v. 2 2 April 2020

ENCE361: Personal Fitness Monitor Project

As a result of the COVID-19 lockdown, this project has been revised to reduce workload. Milestone 2 has been removed and its requirements have been pushed back to near the end of semester 2, replacing the Project Demonstration assessment. The functionality related to fitness goals has been removed. Students are now encouraged to use a predefine step counting criterion, namely:

Every time the norm of the acceleration increases from less than 1.5G to greater that 1.5G another step is counted.

Additions to the original version of these specifications are underlined, specifications that have been removed are ~~struck through~~.

Introduction

The goal of this project is to program a personal fitness monitor to record a user's steps and estimate the distance travelled by the user.

The fitness monitor will be powered by a TIVA microcontroller connected to an Orbit BoosterPack peripheral board. ~~Both boards are available as a set from the Central Library Copy Centre.~~ The main project output will be a program for a TIVA board that contains a small, interrupt-driven kernel able to record and display activity data via a graphical user interface.

Program Specification

The developed program will be assessed against the following specifications:

1. The program should estimate the number of steps taken by a user based on data from the accelerometer on the Orbit board.
2. The program should estimate the distance travelled by the user.
3. ~~The program should allow the user to set step- and distance-goals and should display progress towards these goals.~~
4. ~~The program should notify the user when a goal is completed.~~
5. The RESET button on the TIVA board should restart the program and clear any stored step, distance and goal values.
6. In addition to the requirements above, the display, buttons, switches and potentiometer on the TIVA and Orbit boards should function according to the requirements given ~~in Milestone 2~~ below:
   - 6.1. At startup the OLED board should display the number of steps counted since the last reset.
   - 6.2. Pushing the LEFT or RIGHT buttons on the Orbit board should toggle the display between total distance travelled since last reset and number of steps counted since last reset.
   - 6.3. The total distance travelled should be calculated from the number of steps multiplied by 0.9 metres.
   - 6.4. When the distance travelled is displayed, pushing the UP button on the TIVA board should toggle the units between kilometres and miles.

6.5. A long press on the DOWN button when the number of steps are displayed should reset steps to zero; similarly a long press on the DOWN button when the distance travelled is displayed should reset the distance travelled to zero.

6.6. Power cycling the board should reset step count and distance travelled to zero.

6.7. Setting SW1 to the UP position should put the fitness monitor in a test mode, where the functionality of the GUI can be verified. In this test mode each push of the UP button should increment the step count by 100 and the distance by 0.09 km. Likewise, pushing the DOWN button should decrement the step count by 500 and the distance by 0.45 km. Note that the other functions of the UP and DOWN buttons, namely toggling units and resetting counts, should be disabled while SW1 is UP. The functionality of the LEFT and RIGHT buttons should not be affected by SW1. Setting SW1 to DOWN should restore the normal functionality of the UP and DOWN buttons.

7. When the program starts, which may happen after a ‘reset’ operation or after reprogramming, the step and distance values should be zero~~ and the goals should be returned to default values, namely 1,000 steps and 100 m~~.

8. The program should have a real-time foreground/background kernel operating on a round-robin basis for background tasks. Robust behaviour should be maintained at all times.

Assessment

The project is worth 40% of your final grade for ENCE361. Marks for the project will be awarded on a group-by-group basis. To maximise the chance of successful completion, the project is structured in terms of three sequential milestones, as follows:

Component	Due	Weight (as % of ENCE361 grade)	Notes
Milestone 1	Week 5 (16–20 March)	4	Assessed during lab time
~~Milestone 2~~	~~Week 8 (28 April–1 May)~~	~~8~~	~~Assessed during lab time~~
Demonstration	Week 11 (18–22 May)	10	Assessed during lab time
Report	29 May 23:59*	~~8~~ 12	Dropbox on Learn
Code	29 May 23:59*	~~10~~ 14	Commit to Gitlab

*Assessment due dates cannot fall during exam study period. Marking of reports and code will commence at start of business on Tuesday, 2 June.

Grades for the milestones will be awarded for how closely the milestone specifications are met (see separate Milestone Specification documents). Grades for the demonstration will be awarded for how closely the Program Specification is met ~~and for the accuracy with which step count and distance travelled are estimated~~. Grades for the code and report will be awarded according to the following rubric:

Code	Marks	Report	Marks
Comments and layout	7	Identification of tasks & analysis	5
Modularity and structure	7	Schedular and inter-task communication	6

Code	Marks	Report	Marks
ISRs	7	Design and justifications	8
Scheduling of BG tasks	7	Conclusions and overall report quality	5
Total ÷ 2	**14**	**Total ÷ 2**	**12**

Group Work

The project is designed to be completed by groups of three. Students can sign up to a group by following the relevant link on Learn, with group selection open until 23:59 on Monday, 9 March. Students not allocated to a group of three after the deadline will be randomly assigned to groups on Tuesday 10 March. Even after this reallocation, a small number (up to two) groups in each lab stream may have only two members each. Group membership cannot be changed after the assessment of Milestone 1.

Group members should ensure that they have at least one TIVA-Orbit board between them.

A git repository for each group will be created at https://eng-git.canterbury.ac.nz; you are encouraged to commit code regularly to this repository. The portion of the project mark relating to source code will be based on code checked out from your repository at the project deadline. Remember to include your group name [**`FitnessDddttGroup#`**] on all source files, in your lab book, and on your project report, where ***`Ddd`*** *`∈ {Mon, Tue, Wed}`*, ***`tt`*** *`∈ {9, 11}`* (for the Tuesday labs only)*`,` and `# ∈ {1…25}.`*

Source Code

Source code from labs and lectures are available on Learn. TivaWare examples are also available from TI. You are welcome to reuse this or other code in your project on **two** conditions:

1. You explicitly acknowledge the provenance of any code you use, typically by way of comment in your source files.
2. You take responsibility for any code you use: this responsibility includes changing the comments to reflect your use of the code.

You will lose marks if you do not meet these conditions.

Serial Port

During development groups may choose to output debugging information over the serial port. This debugging information **should not** be included in the final version of the code. You are strongly encouraged to use compiler directives to control the transmission of such debug information. For example, a section of code which should not be compiled can be bracketed as follows:

```
#ifdef DEBUG_ONLY
// Code for debugging here; only compiled if DEBUG_ONLY is TRUE.
#endif.
```

Learning Support and Troubleshooting

Tutors and lecturers will be available during regular lab sessions throughout term 1 and 2. Additional support is available via the Learn forum, where students can ask (and answer) questions related to the course.