02135 - Assignment 3
Getting started with the Huzzah32 board and with MicroPython

Edward Todirica and Luca Pezzarossa [lpez@dtu.dk]
March 11, 2022

1 - Introduction

This document describes the third assignment of the course 02135 – Introduction to Cyber Systems. The goal of this assignment is to get familiar with programming an embedded computer and controlling various I/O devices. The embedded computer used in this course is a Feather Huzzah32 developed by Adafruit. We are going to program it using MicroPython, which is an embedded version of the Python language that you have used so far during the course.

Similarly to the previous assignments, this assignment should be carried out in groups of two or three people. The groups can be different from the ones of Assignment 1 and 2. In this assignment, you are **not** required to prepare a report. You should only **give a DEMO of the tasks to the TA or to the teacher and hand-in your code**.

All the material related to this assignment can be found on the DTU-Inside course page at the following location:

DTU-Learn/Course Content/Content/Assignments /Assignment 3

The deadline for this assignment is **Monday 28th March 2022 at 23:59**. By this date, you must have given the DEMO and handed-in your code as a ZIP archive using the assignment utility in the DTU-Inside course page at the following location:

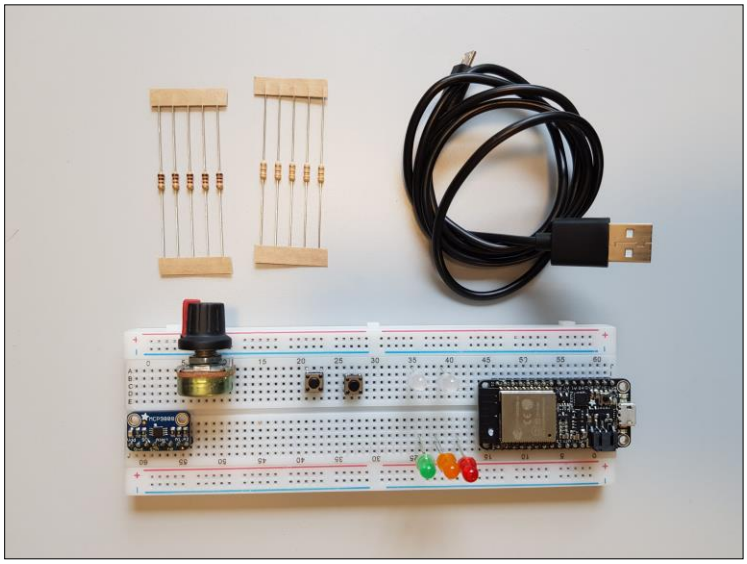
DTU-Learn/Course Content/Content//Assignment 3

Remember to structure your code such that it is clear what the different parts are doing and to document your code with comments and, if necessary, a README file with instructions on how to run it.

Feel free to ask or send an e-mail to the teacher if you have questions regarding practical matters and the assignment in general.

2 - The provided kit

For this lab, each group is receiving a kit. You will use the same kit in Assignment 4.



At the end of the course, you must return the kit. You can do this during the last lab session or by making an appointment at *lpez@dtu.dk* for drop-off. **When returning the kit, the components MUST be placed on the breadboard as shown in the figure (except for the resistors). Please fill in the form included in the kit when returning the kit. Please write if some components are broken or lost.**

The content of the kit is:

- 1 x Feather Huzzah 32
- 1 x Temperature sensor MCP9808
- 1 x 10K potentiometer
- 2 x Push buttons
- 3 x LEDs (green, yellow, red)
- 2 x Neopixel LEDs
- 5 x 1kΩ resistors (for pull-up/down)
- 5 x 330Ω Resistors (for LEDs)
- 1 x MicroUSB cable

All the Huzzah32 boards are ready to use, and the latest version of the MicroPython interpreter is already installed. On DTU-Inside, you can find a document named "Huzzah32 notes" containing instruction on how to interact with the board. In addition, we recommend you also take a look at Adafruit intro to Huzzah 32- ESP32.

3 - Assignment tasks

The assignment consists of 5 mandatory tasks. Solve as many of the following tasks as you can during the lab session time. You will have 3 lab sessions to do these exercises in weeks 7, 8, and 9 of the course. The assignment tasks that you do not finish during the lab sessions must be done as home assignment. Remember that you have to demo all the tasks to the TAs or to the teacher.

