GPS Tracking System

Project Description

In this project you will develop a GPS tracking system using TM4C123G LaunchPad, the system should store the moved trajectory after power-on until a destination point is reached and calculate the taken distance.

There are 4 possible options for the definition of the destination point:

1. Once the moved distance exceeds 100m, stop adding new points to the trajectory and the last point added to the trajectory is the target destination.
2. Stop adding new points when a push button is pressed, and the last point added to the trajectory is the last destination.
3. Predefine the destination point in your code and when the system reaches this point, stop adding new points to the trajectory.

Other functions required by the system:

1. When the destination point is reached the build-in LED of the launchpad should be turned on.
2. Display the calculated distance on 3 7-segment displays. You can update the 7-segment display continuously or once the destination point is reached.

Functional Requirements

1. After power-on, the system should configure the following interfaces:
    1. UART to interface with the GPS module
    2. Digital Output for built-in LED
    3. Digital Output for 7-segment displays

2. The system should read GPS module data and wait until there is a GPS fix (Check GPS module datasheet to check how can this be done).

3. The first point read with GPS fix should be appended to the trajectory as the start point.

4. Your system needs to read the coordinates from the GPS module in a periodic manner to get your trajectory. Software delay or Systick timer should be used to control rate of execution of main function depending on the update rate of the GPS module.

5. After reaching the destination point, the system should stop appending new points.

6. The output will be translated as the following.
    1. Stage 1: The built-in LED will be turned on when the target destination is reached or when you exceed 100 meters.
    2. Stage 2: The taken distance will be displayed on 3 digits 7 segments.

7. The trajectory of the distance should satisfy the following criteria:
    1. The total distance between the start and the end point should be > 100 meters.
    2. The path from the start point to the end point should form a non-straight line that is similar to the provided baseline path below.

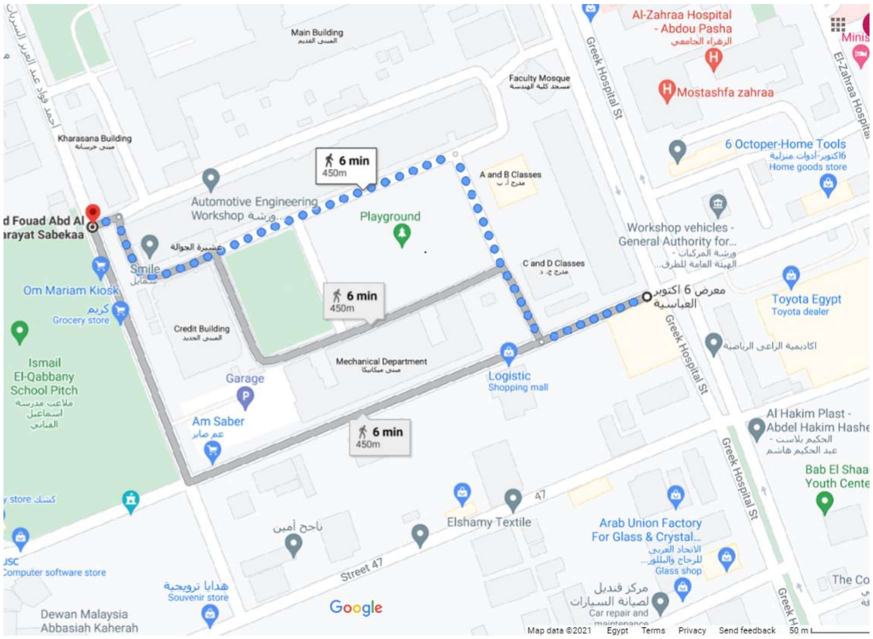

Figure 1:Basline path that you should follow

3. You have the freedom to select any starting point on google maps.
4. Your software should calculate the total taken distance, and you should compare the calculated distance with the distance obtained from google maps.
5. You should ensure that there is no big difference/deviation (error margin should be <= 5%) between your calculated distance and the one shown by google maps.

8. The scenario will be as the follows:

1. The user will open google maps and start his program running on TivaC at the same time.
2. The user should walk holding the kit and the opened google maps and walk till he/she exceeds the 100 meters.
3. The user should follow the path which is given in Figure1 and walk in non-straight line.
4. Once the user reaches the destination point (according to definition chosen from the 3 possible choices stated before), the LED will be turned on and the 7-segment displays should be updated with the calculated distance if they are not updated continuously.

Milestones

First milestone

- You should make sure that you can flash your code from the IDE (i.e. Keil) to your kit.
- You should implement a function that initializes the ports of your microcontroller.
- You should implement a function that turns on the LED when the distance exceeds 100 meters.
- You should implement the function that will display the output distance on the 3-digit 7 segments and test it by writing on them a dummy data.
- You should implement a function that calculates the total taken distance.

Second milestone (Final)

- You should write a function that configure the UART of your kit to communicate properly with the GPS subsystem.
- You should write the function that parse the coordinates sent from the GPS in the form of ASCII.
- You should integrate the developed functions to form your program and test it.

Number of Students

The project team should be between 4-7 members.

Project Instructions

1. Download the kit header file from the below link to include it to use its defined macros in your code.
   https://drive.google.com/file/d/1Gyt1VkYgfyEYHeF1VL6ivI9W2FQkB-GQ/view?usp=sharing
2. Your implementation should be in embedded C.
3. Do not use any built-in drivers in your embedded C code. You should implement everything from scratch.
4. The demo video should be taken as one shot without cuts or edits.
5. Mixture of the teams between CSE, ECE, and EPM departments is allowed. However, the mixed teams will be subjected to the same evaluation criteria of 2nd CSE/ECE project.

First Milestone Delivery

1. The first milestone is expected to be delivered on 6th June at 11:59 pm.
2. The expected to be delivered:
   - Source code compressed in one zip file.
   - GitHub link for your code.
   - A pdf file contains the team member names and their IDs and screenshots for the output of testing of the LED and the 3-digit 7 segments.
   - This will be submitted through this link.
     https://docs.google.com/forms/d/e/1FAIpQLScdR7mIfR3J6I-9R3sD_6CgXKPDscYKeSu5oOgOkBFiqnMyNQ/viewform?usp=sf_link

Final Delivery (second milestone) and project discussion

1. The team should deliver source codes compressed in one zip file.
2. The team should deliver a video for the project. Upload your video on the drive and attach the **video link in the submission form**.
3. The video should be one shot showing you when you are walking from the start point till the target point showing that
   a. you exceeded the 100 meters,
   b. you walked in non-straight line,
   c. the output of the LED,
   d. the output of the 3-digit 7 segments,

4. The team should push their codes on **GitHub repository**. Each team member should **contribute** and push **his/her part of code on GitHub**.
5. The team leader should attach the **GitHub repository link of the team in the submission form**.
6. A project discussion will be held.

Final Delivery Deadline (second milestone)

1. The deadline of the submission will be 13th June at 11:59 pm.
2. The project delivery files will be submitted through this link. https://docs.google.com/forms/d/e/1FAIpQLSevFyVUW36p0hb-Tv651MgRXETCmSZDA0LmpWQQ2nPVe1L1hQ/viewform?usp=sf_link

3. The initial dates for live demo will be held between 14th June and 16th June

Evaluation

1. 25% of the marks for **individual** contribution specially the GitHub repository contribution.
2. 75% of the marks for the project team.

Note: A team member without contribution on GitHub repo will get ZERO.