Design Laboratory

Electronics and Telecommunication

Remote power strip



Realisation:

*Damian Grzesło*

*Przemysław Kapała*

*Kraków: 17.01.2022*

# Table of contents:

[1. Table of contents: 2](#_Toc93318259)

[2. Specification: 3](#_Toc93318260)

[3. Work schedule: 3](#_Toc93318261)

[4. Used components: 4](#_Toc93318262)

[5. Block diagram: 4](#_Toc93318263)

[6. NEC encoding 4](#_Toc93318264)

[7. Implementation: 5](#_Toc93318265)

[8. Software: 6](#_Toc93318266)

[8.1 Variables initialization 6](#_Toc93318267)

[8.2 UART setting: 6](#_Toc93318268)

[8.3 Timer Initialization 7](#_Toc93318269)

[8.4 Relays 7](#_Toc93318270)

[8.4.1 Device Switching 7](#_Toc93318271)

[8.4.2 Relays initialization 7](#_Toc93318272)

[8.5 Times measurement interrupt 8](#_Toc93318273)

[8.6 The main function 8](#_Toc93318274)

[8.6.1 While loop 8](#_Toc93318275)

[9. Test and validation: 10](#_Toc93318276)

[9.1 Diode test 10](#_Toc93318277)

[9.2 Oscilloscope testing with LEDs 11](#_Toc93318278)

[10. Final version after testing and real life application: 13](#_Toc93318279)

# Specification:

The main goal of the project was to create a remotely controlled power strip, in which each individual electric socket can be switched on/off remotely by the end user. The switching of the devices will be done remotely using IR pilot and IR receiver.

Microcontroller will manage the whole system through the relays, according to the signal from the receiver. Any device powered by 230V AC may be connected to one of the relays.

Arduino module with ATmega328p, IR receiver and relays are powered from 5V DC supply, and pilot is powered by the battery.

Final version of the project is assembled on the one board. In order to make plugging devices easier, 4 outlets are connected to the relays. One pin from each socket is linked up directly to one wire from 230V AC power supply and the second pin is connected through the relays.

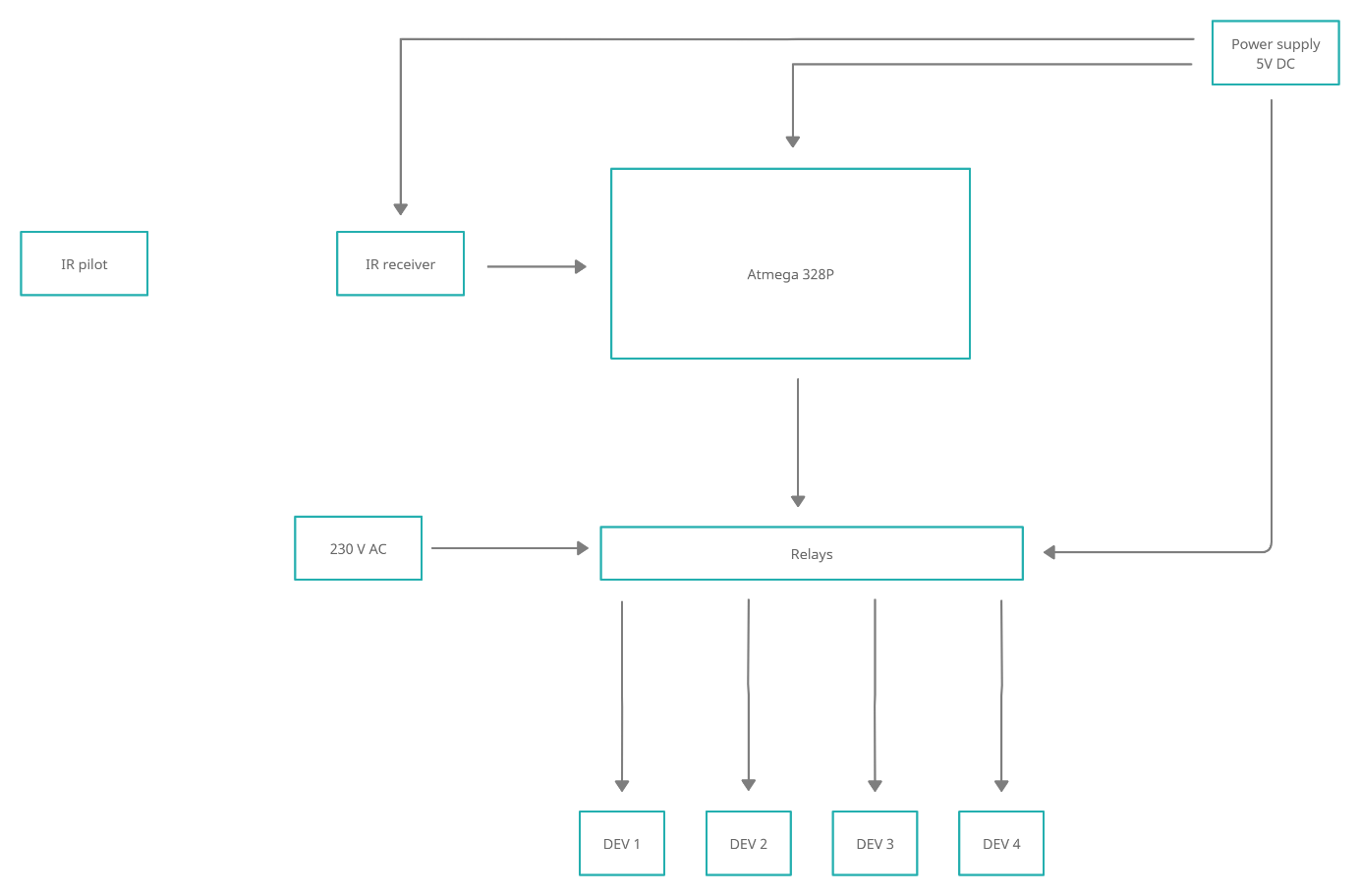
# 3. Work schedule:

