EEP55C21 Cyber-Physical Systems and Control

1 Introduction

This document outlines the overall practical element of the module, along with what's expected from the students regarding their deliverables.

2 Student Deliverables

Students have the following deliverables to demonstrate to the demonstrators/teachers for grading:

- Build a small microcontroller-based controller to fly the drone
- Create and implement a more advanced controller of their choice to fly the drone
- Develop a 2D map and a path-finding algorithm for the drone to navigate from two points on the map in simulation
- Implement the path-finding algorithm in an experiment to help their drone navigate from two points on a real map
- Write a report on their own controller and their path-finding algorithm implementation

3 Required Software

3.1 Arduino IDE

The Arduino IDE is required for programming your microcontroller for Deliverable 1. The link for downloading it is here. There will likely be additional libraries that you will need to install, but you can do that through the IDE.

3.2 Python3

The software for running the drones requires at least Python3.8 to be installed, but can up to version 3.12. Here's a link for downloading.

3.3 Bitcraze CFclient/CFlib

The CFclient can be installed on most major operating systems. Here is a link for installation. Installing the client will also install the lib. You would only install the lib on its own if you have no intention to use the client's GUI.

The client/lib is made for linux, so there is also an option to install a virtual machine, though it has been tricky to use at times, especially if trying to update parts of the client. It is recommended that you avoid using this version.

3.4 MATLAB (Optional)

This is optional for Deliverable 3. It's installed on all the PC's in the lab, but it is possible to just use the web browser version.

3.5 Text Editor (Optional)

Any text editor such as Notepad++ or Sublime will do for this.

4 Deliverable 1: Microcontroller-based Control

The students will be put into groups of 2 or 3 for the first deliverable and will be given a kit containing the following:

- Crazyflie Quadcopter
- Crazyradio Dongle
- Adafruit ItsyBitsy 32u4 - 5 V 16 MHz
- DFRobot Gravity Analog Rotation Sensor
- Seeed Studio Grove Thumb Joystick
- Breadboard

4.1 Crazyflie Quadcopter

Figure 1: Crazyflie Quadcopter

The Crazyflie 2.1 is a versatile open source flying development platform that only weighs 27g and fits in the palm of your hand. it is equipped with low-latency/long-range radio as well as Bluetooth LE. In combination with the Crazyradio PA dongle, you can use your computer to display data and fly with a game controller. It has an extensive ecosystem of software and deck expansions that allow the drone to fly in different environments.

4.2 Crazyradio Dongle

The kits provided will have either one of the following dongles:

- Crazyradio PA
- Crazyradio 2.0

As far as the module is concerned, there is no real difference between them.

4.2.1 Crazyradio PA

Figure 2: Crazyradio PA

Crazyradio PA is a long range open USB radio dongle based on the nRF24LU1+ from Nordic Semiconductor. It features a 20dBm power amplifier, LNA and comes pre-programmed with Crazyflie compatible firmware. The power amplifier boosts the range, giving a range up to 1km (line of sight) together with the Crazyflie 2.X and up to 2km Crazyradio PA to Crazyradio PA (line of sight).

The hardware comes shipped with the latest firmware as well as a bootloader that enables firmware upgrades via USB without any additional hardware needed.

4.2.2 Crazyradio 2.0

Figure 3: Crazyradio 2.0

The previous dongle was discontinued and replaced with the Crazyradio 2.0, which enables the Crazyflie to easily connect to a computer for more advanced use-cases. It is delivered with a USB bootloader, where you can easily drag and drop your firmware. Here's information on how to update the dongle firmware.

Although designed for our Crazyflie products, the USB dongle can be used for a range of applications since custom firmware can be flashed on it.

Note: Crazyradio 2.0 is not compatible with custom firmware made for the Crazyradio PA nRF24 MCU (for instance various security firmware projects) since it's based on a newer nRF52 chipset.

4.3 Adafruit ItsyBitsy 32u4 - 5 V 16 MHz

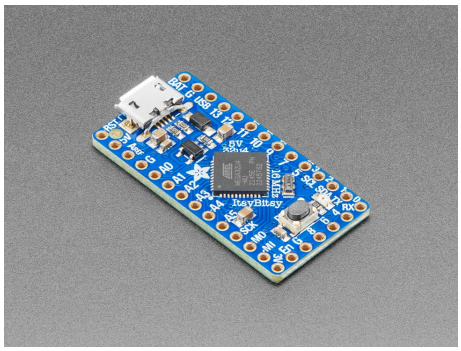

Figure 4: ItsyBitsy Microcontroller

The ItsyBitsy is a microcontroller board that has 6 power pins, 6 analog/digital pins and 17 digital pins. It uses the Atmega32u4 chip. It runs at the same speed and voltage of an Arduino UNO or Leonardo. It is programmable using the Arduino IDE.