Task 1

Make a red LED blink with a frequency of 1 Hz (500ms on/500ms off) as long as you press a button. For this exercise, you can use the 5mm red led, a resistor (330 Ω), and a button. For more information about using the button and the led you can see these references:

- MicroPython Basics: Blink a LED
- MicroPython Basics: Digital I/O
- Pinouts Huzzah32 board
- Working with GPIO pins

***Note**: In the Adafuit tutorials, it is mentioned that for setting an output pin high and low you are supposed to use the high() and low() functions. You should use the value() function instead. Make sure that you are using the GPIO pins available on the Huzzah32 board (the tutorials from Adafruit are referring to a previous version of the Huzzah board).*

Task 2

Connect a button and all the three 5 mm colored LEDs (green, yellow, and red) to the Feather Huzzah32. **Remember that for each LED, you should use a resistor as well.** Write a MicroPython program that each time you press the button will light up one LED in a cyclic fashion. When the program first starts, only the green LED is lit. After the first button press, only the yellow LED is lit. After second button press, only the red LED is lit. After the third button press, the green LED is lit on and the cycle repeats. In other words, after each button press, the next LED will light up. **Please note that if the button is kept pressed, it still counts as 1 press and the LEDs should not change until the button is released and then pressed again.**

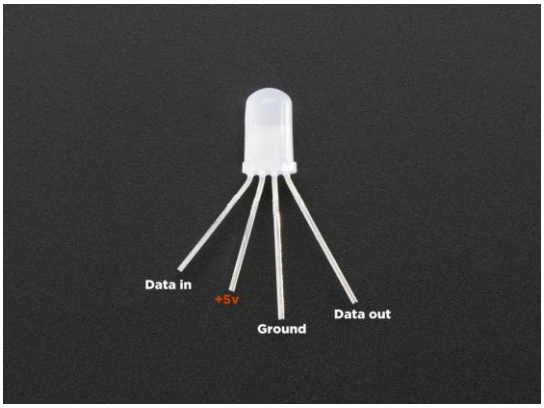
Task 3

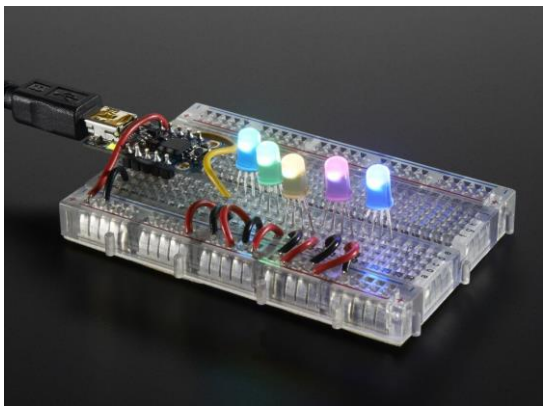
Keep the green, yellow, and red LEDs connected to the Feather Huzzah32. Connect also the temperature sensor MCP9808 to the Feather Huzzah32. Write a MicroPython program where only one of the LEDs is lit at a time depending on the temperature read from the temperature sensor. For example, when the sensor is at the room temperature, only the green LED is on. When you touch the temperature sensor, the temperature read from the sensor will start to rise and when it reaches a threshold value only the orange LED should be on. If you continue to keep the finger on the temperature sensor, it will reach another higher threshold value and only the red LED will be on. If you stop touching the temperature sensor, then the temperature will start falling and then only the orange or green LEDs will be on. You should find some suitable threshold values so that you can easily control which LED is turned on. For this task, you can use this tutorial as a reference.

***Note**: Make sure to use the correct I2C pins for the Huzzah 32 board as described here. Remember that you also need to add pull-up resistor for SDA (but not for SCL).*

Task 4

For this task, we will implement the same functionality of Task 3, but instead of using the 3 colored LEDs we will use two Neopixel LEDs. This time, both the Neopixel LEDs must show the same color, which changes depending on the current temperature read from the sensor (green, yellow, and red, just like in Task 3). In the images below, you can see the pinout and an example of how the Neopixel LEDs can be connected together.





You can find out more about Neopixels from these references:

- 5mm neopixel LED product page
- Controlling neopixels
- Adafruit NeoPixel Überguide

Task 5

In this task, we will connect an analog input to the Huzzah32. The analog input is provided by the potentiometer available in your kit. Write a MicroPython program that can control the intensity of light of the two Neopixel LEDs with the help of the potentiometer connected to one of the Analog input of the Huzzah32 board. By turning the potentiometer, you should be able to control the brightness of the LED from 0 (off) to full brightness. For solving this task, you might find it useful to take a look at these references:

- MicroPython Analog To Digital Conversion