|  |  |  |
| --- | --- | --- |
| Time | Tasks | Responsible |
| 17.11.2021 – 24.11.2021 | Project requirements and schedule:   * Looking for datasheets of elements * Working on schedule * Making block diagram | Przemysław Kapała,  Damian Grzesło |
| 24.11.2021 – 8.12.2021 | * Studying IR encoding standard of the pilot. * Analysis and creation of the algorithm to process the signal from the receiver. | Damian Grzesło |
| 8.12.2021 – 22.12.2021 | * Implementation of receiver and decoding algorithm to create IR decoder. * Programming of relay control system for device switching. | Przemysław Kapała |
| 22.12.2021 - 5.01.2022 | * Prototype creation * Verifying of decoding algorithm correctness. * Connecting various devices to relays. | Damian Grzesło |
| 5.01.2022 – 12.01.2022 | * Final testing * Project assumptions verification * Project validation | Przemysław Kapała |

# 4. Used components:

|  |  |  |
| --- | --- | --- |
| Component: | Description: | Purpose: |
| ATmega328p | Microcontroller on the  development board Arduino UNO R3 | Control and management of the whole device |
| IR receiver TSOP31236 | Receiver of infrared rays | “catching” the signal from the remote |
| Car MP3 remote | Arduino IR remote control | Sending signals to control the device |
| 4 channel 5 V relay (SRD-05VDC-SL-C) | A relay module with 4 entries. | Power supply for sockets. |
| 4 electric sockets | A socket for device | Plugging things into it. |

# 5. Block diagram:



# 6. NEC encoding

In our IR pilot , NEC coding works as fallows:

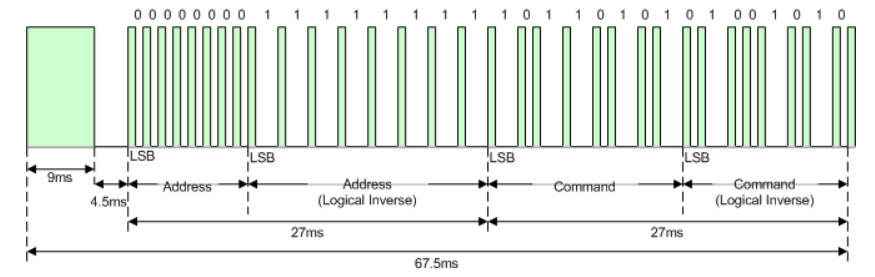
Carrier frequency is on 38 kHz.

Logic states of “0” and ”1” are coded like this:

* “0” – high logic state of signal spans for 562.5µs and then for 562.5µs there is low logic state. Total time is 1.125 ms.
* “1” – high logic state of signal spans for 562.5µs and then for 1.6875ms there is low logic state. Total time is 2.25 ms.

