Project Description – SpaceShip

Objectives

The aim of the project is the implementation of an application in C that is executed on a microcontroller system. The second objective of this project is to analyze a medium-size programming problem. The third objective is to plan and manage the execution of such project with representative industry standards.

General Requirements

Req. 1.- The implementation shall be carried out using the ARM Cortex M4 CPU & STM32CubeIDE v1.8.0.

Req. 2.- The program C language code files shall be structured with clear and logical separation between functional libraries, and therefore with .h files. (Ref: Slide #13 of Course Lecture 1).

Req. 3.- The program abstraction (or architecture) shall contain at least three layers: Application Layer (project specific functionality), API (Application Interface Layer; functions that can be reused without modification in other projects) and HAL (Hardware Interface Layer). (Ref: Slides #8-9 of Lecture 2).

Req. 4.- The application shall use GPIOs, Timer, IRQ, a screen via Putty and the onboard LCD display.

Req. 5.- The Application Layer shall be documented using flowcharts. (Ref: Slide #10 of Course Lecture 2, and fx. https://www.draw.io/).

Req. 6.- The API (Application Interface Layer) and HAL (Hardware Interface Layer) shall be documented using a box containing name, variables and internal functions if any. A description for the module and resources used (eg. Hardware) if any. (Ref: Slide #11 of Course Lecture 2).

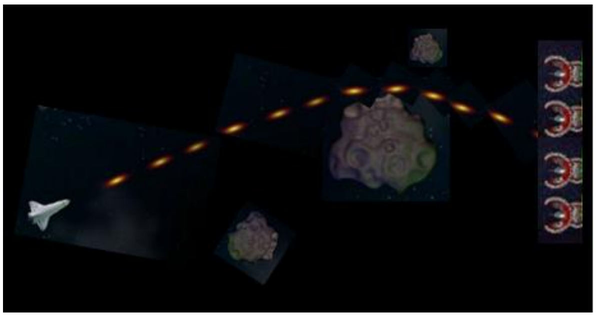
Req. 7.- Global variables may only be used if strictly necessary. If used, a strong argumentation shall be reported as well as a clarification with the reasons for not using local variables and passing arguments to functions. Also use of global variables requires additional commenting in the source code.

Req. 8.- All operations and numbers must use fixed-point representation / arithmetic

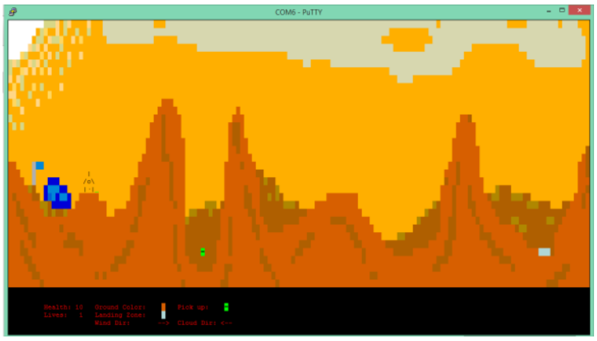
Project Proposal

The µP application to be implemented is a **SpaceShip** on either an interstellar travel or a planet rescue mission.

A.- SpaceShip interstellar travel mission: a space ship that in on its way to conquer another planet in the interstellar media travelling at a high speed is going through regions with asteroids, black holes and sometimes is attacked by alien ships. Our spaceship has been equipped with a lethal weapon can shoot with a different types of bullets and different angles. The bullets are subjected to a potential field and therefore trajectories are not straight. Some of the asteroids may contain resources, where spaceship can get supplies.



B.- SpaceShip planet rescue mission: a space ship is to be precise landing on the surface of a planet to a rescue mission. In the planet there are light and heavy asteroid storms often. The spaceship is equipped with weapons and shields for protection. The planet has small and big craters with regions of resources for resupply. Habitats have to be built before rescue humans that have been stranded in some remote and adverse areas of the planet. The potential field of the planet is not uniform and the trajectories of the spaceship may be heavily affected by it.



```
Health: 10   Ground Color:      Pick up:
Lives:   1   Landing Zone:
             Wind Dir:      --> Cloud Dir: <--
```

Our space ship defense capabilities are various: weapons and shield and other. There shall be three types of weapons and two types of shields. The shutting shall be controlled at more than 5 angles. Energy on the spaceship is limited and using the defense capabilities uses fuel. Fuel is also used for accelerating in the interstellar media and also for precise landing and taking off the surface of the planet. Refueling can take place at resource friendly locations.