The reason for its use is that it is a cheap alternative to Atmega32u4-based Arduinos. The chip itself can be programmed as a Physical Interface Device or Human Interface Device, which operating systems recognise computer joysticks as, whereas the more common Atmega328p and many other microcontrollers cannot.

4.4 DFRobot Gravity Analog Rotation Sensor

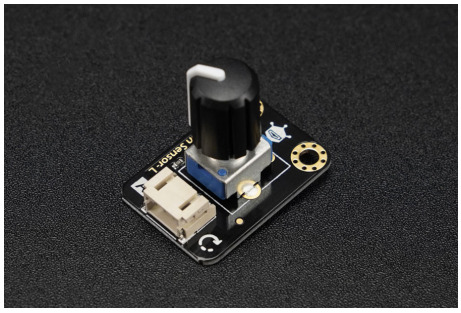

Figure 5: Analog Rotation Sensor

This is an rotation potentiometer that is able to rotate up to 300-degree. It can be very easy to achieve position-dependent interaction with the rotating effect to control the drone's thrust. So with this sensor, you can learn how rotary potentiometer works and what analog signal is. You can have much fun with this basic input device.

4.5 Seeed Studio Grove Thumb Joystick

The Thumb Joystick is a module, which is very similar to the 'analog' joystick on PS2 (PlayStation 2) controllers. The X and Y axes are two 10k potentiometers which control 2D movement by generating analog signals. The

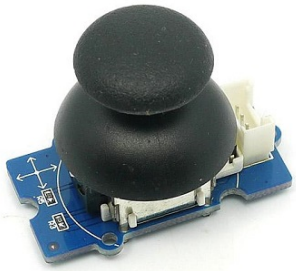

Figure 6: Thumb Joystick

joystick also has a push button that could be used for special applications. When the module is in working mode, it will output two analog values, representing two directions. This can be used for the pitch and roll of the drone.

4.6 Breadboard

This sections demonstrates the layout of a breadboard. This figure below shows how the top and bottom of a breadboard looks. Note that there are metal contact strips for each column with a ridge in the middle. In order to prevent the chips from shorting, they must be placed in the middle with both sides being separated by the ridge. This means that each pin has its own column for components and wires to connect to it. The two long horizontal strips are used as rails for the supply voltage and ground. Your supply voltage is connected to red rail, while the blue rail is connected to ground. Wires are used to connect the strips at the top of the breadboard to the strips at the bottom.

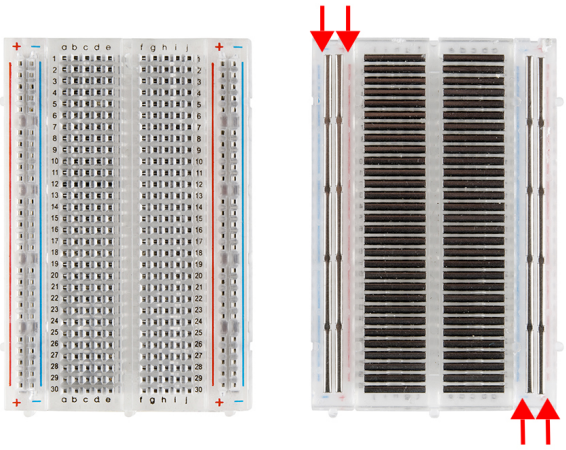

Figure 7: Breadboard Back and Front

4.7 Student Objectives

Students are expected to make a circuit with the above parts and programme the ItsyBitsy to interpret the sensor and joystick interactions as commands for the drone to move. They aren't restricted to these parts and can consult demonstrators/teachers for extra parts if needed. The big thing to get across to the students is that the ItsyBitsy needs to be recognised as a HID/PID by their computer, so that should influence how they approach that problem.

Seperate to the ItsyBitsy programming, a python script needs to be written to help communicate these commands from the circuit to send to the drone. The block diagram below give an overview of the system, where

the game controller represents the sensor inputs into the microcontroller.

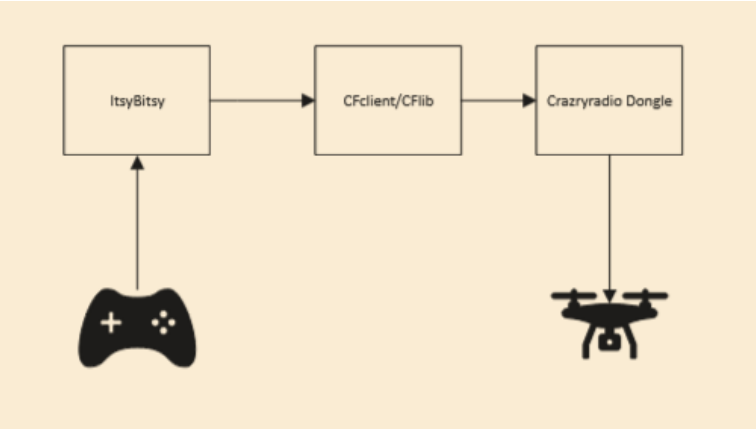

Figure 8: Controller Block Diagram

The students are also free to just use the CFclient GUI to interpret their controller inputs. Here's information on how to work a game console controller.