Coding of “0” and “1” differs only in how long is the space between high states of coded signals. For “1” this space is 3 times longer than for „0”.

Example of NEC transmission:



At the beginning of the code, there is an identificatory which signalizes the beginning of new transmission. It consist of 9 ms burst and then 4.5 ms pause. In the next 27 ms remote’s address is send. At first normally, then in inverse to check it’s correctness. Following the address there is an appropriate command, also send in two sections, first normally, second in logical inverse.

Every command is unique set of “0”s and “1”s for every button.

Transmission is ended by 562.5 us burst.

In the case button is held for a longer time, pilot sends periodical code different from the initial every 108 ms. This code consists of:

* 9ms Burst, the 2.25ms space and 562.5µs Burst at the end. This end Burst always means end of transmission.

As seen, code of the repeated signal doesn’t bring in any new information.

# 7. Implementation:

The device was built on Arduino UNO development board. The reading of the signal sent from IR remote is done by IR receiver, signal form the pilot is coded in the NEC standard.

The information is sent to the board where it’s decoded. Based on the decoding result a specified relay is switched on/off.

Relay used in the project is one module with 4 channels so it can be easily linked up to 4 electric sockets. It is possible to switch one of them or all of them using buttons from remote.

**NEC decoding algorithm implementation.**

Firstly we must detect whether the message is initial or repeated.

After initial Burst repeated signal has pause shorter by half than in the initial signal.

It’s duration is 2.25ms. Measuring this will allow to differentiate between them.

Time measurement in between successive high states will be the most important thing to do because it’s the only difference between “1” and “0”. High state in signal always takes the same amount of time: 562.5 us.

Proposed method is to measure time between signals rising-edges slopes and checking to which time interval it belongs to:

„1” – time measurement 2.25 ms ± error margin.

„0” – time measurement 1.125 ms ± error margin.

Signal with information – time in between first rising-edge slopes 5.0625 ms ± error margin.

Repeated signal – time in between first rising-edge slopes 2.8125 ms ± error margin.

Error margin is going to be experimentally appointed in the implementation phase.

It will still be possible to clearly read what was sent, also allowing for some error margin in case of hardware inaccuracy. Every command and address will have it’s correctness verified by comparing it with it’s inverted version.

Then using interrupts from IR receiver the measurement will start. Depending on whether it’s initial or repeated transmission it will be decided whether to continue handling interruption or not.

Remember: IR receiver works in negative logic.

If IR receiver receives high logical state, then microcontroller will get low logical state from it and for pause it will be high logical state.

# 8. Software:

## 8.1 Variables initialization

Obraz zawierający tekst

Opis wygenerowany automatycznie

In those lines there are six variables and two arrays initialized. All of them except for rx\_buf are declared as volatile so that compilator does not optimize them during compilation.

## 8.2 UART setting:

It isn’t necessary for the project to work correctly, but if the user wants to lookup what’s happening inside the program, ex. check the NEC code for other buttons or if buttons are decoded correctly.

Obraz zawierający tekst

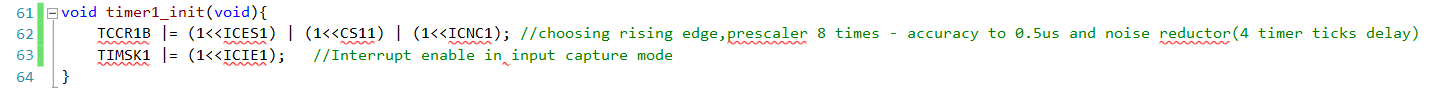
Opis wygenerowany automatycznie

Function in lines [41-45] is used by the uart\_puts function.

Function in lines [47-52] is for used for printing words to uart.

Function in lines [54-59] is used to initialize the uart.

## 8.3 Timer Initialization



This function initializes timer important for NEC decoding, as information is coded by impulses of different duration.

It must be set on the rising edge, and have high enough accuracy for the program to differ different impulses.

## 8.4 Relays

### 8.4.1 Device Switching

Obraz zawierający tekst

Opis wygenerowany automatycznie

Depending on the command send by the IR pilot, different device will be switched ON/OFF.

There are 2 cases in which allow us to switch ON/OFF all devices simultaneously.

