power and IO will now have to be used to make the EAGLE smart.

2.1 IO on the ZyBo

To connect the IMU, the motor speed control, the camera, the remote control and the wireless video link, we need a lot of IO. Could you already figure out now, where you would connect your camera, where the wireless link? Quite challenging and complex system, many design choices to be made. We connect the camera board through the HDMI port, and the wireless link through the USB 2.0 port. Fig. 2 details already the IO used for EAGLE. If you want to understand better why, don't hesitate to ask the experts.

2.2 Programming and using the ZyBo

And where would you run all the needed software? The ZyBo can be programmed in VHDL (using Vivado) and C/C++ (using Xilinx SDK). So, for EAGLE, we will have *Hardware* and *Software* work to be done:

- *Hardware:* The ZyBo has a large Zync FPGA which can be programmed directly using a hardware description language (HDL). Although you are using software tools to implement the functionality, you are really programming hardware. A lot of the functionality for EAGLE is implemented directly on the FGPA, especially the functionality that requires a lot of (fixed point) signal processing, such as the wireless modem. You can open the EAGLE hardware project when you want to change something on the FPGA (e.g., complete the wireless communication). You can also open this Hardware project when you want to understand how the ports are connected from FPGA to pins or see what functionality was implemented directly on the FPGA. Tool: Xilinx Vivado 2016.2.
- *Software:* The Zync FPGA also contains two embedded ARM cores (CPU0 and CPU1). These run an operating system (e.g., Linux) and can be programmed using any software programming language you know (e.g., Python or C/C++). You can write any application (e.g., to control the drone, write an autopilot, analyse the incoming video stream). Tool: Xilinx SDK.

3 The motion sensors

The IMU (Inertial Measurement Unit) consists of various sensors that enable to measure motion of a device. An IMU detects the acceleration using one or more accelerometers. Inaddition, it detects changes in the position of the device (yaw, pitch and roll) using one or more gyroscopes. Yaw means going up or down. Pitch means going left or right. Roll means the system is not

Figure 3: Inertial Measurement Unit

Figure 4: Camera + odroid/pi zero: HDMI+USB stream

perfectly horizontal, i.e., left and right side are not at the same height. The EAGLE IMU also includes a magnetometer, which can be used for calibration against orientation drift. But this is not (yet) used in EAGLE.

The IMU [1] (see Fig. 3) gathers information from its sensors and writes it into its internal registers. By use of I2C or SPI-communication we can read out this information and write them into the memory of the ZyBo.

4 The camera

The camera is a simple IP camera (see Fig. 4) that outputs a compressed and uncompressed video stream. The uncompressed stream goes through the HDMI IO into the ZyBO, where it can be processed by the vision algorithms. The uncompressed video has 640x480 pixels per frame, and are processed in to CPU0 ARM core, which is a Linux core. The compressed stream goes through the Ethernet port to the FPGA, where it is modulated and transmitted by the wireless link module.

5 Remote control

A remote control is needed to control the EAGLE when it is not flying autonomously. Even when it is flying autonomously, there is a killswitch that will shut down the engines. The kill switch functionality is embedded in the FPGA, with some safety procedures, so it should work reliably! It should ovbiously only be used when the drone is really out of control.

Basically, there are three modes for the RC switch, which can be used to enable or disable the remote controlled navigation:

- manual flight;
- autonomous flight;

- wireless charging.

We make use of a Turnigy 9XR which transmits 9 channels combined in one PPM signal towards its receiver. You can search for *Turnigy 9XR PRO Radio Transmitter Mode 2* if you need more information about the remote controller.
In the receiver the PPM signal is decomposed in 9 independent PWM channels. Each PWM signal is captured in the ZyBo via the PMOD connectors and written in the RAM by a dedicated PWM-block.

5.1 Special functionality RC

The remote control has two dedicated switches:

1. Killswitch
2. Modeswitch

5.1.1 Killswitch

The killswitch is located in the upper left corner of the front panel (with ELE DIR written next to it). This two-position switch is a safety measure incorporated in the drone. It is directly connected to a hardware block on the FPGA: If the switch is flicked, the outputs towards the ESCs and the engines are being tied to ground. This safety feature is a necessity, students always need to turn this on when not operating the drone or when approaching the drone.
Additionally there are some more safety checks inside this hardware block: a pulse is transmitted from the software on the bare metal core, if for some reason this pulse is not transmitted (e.g. the software is in a deadlock or infinite loop) the drone kills itself, reducing dangerous behavior.

5.1.2 Modeswitch

The Modeswitch is located second to the upper right corner of the front panel, with F-E LND written next to it. It is used to be able to switch between three determined scenarios:

- Manual flight
- Automated flight
- Power transmission

5.1.3 Manual flight

When the switch is in the up position, the RC is being used to control the drone.

