



Faculty: Computer Science, Electronics and Telecommunications Faculty
Electronics and Telecommunications

Year of Study: III

Microprocessors Technology 2

Topic: Led Cube with animations and games

Date: 24.01.2021

Group no.: Friday 11:15

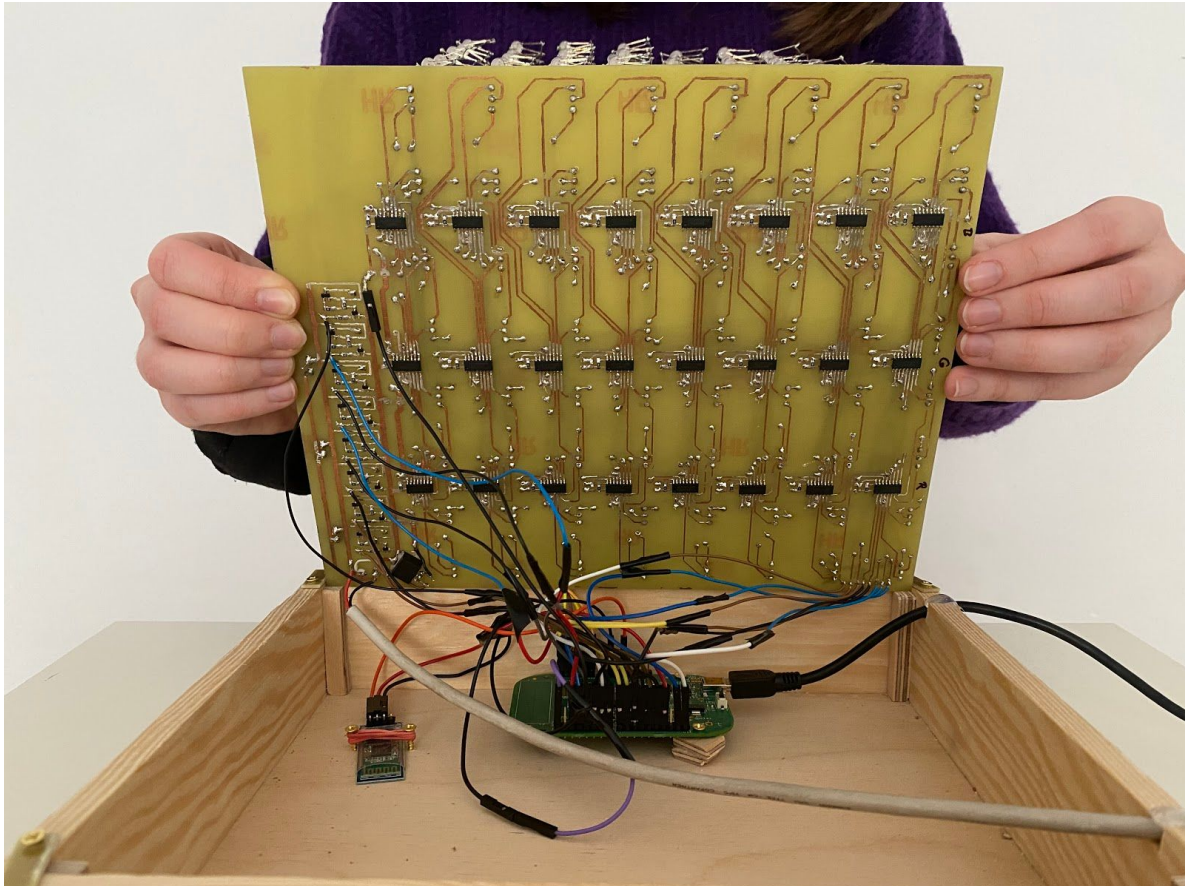
Students names and index nr: Godula Kacper - 305255

Macura Aleksandra - 305258

1. Introduction

As our project we prepared RGB Led cube 8x8x8. Our main goal was to create some interesting games and nice looking animations. In the end our project consists of 2 games - snake and tetris, which can be controlled either by computer keyboard via PuTTY or in the app on the phone using Bluetooth. In this report we are going to briefly describe the hardware part of our project as well as the individual files of the code part.

