

# Milestone-3

You will now combine the work done in the previous two milestones. After completing this milestone, the Raspberry Pi will send a collection of commands to control the car and simulate driving through a predetermined path. These should be sent to the HiFive board using the UART serial communication configured in Milestone 2.

## Part 1: Modifying the HiFive Code

You should first download the source files for Milestone 3 and add your function implementations from Milestone 1 into the file `'eecs388_m3.c'` located in the `src` folder. Reuse the python file containing the serial communication code you wrote for Milestone 2 on the Pi for this milestone as well.

In Milestone 2, you had set up the HiFive to receive strings via UART from the Pi and send it back verbatim. For Milestone 3, you will need to modify the `'eecs388_m3.c'` file to use the values received from the Pi to set the steering angle and driving speed and direction. You need to implement a loop that receives a series of (string) commands from the UART, converts them into usable integer values (for speed, angle and duration) and feeds these into the steering, drive and delay functions.

Two helpful C functions for the above task are `'strncmp'` and `'sscanf'`. There are two useful components of the following code example:

1. the lines up to the `strncmp` function perform a 5-character comparison between the strings "Hello World!" and "Hello" - since the first 5 characters are the same, the function returns 0 into ret.
2. the last few lines demonstrate how to pull an integer value out of a string. Here we have the string "Number:388" and we want to assign `val` to be 388. The `sscanf` function

call accomplishes this (the number starts at the 7th index which is why we add an offset of 7 to str).

```
char * lhs = "Hello World!";
char * rhs = "Hello";
int count = 5;
// Zero if lhs and rhs compare equal, or if count is zero.
// This would return 0 since the first five characters are the same
int ret = strncmp(lhs, rhs, count);
char * str = "Number:388";
int val;
// This assigns the int 388 to val
sscanf(str+7, "%d", &val);
```

## **Part 2: Follow a predefined path**

For the final demo, we will provide you with a **csv** file that contains commands in the following format:

**<Angle>, <Speed>, <Duration>**

- The angle value will be an integer between -45 and 45.
- The speed value will be 1, 2 or 3 for driving forward (at different speeds), -1, -2 or -3 for backward and 0 for stop.
- The duration will be the time, in *seconds*, for which that command should run.

The following input file, for example, stops for 5 seconds with 0 degrees turning, drives forward for 5 seconds, drives reverse for 6 seconds, drives forward for 2 seconds with a 45-degree turn, drives forward for 2 seconds with a -45-degree turn:

```
0, 0, 5
0, 1, 5
0, -2, 6
45, 1, 2
-45, 1, 2
```

### **Submission:**

Two weeks will be provided to complete the milestone from the published date. After completion, each team member should submit the **eecs388\_m3.c** file to Canvas.