5.1.4 Automated flight

This mode is engaged with the mode switch in the middle position. First the values of the RC are copied to a new set of registers from which the control team gets its inputs (as to not have zero-values inside these registers). After that the camera starts transmitting its video stream to the Zybo and video processing on the Linux core is taking care of calculating new output values for the motors.

5.1.5 Power transmission

When we flick the modeswitch in its lowest stand, we engage the power transmission mode. All outputs for the motors are tied to ground (to limit interference) and another PMOD pin starts transmitting a PWM wave towards the on-board coils. This is only to be done once landed on the final land spot. Engaging this mode while flying will kill the motors and make the drone fall to the ground.

5.2 Arming procedure Drone

If the drone boots correctly, it is not able to fly immediately. First we have to arm the drone. This is done by keeping the left (throttle) stick on the RC in the lower right corner for some seconds (the second LED on the zybo should be lit when armed and extinguished when disarmed, or when a beeper is attached to pin JB7, two consecutive beeps will be produced).
When keeping the throttle stick in the lower left corner for a few seconds, the drone gets disarmed and no flight should be possible (again, sounder creates beeps and led is extinguished).

6 Local speed control

The engines of the drone need three-phased large PWM signals as they are current driven. Unfortunately the Zybo cannot give this current, therefore ESCs are uses as a connection module. They convert the 1 phase PWM signal of the ZyBo to a high-frequent and high- 3-phase PWM signal towards the engines. The three phases are required to be able to let the propellers rotate inside the magnets of the engine.
So to summarize: The ZyBo generates one PWM-signal for each motor, which is being captured by the ESC. The ESC then transforms it into a 3-phased PWM signal with the same duty cycle as the input, but higher in magnitude and frequency.

7 Power

In this section, we briefly introduce the batteries (handle them with care!), the power distribution and finally, the wireless power transfer of the EAGLE.

7.1 Power in: Battery

The main power source for the drone and its peripherals are LiPo batteries (see Fig. 5). LiPo batteries are common used in drone applications or mobile phones. They have a large capacity, almost no self-discharge, are capable of delivering high currents and have practically no 'memory effect'. Mostly they consist of different cells placed in series, typically noted by xS where x is a number representing the amount of cells in series. Care must be taken NOT to charge these batteries with standard power supplies or standard battery chargers: when not balancing the voltages, one cell can easily become overloaded and explode or catch fire! Therefore specific LiPo chargers must be used, which measure the voltage of every cell independently and make sure they are not being overloaded. Also you must not let the battery run empty, as this will also destroy your LiPo battery.
The batteries provided to you are 3S (11.1V nominal) batteries with a capacity of 5.2 Ah. When charging them always confirm that the charger recognised them as 3S batteries and set charge current to no more then 5.2A. Do not discharge the batteries at more then 52A (10x their capacity or 10C) for a long time (the drones have a 50 amp slow fuse)

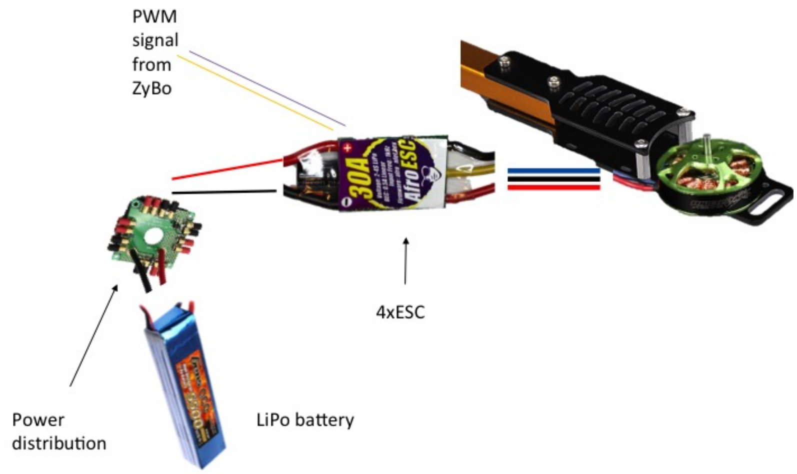

Figure 5: From the battery to the power distribution board to the ESC and engines.