### 8.4.2 Relays initialization

Obraz zawierający tekst, zegar

Opis wygenerowany automatycznie

They are initialized on PORTD, on the four pins. Each pin is connected to different relay and controls it.

## 8.5 Times measurement interrupt

Obraz zawierający tekst

Opis wygenerowany automatycznie

When the IR receiver, get’s the first rising edge, value of time is saved in ICR1 register by the microcontroller.

Time is measured between two rising edges, then saved to the array of times “pomiary”, in the index indicated by “licznik” variable.

Interrupt finishes it’s job after all 34 times are measured and saved.

On the lines [102-106], if the second time in “pomiary” in not within set boundaries, then the program knows that the button was held for extended period of time, and signal was send again.

In that case, “licznik” is cleared, second index in “pomiary” is cleared, and “przytrzymanie” flag is set to inform the rest of the program about this event.

## 8.6 The main function

Obraz zawierający tekst

Opis wygenerowany automatycznie

Functions are activated/initiated. Global interruptions are enabled

### 8.6.1 While loop

Obraz zawierający tekst

Opis wygenerowany automatycznie

Obraz zawierający tekst

Opis wygenerowany automatycznie

On the line [123], if statement checks, whether the received signal is an incomplete message.

If condition is true, then there is no change on relays, “przytrzymanie” flag is cleared.

On the line [128] if statement checks, whether the received signal is an complete message.

If condition is true, then there is 200 ms delays to ensure programs correct work (Written on the

screen one).

On lines [135-162] there is NEC decoding algorithm implementation.

On lines [163], it’s checked if command was correctly received. There could have been receive error,

or command was incorrectly decoded.

If condition is true, then react() function is activated and device or devices change it state/states.

Command can also be written on the console through uart, if the user wants it.

If condition is false, the receive error flag is cleared and the error message can be written on the

console through uart. User can check if something was wrong and for example he can press the

reset button.

At the end lines [174], “flaga” is cleared, command was decoded, program waits for the next

complete signal.

# 9. Test and validation:

After projects assembly and microcontroller programming, tests were carried to determine correct operation of the elements.

## 9.1 Diode test

This test’s goal was to determine whether signal decoding is correct and there is no hardware problem. It was conducted on the breadboard with the use of LED’s as in the photo nr 1.

Then different buttons were pressed.

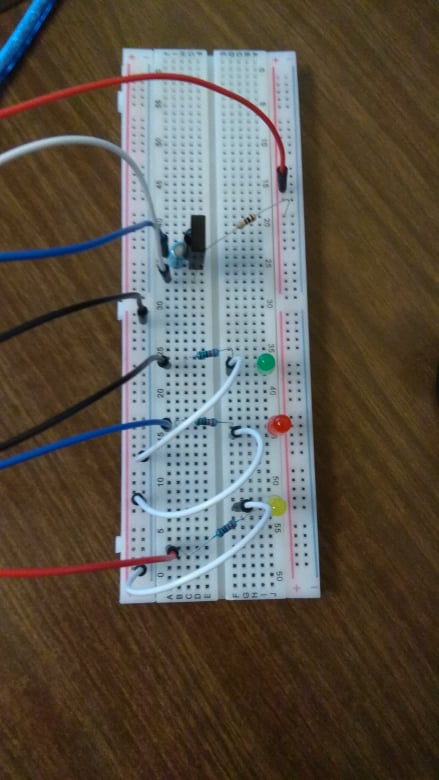


Photo nr. 1 – connection

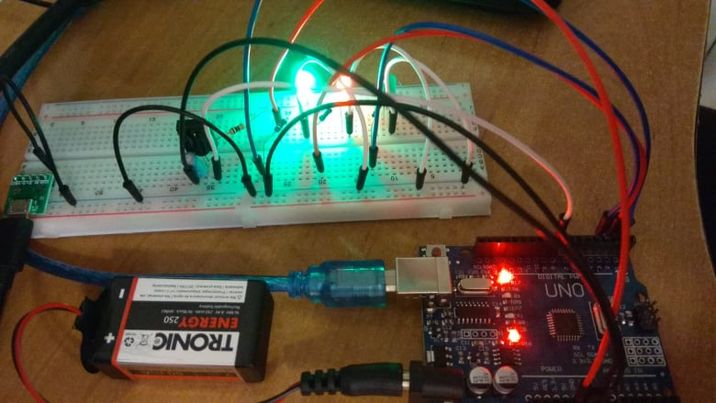


Photo nr. 2 – photo of the project during testing

As seen in the photo nr. 2, there were no hardware problems.

