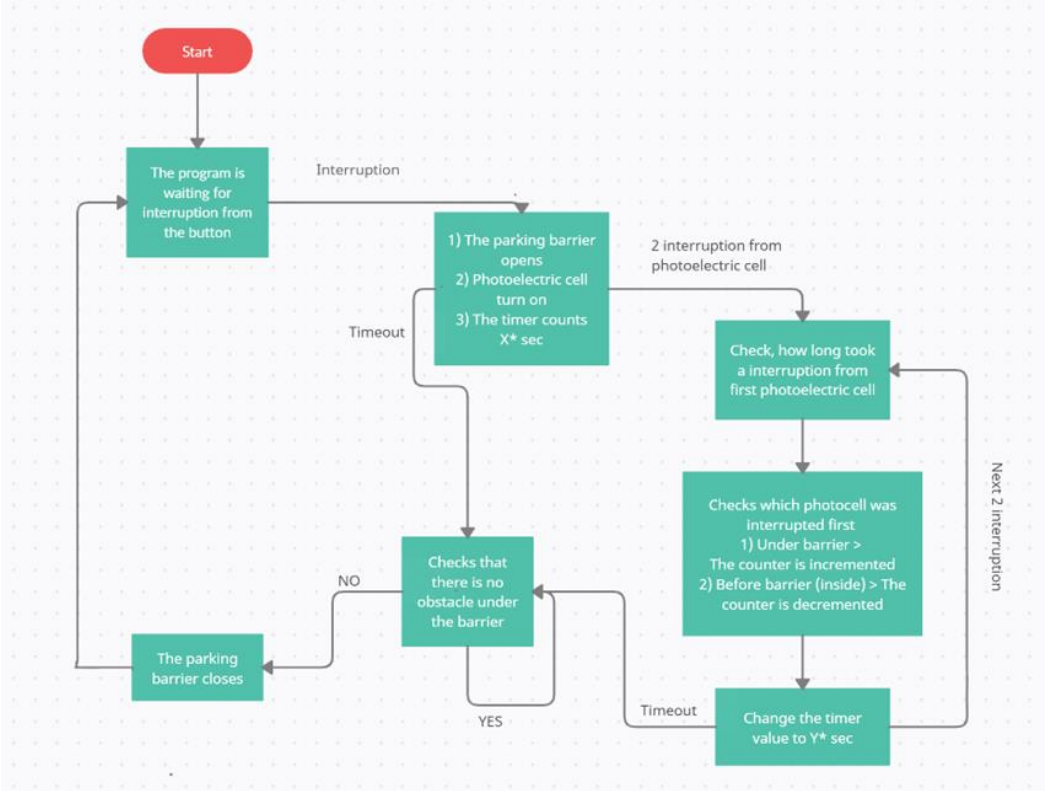
| AGH<br>Design Lab | **Maciej Zieliński**<br>**Bartosz Żołądź** |
|---|---|
| Course: Electronics and Telecommunications | Group thursday 9:40 |
| Date of submission:<br>19.01.2023 | |
| **Parking System Project** | |

## 1. Introduction

This is a project documentation prepared for a credit project in the Design Laboratory course at the Department of Electronics, Computer Science and Telecommunications at AGH University of Science and Technology. Created by Bartosz Żołądź and Maciej Zieliński from 3rd year of Electronics and Telecommunications 2022/23. All our progres and a project documentation can be found on our github page: https://github.com/baruselski/parking-system-project.

## 2. Short project description

The aim of the project is to create a simple parking barrier control system, which can be applied to e.g. housing estates, public or private car parkings etc. The main idea is having only one enterance to a parking with a specific number of parking spots. The parking barrier placed at the enterance would double as an enterance and an exit. Opening the barrier would be triggered by an external button (e.g. a security guard controlling the system). By using 2 photocells, it is possible to count vehicles entering and leaving the car park, as shown in the diagram below:



## 3. Project goals and assumptions

The main practical goals of the project are:

- the opening/closing of the barrier – done by using a servo control mechanism
- displaying of information on the current status of vacant seats – accomplished with two photocells placed in parallel at a certain distance and a 7 segment display. The function analyses which photocell interrupted first. In this way, we are able to distinguish between a vehicle entering or leaving, and then set the value on the 7-segment display visible to entering vehicles accordingly
- ensuring safety against the possibility of the barrier hitting an obstacle - executed by one of the photocells from the previous point, which also acts as a safety device
- preventing the free spots counter from increasing in case of a pedestrian passing under an open barrier (if object passing under the barrier takes less than 1 second – counted as a pedestrian) and two photocells - from the numerical assumptions made below about the speed limit in a given area, it is possible to calculate the theoretical minimum time for which a photocell is interrupted for an assumed average vehicle length. The result is the time that can be compared to the photocell interruption time
- the ability to manually change/correct the current number of free places on the display by the administrator - function realised by external interruptions from 3 buttons:

S1 - functional button to activate the parking barrier

S2 - functional button to activate changing of parking spots count with S3 and S4

S3 - increment

S4 - decrement

- a physical set-up of the system model so that it can be tested

**Design assumptions**

Free parking spots: 99,
Barrier opening time(X): 10 seconds,
Hold time of the barrier after interruption of the photocells(Y): 2 seconds,
Speed limit applicable in front of the barrier and throughout the site: 10km/h (2,77m/s).

