



Assignment 5

Implementing a closed loop control application in Zephyr

Autores: Bruno Feitais 93384

Afonso Lavrador

Turma: P2

Data: 21/06/2022

Docente: Pedro Pedreiras

Introduction:

The aim of this assignment is to learn how to implement a simple closed loop control application in Zephyr, using the real-time model. The application mimics the typical structure of reactive real-time applications, comprising a sensor, decision logic and actuation.

Specification:

The system comprises a light sensor, an illumination system and a Human-Machine Interface. The system can operate in two modes:

- Automatic: programmable via the terminal and via the DevKit buttons. Allows setting On/Off periods and the corresponding light intensity;
- Manual: interface via the DevKit buttons. Allows to turn the system On/Off (when in "Off" the system operates in automatic mode), via two of the buttons. The other two buttons allow to set (increase/decrease) the desired intensity.

In terms of external hardware, the system comprises:

- Light sensor: phototransistor, Vishay TEPT5700;
- Illumination: a simple led.

Shared memory + Semaphores:

Thread A:

- Detects if the user choose Automatic or Manual mode.
- Periodic Thread.
- Gives semaphore AB.

Thread B:

- Lets the user increase the LED intensity.
- Takes semaphore AB, gives semaphore BC.

Thread C:

- It gets the average and implements it on the LED 1.
- Takes semaphore BC.

Thread D:

- It reads 10 ADC values and saves it on the shared memory.
- Takes semaphore AD, gives semaphore DE.

Thread E:

- It gets the 10 ADC values, does the average and saves it on the shared memory.
- Takes semaphore DE, gives semaphore EF.

Thread F:

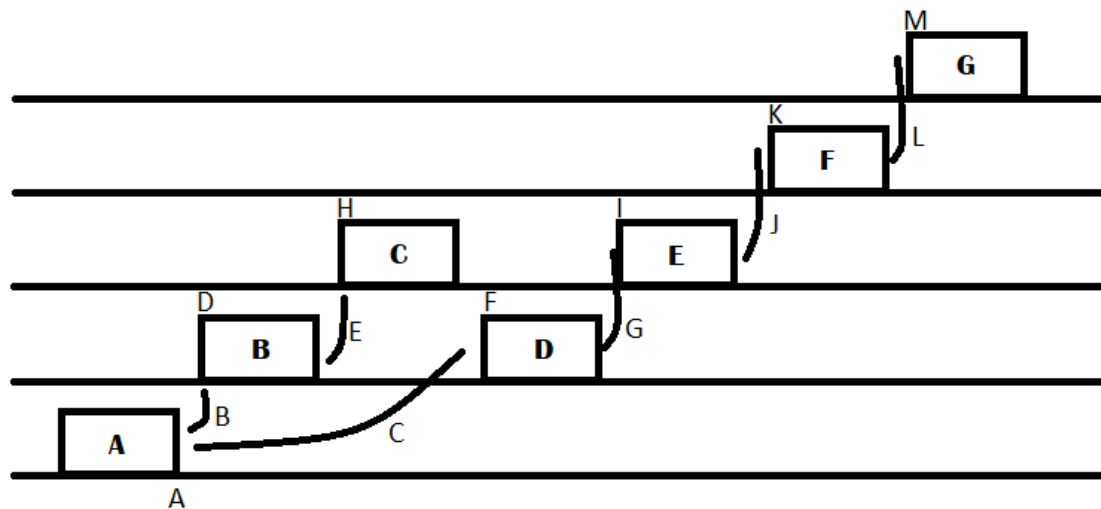
- Lets the user chose how much light intensity wants. After that compares the real light intensity with the wanted from the user and adjusts it by changing the light intensity of the led.

- Takes semaphore EF, gives semaphore FG.

Thread G:

- It gets DadosFG and implements it on the LED.

- Takes semaphore FG.



A – Period = 100ms

B – Give semaphore AB – Manual mode

C – Give semaphore AD – Automatic mode

D – Take semaphore AB

E – Give semaphore BC

F – Take semaphore AD

G – Give semaphore DE

H – Take semaphore BC

I – Take semaphore DE

J – Give semaphore EF

K – Take semaphore EF

L – Give semaphore FG

M – Take semaphore FG

GitHub:

On this link is possible to find the project repository containg the project code.

<https://github.com/BrunoFeitais/assignment5SETR.git>