JIEE Časopis priemyselnej elektrotechniky
Journal of Industrial Electrical Engineering
ISSN 2454-0900
www.jiee.eu

Intelligent traffic lights management system design based on ARM processor

[1]*Martin JAKAB*, [2]*Patrik JACKO*

[1] Dept. of Theoretical and industry Electrical Engineering, FEI TU of Košice, Slovak Republic

[1]martin.jakab.2@student.tuke.sk, [2]patrik.jacko.2@tuke.sk

***Abstract* — The article is focused to the intelligent traffic light system. Nowadays, with a lot of cars on the road, high efficiency traffic light system is required. It is caused, because, from some road way there are many cars and from the other road way, less cars can be. In this situation intelligent traffic system is required. The paper describes hardware and software system realization with few modes such as automatic, manual or night modes.**

***Keywords* — intersections, PCB board, STM32F7, traffic light**

I. Introduction

In this article we will describe the individual procedures as we have worked on the model of Crossroads itself. The procedure is divided into Hardware frequent and software section. The Hardware part consists of the formation of a PCB as well as the intersection model itself. The software section consists of creating software for intersection solutions. On the principle of microcontroller STM32F746ZG.

II. Block scheme Design

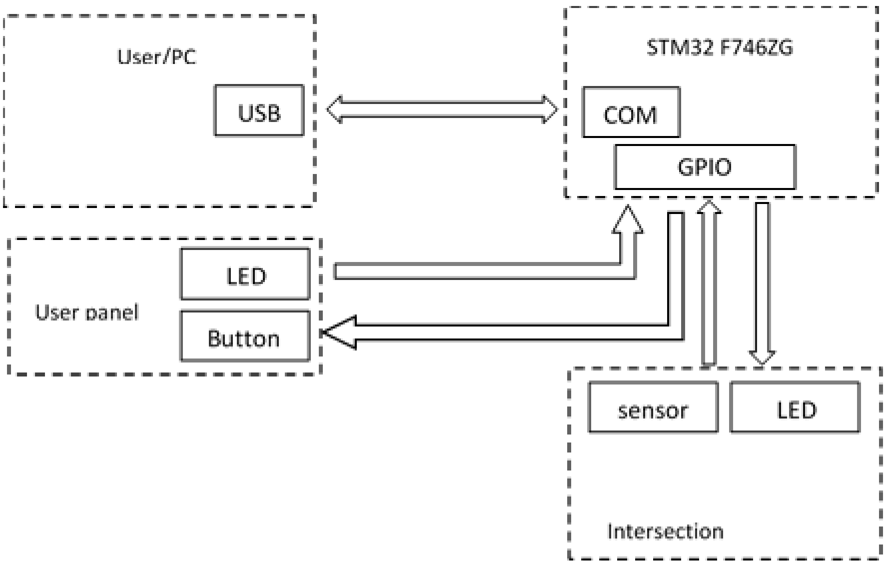

Fig. 1 Block diagram

The Fig. 1 shows a block diagram of a crossover solution based on the STM32F746ZG microcontroller principle. As we can see, the user communicates with the microcontroller via a USB port, which also uses the program on the board. Of course, the microcontroller also communicates with the environment using the COM port. It also communicates with junction using ADC converters and GPIO pins. ADC converters serve to communicate with inductive sensors. We can use the STMStudio program to read the sensor analog value. GPIO pins serve for communication with LEDs and T1 to T4 keys. Buttons T1 to T4 are located on the control panel.

III. THE INTERSECTION SCHEME

The following figure shows the interlocking scheme using the STM32F746ZG microcontroller.