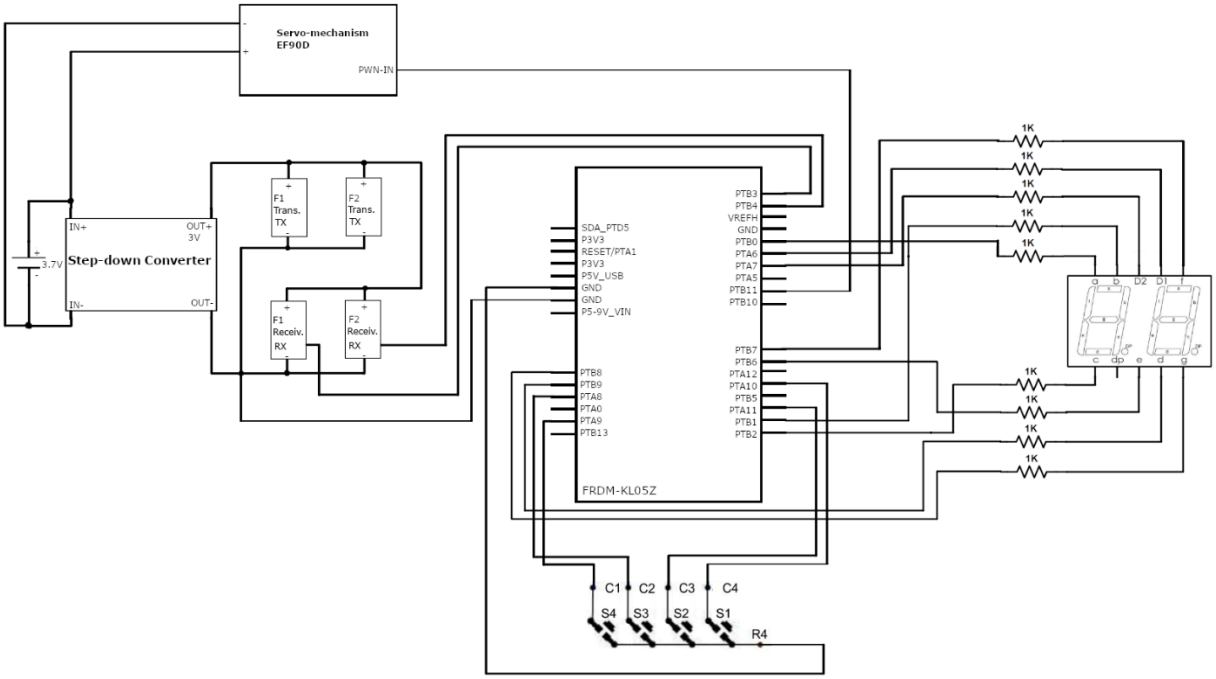
## 4. Elements specification

Here is a list of elements used for the project:

- microcontroller FRDM-KL05Z
- 2x photocell
- digital servo EF90D
- 7-segment display (2 digits)
- 4 buttons
- Li-Pol battery 3.7V, 250mAh
- Step Down Converter LM2596

## 5. Circuit schematic

Here is a schematic for all the connections necessary for the parking system to work correctly:




## 6. Program code

Here is a list of .c and .h files used in the project:

- main.c – main function (used for calling opening/closing/display functions, changing counter value and checking most of the safety measures), also contains SysTick_Handler function used for handling SysTick interrupts,
- display.c, display.h – used for programming the 7-segment display (initialization, display function),

- klaw.c, klaw.h – used for programming S1-S4 buttons (initialization, interrupt initialization, interrupt handling),
- serwo.c, serwo.h – used for programming the digital servo mechanism (initialization using TPM0, opening, closing and stopping functions),
- photocell.c, photocell.h – used for programming all photocells used in the project (initialization, interrupt initialization, interrupt handling),

For more information about the above functions, comments were used directly in the program code.

## 7. Physical build

A 3D printer was used to create a physical case for the parking system. The case was attached to a plastic plate for easier demonstration.