LEDs were switching for corresponding buttons.

## 9.2 Oscilloscope testing with LEDs

To make sure that IR receiver works fine another test was conducted. An oscilloscope was connected to receiver. As seen in the photos nr. 3, 4, 5 receiver works correctly, and each diode is switched by corresponding signal.

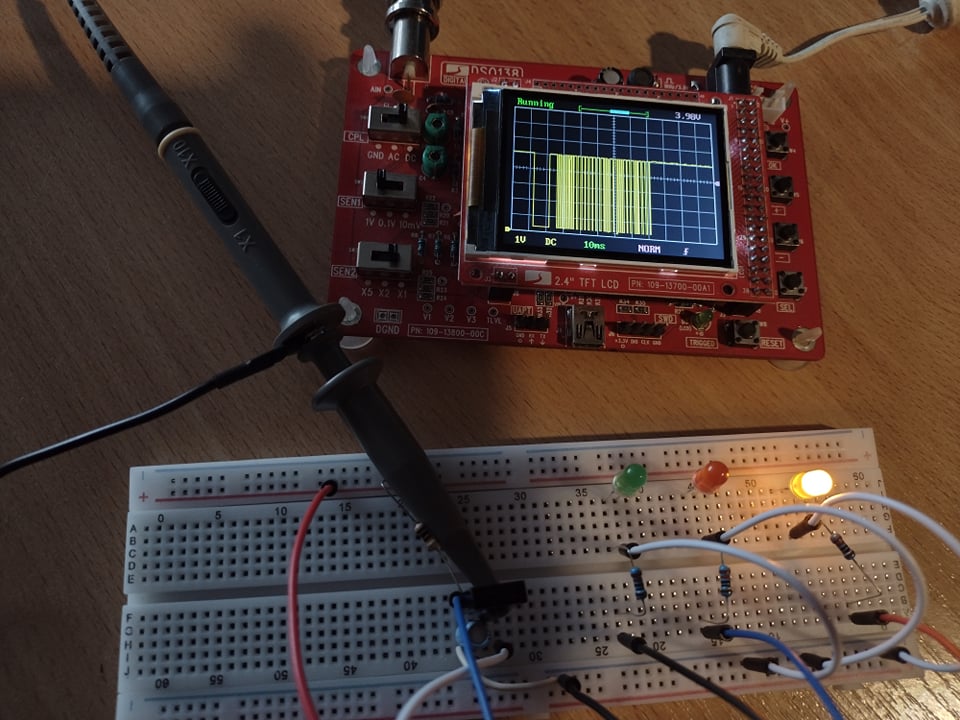


Photo nr. 3 – yellow diode switch on

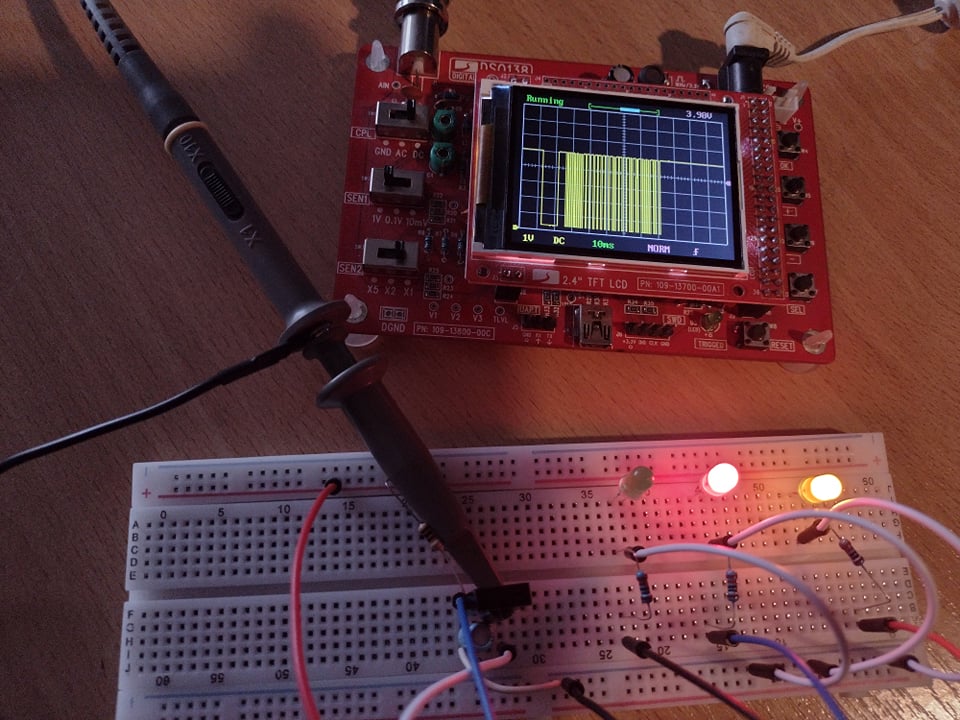


Photo nr. 4 – red diode switch on

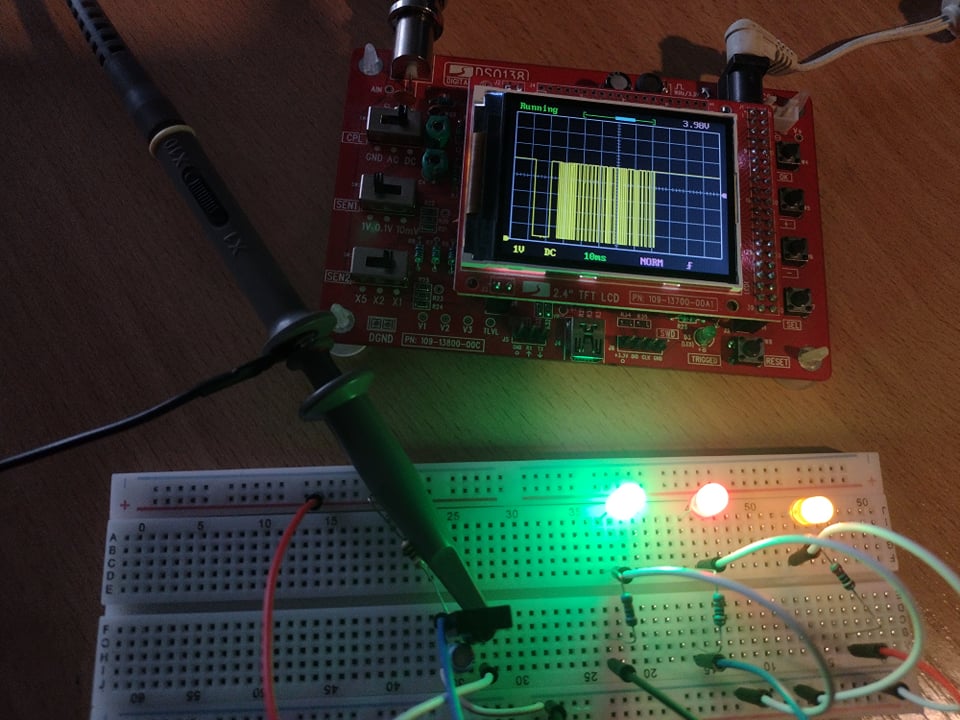


Photo nr. 5 – green diode switch on

As seen in the in the photos above, every diode was switched on by pressing different button on IR pilot. On oscilloscope display we see different waveform read from the IR receiver with each button. After holding the button, led switches on or off depending on what condition it was in, and nothing more happen, so it works correctly.

In case that 5 V DC power supply is cut off for a while, program will be restarted, and the relays settings will revert to the default.

In case that 230 V AC power supply is cut off, devices will be turn off and after the power is restored, it will return to the state it was in before turning off or different if we press some buttons in that time.

In case the receiver didn’t catch the whole signal, user must press button again and for next pressing it will be working properly, so everything is as it was meant to be.

**Measurements:**

Average current with all relays off is 25 mA

Average current with all relays on is 260 mA

# 10. Final version after testing and real life application:



Photo nr. 6 – final version of the project

Before connecting power supply, it was checked if there are no unexpected circuit breaks, also everything was correct.

Then a simple test with light bulb connected to the first power outlet was conducted.

Obraz zawierający łącznik, adapter

Opis wygenerowany automatycznie

Photo nr. 7 – light bulb connected to the outlet

Obraz zawierający wewnątrz

Opis wygenerowany automatycznie

Photo nr. 8 – light bulb on

After checking with other devices powered by 230 V AC connected to other power outlet, control works also correctly.