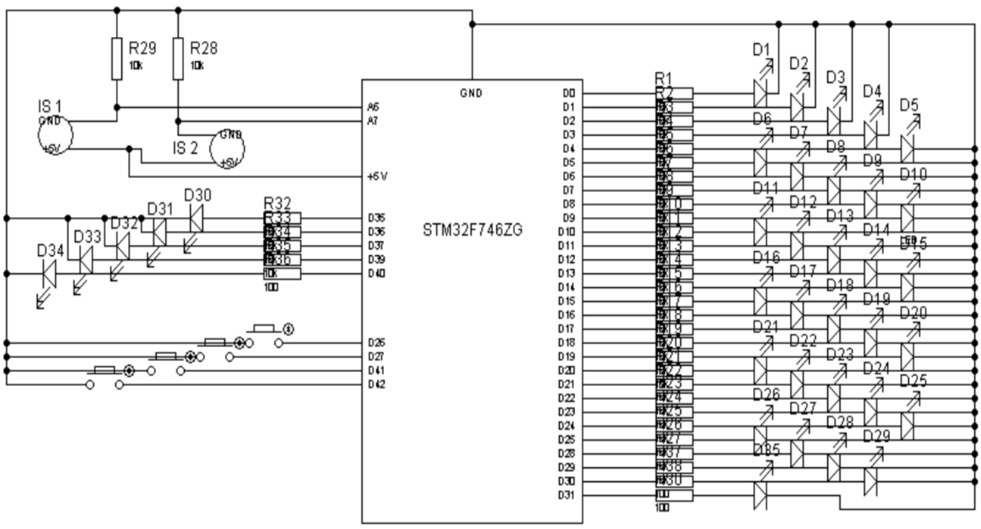

Fig. 2 Intersection scheme connection

The scheme is created in the program Proteus 8 Professional. When creating the scheme, we created our own component STM32F746ZG. Here we have created a rectangle in the case of the STM32F746ZG component, in case of induction sensor we have created a circle. In the next step, we assigned the necessary pins to the service. We did this by clicking the Device Pins Mode icon, located in the left vertical panel. There we chose Default. After adding all the pins needed in our involvement, we've tagged the entire object, right-clicking on the selected object, and then selecting Make Device. We then filled in the necessary data and the name of the component. The induction sensor in our scheme has a label is. In the design of the wiring diagram, we have not taken into account the number of LEDs due to the transparency of the wiring diagram.

Buttons T1 to T4 are the keys we use to control the intersection. T1 and T2 are the buttons to select the intersection mode, either night mode or main mode. The main mode is one that does not depend on any external influences, interruptions.

IV. HARDWARE PART OF INTERSECTION

As mentioned above, the hardware part of the intersection consists of circuit boards and the crossroads model itself. The printed circuit board includes microcontroller pinhole resistors, Nucleo board connectors, and a 37-pin cannon connector that serves as a crossover model with a microcontroller. In addition, pins are connected to the board to connect external buttons and two LEDs. Also, the supply + 5V supply leads are needed for inductive sensors. The intersection model consists of rigid polystyrene on which the crossed-out model is printed and the plexiglass is placed on the top of the model. In the plexiglass, the holes for the traffic lights are cut out.

A. Printed Circuit Board

Programs such as Orcad, Pads, Eagle began to be used at the end of the 20th century in the 1990s. Nowadays, these software programs are freely downloadable online on the Internet. The most commonly used programs for PCB design are Eagle, DipTrace and Orcad. Of course, there are many other software for the design of printed circuits. The Eagle program is popular not only in Slovakia, but also in Germany, America and Asia. It is created by CadSoft Computer based in Germany. The program consists of 3 main parts namely: drawing scheme, autorouter, and design of circuit board. The benefits of this program are: simple control, program stability, software cost. The other mentioned program is

DipTrace. It was created by Novarm in Ukraine. The master library contains over 100,000 components.
The program in which we designed the circuit board is Eagle with version 6.3.0, which provides designs of large-area boards. Crack for this version is available online on the Internet.

B. Eagle 6.3.0

First of all, we have created a new design for the board design. After creating a new project, we selected the necessary parts to create a printed circuit board. This is done by clicking on the ADD icon, located in the left vertical panel. After jumping the window, we select the required item from the menu. After removing the necessary suits, connect the individual joints as needed. This is done using the Signal icon, located at the bottom of the left panel. After connecting all the necessary paths, we can go about creating conductive paths using the Auto icon. After clicking, we will see the Autorouter setup folder where we set the TOP item to N / A and confirm with the OK button.

Once conductive paths have been created, you need to check if everyone has a blue color link, otherwise you need to complete the route manually with the route icon. The resulting design of the printed circuit board is shown in Fig.3. In addition to the design of the printed circuit board. The Fig. 3 also shows the user interface of EAGLE 6.3.0. The above-mentioned operations (ADD, route, Signal) needed to create the design of the printed circuit board are located in the panels located in the left part of Fig.3.