Implementation and Extensions

There are many solutions to the problem and additional features that can be implemented. You are only limited by your imagination and the fact that you have two weeks to finalize the project.
You can get some inspiration at www.agame.com/game. The basic game shall look like one these arcade games. The resulting game shall also (**obligatorily**) include the following requirements:

- All of these: asteroids/planets with gravity effect, game levels with increasing speed controlled by the timer/irq, power-ups, number of lives on LCD/putty, number of points on LCD/putty, menu, help screen, and boss key.
- More than two of the following: multiple bullets, docking spaceships, animated figure instead of a bullet, score / high score, random delta-angle, two players, game controlled through PuTTY/Keyboard.
- More than one of the following: show info on RBG LED, game controlled by joystick, game controlled by potentiometer (ADC), full game on the LCD, sound from buzzer, use of onboard accelerometer.

- Other extras that may be included after completion of the above: using the RTC, using the PWM, using the DAC, using the flash memory, using an external joystick, using an ext. potentiometer, …..
At PDR we will agree on what you will implement for your project.

Phases

It is important to control the implementation of a project, and therefore industrial standards recommend a balanced split of the tasks of the project into 1) resources (members of your group) and 2) time (last two weeks of the 3 week period). In order to manage the project, we foresee two milestone reviews:

• **Preliminary Design Review (Mon-Tue, Jan 10th-11th):** Here you shall present how you plan to implement the project. The objective of this review is that you have analyzed the project scope and you present and converge into the concrete specifications requirements for the project you are going to implement. At this review it is required to present a **project block diagram**, **flow charts** for different levels and modules. A **project schedule and plan** shall also be presented, taking into account the tasks to be accomplished. (Ref: Slide #12-13 of Course Lecture 2). These will be part of your final report.

• **Status Design Review (Mon-Tue, Jan 17th-18th):** Here you shall present the project status and plans for its finalization. Final versions of block diagram, flow charts, schedule and plan are expected. The objective is to ensure that the project execution can be rounded-up including the documentation and report within the given time frame. You are required to present what has been completed & tested, what is undergoing and/or under testing, what is missing and the strategy for completion and testing of the ongoing tasks.

Documentation Requirements

Documentation is the most important outcome of any project. If the hardware and/or software implementation has been poorly documented it is very difficult to fix problems and/or add improvements later on. That is why you need to focus not only on the software but on the project as a whole including project management, hardware, software and documentation.

The report (you may write it in Danish or English) documents the project work, and it should include the relevant technical aspects of the project. Here follows a suggestion for the report structure, but you may assess whether you use this or not. Formally, it shall contain at least the following information (chapters):

• A front page containing
- DTU Space, Technical University of Denmark, 30010 Programming Project
- group number (you can check on DTU Inside or on the delivered hardware)
- for each group member: first name, last name, stud. nr., DTU Inside picture and **signature**.

• Both (!) an abstract (in English) **<u>and</u>** a “resume” (in Danish), if you wonder what an abstract is, look here: https://www.wikihow.com/Write-an-Abstract
• Introduction, which put the problem in perspective
• Specification requirements, for the project and with the reasons for choosing the given solution
• Project Structure: What modules (one subsection for each) have been developed and **who** created them
• Project verification: Strategy for project testing
• Project plan: What (code, exe, test, doc, etc) has been done, when it took place (timeline), and who did it
• Users manual for the application (that’s because we want to play with your game!)
• Conclusion: What went well, what could be done better, what could be included in the future
• References
• Appendixes: Developed source code, including .c and .h files, *journal* of the exercises with the relevant code

Delivery

The report shall be delivered latest on <u>Friday **Jan 21th**</u>. The HW shall be delivered in building 327 or 341.
Two files shall be uploaded on DTU Inside (under “Opgave” as group, one group member uploads and invites the others to the group, who have to confirm the delivery):
- 30010_GroupXX.doc or pdf: An electronic version of the report including all appendixes
- 30010_GroupXX.zip: compressed file containing the project folder incl. all source code & a <50Mb video clip in avi or mpeg4 format