2. Hardware part



- the cube

It consists of 512 RGB diodes with common anode in layer. The cube is standing on a printed circuit board, made at home using thermal transfer overprinting method - printing the desired circuit from KiCad, ironing it to the plate and then etching the rest of the copper using Iron(III) chloride. On the board there are soldered 24 SCT2167 integrated circuits because of which we are able to send data to our LEDs using the limited amount of pins from the processor. Beside there are also power amplifiers for each layer of the cube consisting of one NPN transistor, one MOSFET and 3 resistors.

- Freescale Kinetis KL05Z

As the base of our project we used the same board as during laboratory classes. This way we already had experience in using the useful things like PWM, SysTick, UART on exactly the same hardware.

- Bluetooth module

We used HC-06. Because of this device we are able to connect to the app on the phone.

- Keyboard

As we didn't have enough pins to connect it whole(4x4) we used only a few buttons, but it is enough for our needs because in this version of our project we use it only as a device to switch between the games.

3. Software part



We wrote our whole project in C language using the Keil uVision5 IDE.
Our code consists of the following files:

- main.c

In this file you can find the most important part, without which the program wouldn't work - eg. Enable clock, interrupts as well as the base of the game - choosing which game we want to play after leaving the demo part.

- tpm.c

It is responsible for controlling the brightness of each of the 3 colors. We are using TPM0 (channels 0, 1 and 3) with modulo value set to 65 in order to get period ~0,2ms - 10 times smaller than SysTick.

- sct.c

In this file we have only one but very important function - Send8BytesT. It is responsible for sending data to the integrated circuits. In every iteration of the for loop in this function we send 1 byte for each color which is later moved further.

- systick.c

Here we are dealing with SysTick_Handler. We are checking whether buttons on the physical keyboard were pressed and also changing currently active layer. We also use the Sent8BytesT function to send data. The last part is handling pressing the buttons on the keyboard.

- uart0.c

In this file we are handling the communication via UART because of which our games can be controlled using Bluetooth or PuTTY on the PC. In the UART0_IRQHandler we enable everything that we need and in UART0_IRQHandler we manage the symbols assigned to all of the buttons that we can press either in the app on the phone or on the keyboard of the computer.

- tetris.c

In the beginning we have declaration of the blocks. We can declare blocks of maximum size of 3x3. In that matrix we declare which diodes we want to turn on by assigning "1" to them. This way we create a number which "describes" the block. Later we say what are the dimensions of the block - without it there were problems with moving the blocks to the edge of the cube. In the next parts of the file we have all of the functions thanks to which playing tetris is possible, for example Draw_Block, Move_Block or uint8_t Tetris(void) which is the function for the main operation of the game. The whole game works as regular tetris - from the top of the cube blocks are falling randomly until they stop on the bottom level or on another block. When the whole layer is full then it disappears and when the block stops on the top level the game ends.

- snake.c

In this part of code we prepared the second game - snake. The main idea is the same as in the original game - the more apples the snake eats, the longer it becomes. The game ends when the snake hits the wall of the cube or itself. Then the result is shown on the first wall of the cube. There are functions responsible for all of the things that are needed for the snake - appearing of the snake, appearing of apple, eating an apple, moving the snake and ending the game.

- demo.c

In this file there are prepared functions with animations. Some of them are really simple, like lighting up all of the diodes of one color or some specific shape, but there is also Demo_Contour which lights up the contour of the cube and then rotates it.

All of the .c files have its header files. This way the code is more readable and easy to change.

Appendix 1: schematics_led_cube.pdf

Appendix 2: Led_cube_bottom.pdf