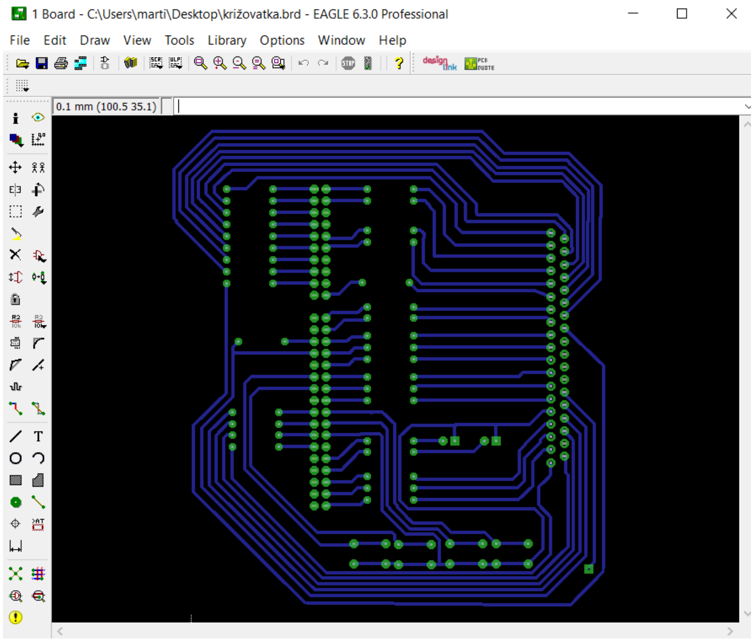

Fig. 3 PCB of the intersection control panel

C. Create 3D model

We used the online AUTODESK Tinkercad software to create 3D models. We created the models using the simple shapes that were provided to us. After creating the model, we saved the project under the .stl extension. Figure 4 shows the traffic light along with the stand.

The traffic light on the left is a traffic light for cars turning left. Next to it, there is a traffic light for cars going straight ahead across the junction. Two traffic lights consisting of two LEDs are pedestrian traffic lights. The last traffic light is designed for cars turning to the right on the side / main road.

Fig. 4 Traffic light

Part of the semaphores is manually created with the aid of a handwriting with plastic ledges intended for construction purposes. Others are printed using the EASY3DMAKER 3D printer. For 3D tiles, you need to have the correct print settings. One of the main features is the thickness and accuracy of the fill inside the model. The smaller the amount of 3D model filling, the faster and sometimes better the print, depends on the complexity of the model.

V. Control panel

The junction controlled by the microcontroller has its own control Panel on its surface. The Control Panel contains 4 user keys T1 to T4, the two LEDs for manual mode alarm. The T1 and T2 buttons are used to change the mode namely: T1 is user button for change mode to main mode, T2 button for change serves as a button served for night mode. The T3 and T4 buttons are used to change the night mode to manual mode. These buttons operate on the counter principle when a number of keystrokes are reached, the junction mode changes to a decent state.

Fig. 5 Control Panel

VI. Modes of intersections

The intersection works in two main modes. The first is the main mode. This starts by pressing the T1 button. When pressed, all traffic lights are set to red. After a few seconds with lots of mode. In each time Interval, all directions, including pedestrians, will be rotated. The T2 button is used for night mode, in other words the orange blink mode. Night mode is special by inclusion and manual mode. Manual mode is meant to use external inputs, whether it is the positioning inductive sensor or the T3 and T4 buttons. Using these component will change night mode to manual mode. Induction sensors working on the principle of metal sensing to a distance of about 6mm. The closer the sensor is given by the metal, the lower the sensor value. For the proper operation of the induction sensor we have used 10 KΩ resistor.

The T3 and T4 buttons work on the principle of external interruption. Pressing the microcontroller button registers the rising edge of the button to change the junction mode using the conditions.

VII. Software part of intersections

The software section includes a program that we used to solve a junction controlled by the STM32F746ZG microcontroller. As mentioned in the previous article, we used the inkscape program to design the mock-up. We used CubeMX to define pins and pin settings. In the Keil uVision5 program, we have solved the junction code using HAL libraries.