Their drone should be able to make basic movements:

- Pitch
- Roll
- Thrust
- Takeoff

Yaw movements are not required, so is an optional function.

The day for demonstrating this for assessment is the Monday lab session of Teaching Week 3.

5 Deliverable 2: Advanced Controller

Once students have completed their first deliverable, they are put into groups of 3 or 4. This deliverable builds on their knowledge of how to interface with the CFclient/CFlib and requires them to make a more sophisticated controller for their drone.

This can be based on their first controller or it can be a completely new design involving no hardware. The main objective for the students is to be as creative as possible. This is where student have most fun and involves little help from the demonstrators/teachers.

Their drone should be able to achieve the same movements are their first controller. This means that the assessment is basically the same with the exception of judging how novel their controller is.

The deadline for demonstrating this for assessment is the Friday lab session of Teaching Week 8.

6 Deliverable 3: Map Generation and Path-finding Simulation

Students are required to randomly generate a 16*16 grid map with a white path and a black background. The path generated should have at least one divergence in it to give the drone options to get from start to finish.

They have the option of choosing how to generate the map and how to implement their path-finding algorithm. The algorithm is chosen by the students themselves. The most obvious options are either sticking with Python or choosing to use MATLAB.

The students will be assessed on this part on the Monday lab session of Teaching Week 10.

7 Deliverable 4: Path-finding Implementation on Crazyflie

The students then take what they have done in simulation and implement that using the following kit:

- Crazyflie Quadcopter
- Crazyradio Dongle
- Flow Deck V2
- Lighthouse Positioning Deck

The students are asked to retain their dongles due to a shortage, while everything else is provided for them. Given that the Crazyflie and Crazyradio dongles have been covered, there's only the two decks to discuss here.

7.1 Flow Deck V2

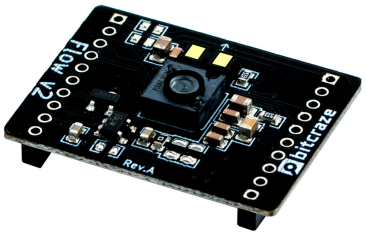

Figure 9: Flow Deck

The Flow deck gives the Crazyflie 2.X the ability to detect its motions in any direction. The VL53L1x ToF sensor measures the distance to the ground with high precision and the PMW3901 optical flow sensor measures movements of the ground.

This deck is placed on the bottom of the drone to allow for the two sensors to work and give proper readings to the drone. The optical flow sensor can give different output readings for dark and light surfaces underneath. This is why the path generated is white and the surrounding area is black.

7.2 Lighthouse Positioning Deck

The Lighthouse positioning deck allows for high precision autonomous flight. The Lighthouse deck uses the SteamVR Base Station 2.0 (aka Lighthouse V2) to achieve high precision positioning. The deck has 4 receivers which gives the full pose of the Crazyflie.

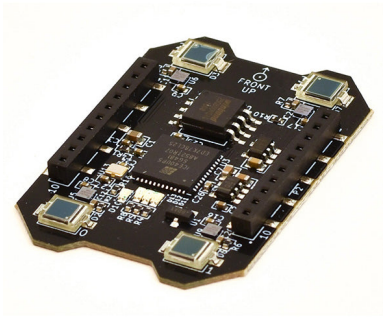

Figure 10: Lighthouse Deck

The deck is attached at the top of drone to allow interaction with the base stations using the four receivers. The position and pose is calculated directly in the Crazyflie, which makes it possible to create fully autonomous systems without external communication.

The students will need to calibrate the drone relative to the base stations in the lab to allow for accurate readings. The CFclient GUI has a step-by-step process for this.

7.2.1 Lighthouse Base Station

Figure 11: Lighthouse Deck

There are three base stations in the lab. They each emit and infrared light in a sweeping motion. Think about the way you might see scanning lasers in sci-fi movies and you get what's happening here. The four receivers calculate the angle at which they receive infrared light from the base stations and use Kalman Filtering to give an estimation of its state.

More information can be found here.

7.3 Implementation

Students need to write a python script to move the drone based on both sensor data and the algorithm they have developed. Information on how to get started can be found here and here. Here is also a GitHub from Bitcraze that has example code for different applications and decks.

The assessment for this will take place at the final lab session of the term.

8 Deliverable 5: Report

Each group submits a report detailing and illustrating their advance controller and their pathfinding algortihm simulation and implementation. Students are also expected to submit all relevant code for this. At most these reports should be 15 pages including diagrams.

The deadline for this is usually some point before students leave during the Christmas break.