



AUTONOMOUS EMBEDDED CONTROL

RECRUITMENT TASK

Cairo University Eco-Racing Team

2023



CAIRO UNIVERSITY ECO-TEAM



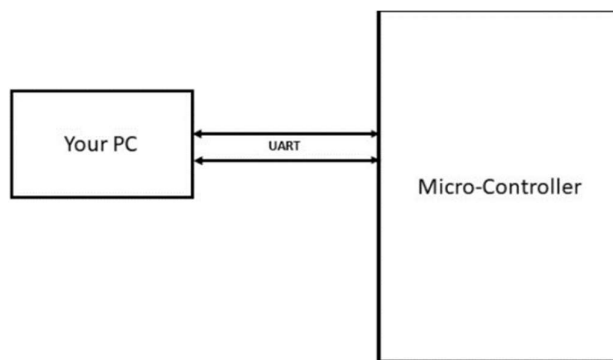
Cairo University Eco-Racing team applies every year for the Shell Eco-Racing 2023 Marathon competition, so it aims to create an autonomous car, by creating an embedded control system that controls the ICE (internal combustion engine) vehicle, and this system contains roughly the following:

1. One DC motor to control the speed of wheels.
2. One stepper motor to move the wheels angle right and left.
3. A decision-maker unit (out of scope) which sends its decisions to our micro-controller.
4. A dashboard to show out the current state and the car parameters, that are always up to date.

This task will focus on your coding skills in writing efficient, clean and robust embedded-c code, and your ability to integrate multiple blocks that work in harmony in a single design.

Statement:

Implement a simplified version of said system, that receives a data frame, through a UART-PC communication.



Your PC endpoint should be modeled as a virtual terminal that takes the communication frame and maps its values into speed, direction (forward or backward) and orientation of the autonomous car.

EX:



The UART-PC communication frame will be as following:

XXXAYYBE

1. XXX -> for the speed of the motor, controlling the speed of the wheels.
(0 -> 100)
2. A -> for the direction of the motor, controlling the speed of the wheels.
(F or B)
3. YY -> for the angle of the second motor, controlling the angle of the wheels
(0 -> 45)
4. B -> for the direction of the second motor
(R or L)

(with the north of the car being the reference)

5. E -> is an indicator of the end of the frame.
(if the user doesn't write E at the end of the frame, then it's a wrong frame and ignore it)

EX:

- I. **076F18LE** -> means the car should change its speed to 76% of the maximum speed, the direction of movement is forward, the wheels should tilt to the left from its axis by 18 degrees.
- II. **76F18RE** -> This frame looks fine but it violated the standard length we agreed on.
- III. **F100R09E** -> This frame is wrong because it's not matching our standard.
- IV. **076F18R** -> This frame is very correct, but unfortunately, When E is not included, we should consider the frame as noise only should be ignored.

General Assumptions:

1. In case of a DC motor, assume: (ACW -> forward movement, CW -> backward movement).
2. In case of stepper motor, assume: (each step = one degree for simplicity).

Required:

You should **enter the speed, direction and orientation**, from your PC/laptop keyboard through the virtual terminal by using the above-mentioned communication frame, and **the motors should adjust to the entered speeds and directions**. In order to implement a simplified version of the dashboard, **show on LCD the speed percentage and the angle of wheels** (+ve angle means at the right of the car's north, -ve angle means at the left of the car's north), and the time, the time should start from 00:00:00 AM (following the hour, minutes, seconds format).

EX:

```
=====
|| Speed: 76% Direction: -18 ||
|| Time: 00:00:28 AM      ||
=====
```

Deliverables:

1. Zip file containing the following:
 1. A folder named "Code", containing the following:
 - I. All .c and .h files only. (with no compilation/project/IDE files).
 - II. One text file for any assumptions you made. (optional).
 2. A file outside the folder named "Video", that contains a screen recording showing the system working and running on Proteus, while using the virtual terminal feature and testing the system using different data frames. (And explain your work by your voice). [Maximum 15 min, and minimum 5 min]
 3. In case you implemented the drivers by yourself (It's not required but preferable), Create the following folders:
 - a) Code
 - i. MCAL folder
 1. Inc folder
 - a) All .h files
 2. All .c files
 - ii. HAL folder

1. Inc folder
 - a) All .h files
 2. All .c files
- iii. SERVICES folder
1. Inc folder
 - a) All .h files
 2. All .c files
- iv. APP folder
1. Inc folder
 - a) All .h files
 2. All .c files
- v. LIBRARIES folder
1. Config folder
 - a) Inc folder
 - i. All .h files
 - b) All .c files
 2. Common folder
 - a) All .h files

2. Deliverables Email: cuert.autonomous@gmail.com

Email Subject: YourFullName - Autonomous Embedded Control Task - 2023

Task deadline: 13th of September, 11:59 PM

If you have any questions regarding the task or the deliverables, please contact us on the same email with:
Email Subject: Task Enquiry

Goodluck!