VIII. Intersection Modes

As we have already mentioned, crossing is a two-way control, namely: main mode of control or night mode. In this section, we'll get closer to the cross-over mode.

A. Main function

The main function of the program is assembled from the switch when the appropriate junction mode is set according to the variable mode. If the value of the variable mode is 1, the junction is set to master or automatic mode respectively. If the value of the variable is 2, the junction is set to night mode.

B. Automatic mode

Automatic mode of control is the mode when the T1 key is pressed. The principle of the mode is that the individual semaphore signaling states are followed consecutively until the mode change signal is pressed by pressing the T2 button. The initial situation in the main mode is to set all traffic lights to red. After a few seconds, the intersection changes to a state where green has Q1 and Q3. After certain time intervals, signaling of traffic lights is changing for all directions of the intersection, including pedestrian and tramway traffic lights.

C. Night mod

The night mode of the intersection is basically the orange flash mode. The mode is started by pressing the T2 button. When you press the T2 key, the main function tests whether the mode is running for the first time or the main intersection mode has been started. In case the main mode was started, before the night mode, the transition_resize function is performed, which in the real intersection serves for operational safety.

The night mode function works by initially testing the condition when the variable mode is equal to the night mode. Otherwise, the main function is performed. If the variable is equal to 2, the function is started by initializing the external interrupt functions of the buttons or sensors. Furthermore, there are conditions for individual external inputs, but this is the T3 and T4 button or the inductive sensors. Subsequently, for testing the conditions, the switch is followed to evaluate the conditions and their status.

D. Night mod – button T3 and T4

In night mode, the T3 and T4 buttons are specific. With these buttons, we can change the manual night mode to the mode selected by us. The T3 and T4 buttons work on the principle principle, only with different functions. This choice was based on the idea that if a traffic jam would come during the night mode at a intersection, for example, a pedestrian or a car, the signaling to green would change for traffic because of the traffic accident, because it was not a tragic accident one night. To do this, press the T3 button. Press the T3 key to increment the variable number. After performing the sensor_T3 function, it is tested whether the number is 3. This means that the button has been pressed at least 3 times. If it has been pressed just 3 times, the appropriate function is executed, if not, the program continues and the function is terminated or jumped out of the function by the break command.

E. Night mod – snimac1

This function responds to value_IS1 and value_IS2. At the moment the sensor responds to a change of value, at that moment, from the function night we get into the snimac1 or snimac2 function, depending on which sensor has registered the change of value. The *snimac1* (*snimac2*) function is a cycle that tests every second of the variable time. The cycle is set for 5 seconds. After each cycle, it is tested whether the time is equal to 5. It is assumed that during the entire cycle, the sensor must perform a change of value, so it must be active. If for 5 inactive at least once, the variable time is not magnified, and the function is not performed. Then it continues night time.

IX. CONCLUSION

This work is focused on the use of a microcontroller in industrial electrotechnics. More precisely this is the design of a junction system using a microcontroller. In our case, we used the STM32F746ZG microcontroller. We used semiconductor components in the form of LEDs. We used them to signal traffic lights. To change modes, we used the panel buttons labeled on the control panel as the T1, T2, T3, and T4 buttons. Semaphores as we yourself were created using 3D printers. We were first designed in a format that supports 3D printing. Stands are made of aluminum tails, which are slowly bending the bend to the desired angle. Other materials that are used in the work are plexiglass, woodboard and styrodur. In the software section, we have solved the periphery by using external interrupts and ADC converters. External interruptions have been used for all buttons located on the control panel. These, as mentioned above, have served to change the cross-border modes.

REFERENCES

[1] Cortex-M7[online][cit.16.12.2017] – available to the internet <https://developer.arm.com/products/processors/cortex-m/cortex-m7>

[2] STM32f746ZG-datasheet [online] [cit. 16.12.2017] – available on the internet - <http://www.st.com/en/microcontrollers/stm32f746zg.html>

[3] Nucleo 144 [online] [cit. 16.12.2017] – available on the internet - <http://www.st.com/content/ccc/resource/technical/document/user_manual/group0/26/49/90/2e/33/0d/4a/da/DM00244518/files/DM00244518.pdf/jcr:content/translations/en.DM00244518.pdf>