To summarize:

1. Always charge the batteries with the correct LiPo chargers and use a lipo safety bag
2. Always unplug the batteries when not using and store them in the safety bag
3. never allow the batteries to run dry
4. never charge the batteries when there is no-one around.
5. always check correct charger settings : 3S (11.1V nominal) and max 5.2A charging current
6. Always connect the balance plug with the black wire on the - sign and with pin 4 unused

7.2 Power Distribution

It is clear by now that we need many boards that need to be powered by one battery. Some parts need 5V, the ESCs need 12V and lots of amps, therefore we use a separate board. The Power distribution board has two output circuits to be able to attach all necessary components at the right voltage:

- $12V_{DC}$:
  - ESCs: they need a lot of amps so are actually connected directly to the battery, the Power distribution board helps for a clean looking connection
- $5V_{DC}$:
  - ZyBo

– RC Receiver

All other boards can be chosen to be powered by the ZyBo: By using the 3.3V connections on the PMOD Connectors. The ZyBo itself can be powered by USB-connection or with an external power-supply (the battery in our case), it must be noted that if the ZyBo is powered by USB, it has less power to connect peripherals. When connecting lots of peripherals while powering through USB voltage brownouts may occur. This is easily notable since the ZyBo will reboot itself in that case.

7.3 Power out: Wireless power transfer

Your smart EAGLE's mission is to power as much LEDs as possible, by autonomously finding the spots where the inductive power circuit receiver is installed (with the LEDs). So, the goal is not to wirelessly power the EAGLE, but to have it transmit power to the LED strip! The more efficient your wireless inductive link is, the more LEDs you will be able to turn on! (See Fig. 6).

You will need to design 1) the transmitter circuit to mount on the EAGLE and fed by its 12V DC battery; and 2) the receiver circuit to connect to the LED strip. How close do you need to be to get the power transfer working? Does it still work in case of misalignment?

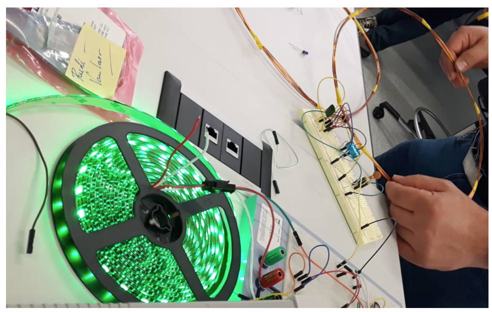

Figure 6: Wireless power transfer used to lighten LEDs up.

8 Conclusion

This document gives a very high level overview of the EAGLE hardware components, and just serves to get you started. By no means, you should understand all the details of all the parts, and the idea is now to *divide and conquer*. The work can be split in the design of 6 tasks, some requiring PCB design, others FPGA programming and some only software and algorithms. If you complete all parts, and add them to the EAGLE frame, your system will ook like the one represented in Fig. 7.

References

[1] LSM9DS1 Data sheet, "iNEMO inertial module: 3D accelerometer, 3D gyroscope, 3D magnetometer", from *https://www.sparkfun.com/products/*13284 Sept 2016.

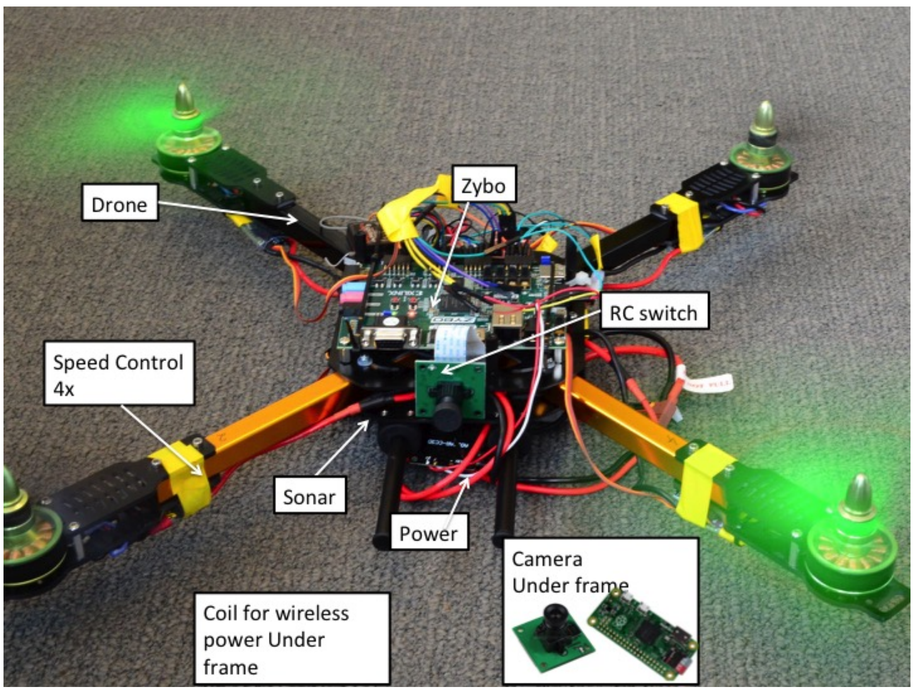

Figure 7: This is how the complete EAGLE looks like.