|  |  |  |
| --- | --- | --- |
| Logo__SSPU_2016_Barva | | |
| **Závěrečná studijní práce**  **dokumentace** | | |
| **Laserová závora** | | |
| Jakub Klíštinec | | |
| https://i.ytimg.com/vi/xS3ipW3-cMU/hqdefault.jpg | | |
|  | |  |
| **Obor:** | 18-20-M/01 INFORMAČNÍ TECHNOLOGIE  se zaměřením na počítačové sítě a programování | |
| **Třída:**  **Školní rok:** | IT4  2018/2019 | |

##### **Poděkování**

*Chtěl bych poděkovat panu Ing. Petru Grussmannovi za pomoc při řešení určitých problémů, panu Mgr. Godovskému za mikročip a panu Ing. Jiřímu Miekischovi za konzultace. Rovněž musím poděkovat svému strýci za pomoc se sháněním správných součástek.*

Prohlašuji, že jsem závěrečnou práci vypracoval samostatně a uvedl veškeré použité   
informační zdroje.

Souhlasím, aby tato studijní práce byla použita k výukovým účelům na Střední průmyslové   
a umělecké škole v Opavě, Praskova 399/8.

V Opavě 14. 1. 2019

*podpis autora práce*

**ANOTACE**

Laser svítící na odrazku a z odrazky následně na fototranzistor slouží jako bezpečné chráněné území, jejímž výsledkem je zmenšenina jedné místnosti. Laser je napájen na 6V z externího zdroje a je ovládán Wi-Fi mikročipem D1 Mini Pro ESP8266-12, který vysílá přerušovaný signál tak, aby zamezil jakémukoliv zneužití a možnému vniknutí. Vyhodnocování tohoto signálu nemá na starosti pouze již zmiňovaný mikročip, ale také komparátor. V případě jakéhokoliv narušení bezpečnosti je okamžitě informován vlastník skrze odesílání dat na internet za pomoci MQTT serveru. Program pro bezchybnou funkci tohoto projektu byl napsán v programu zvaném Visual Studio Code s prostředím PlatformIO IDE v programovacím jazyce C++.

**Klíčová slova**

Laser, Fototranzistor, ESP 8266, přerušovaný signál, MQTT server, PlatformIO IDE.

**ANOTATION**

The laser shining on the reflector and the reflector on the phototransistor as a safe protected area results in a diminution of the room. The laser is 6V powered from an external power source and is controlled by the D1 Mini Pro ESP8266-12 Wi-Fi microprocessor which emits the intermittent signal to prevent any misuse and possible intrusion. Evaluating this signal is not only the microchip mentioned above, but also the comparator. In case of any breach of security, the owner is immediately informed by sending data to the Internet using the MQTT server. A flawless program for this project was written in a program called Visual Studio Code with the PlatformIO IDE in the C ++ programming language.

**Key words**

Laser, Phototransistor, ESP 8266, intermittent signal, MQTT server, PlatformIO IDE.

OBSAH

[Úvod 6](#_Toc535166717)

[1 Využité součástky a software 7](#_Toc535166718)

[1.1 Klíčová komponenta 7](#_Toc535166719)

[1.1.1 ESP D1 Mini Pro 7](#_Toc535166720)

[1.2 Vysílač a přijímač 7](#_Toc535166721)

[1.3 VS Code a PlatformIO 7](#_Toc535166722)

[1.4 Ostatní součástky 8](#_Toc535166723)

[1.4.1 Rezistory 8](#_Toc535166724)

[1.4.2 Piezoelektrický bzučák 8](#_Toc535166725)

[1.4.3 Komparátor 9](#_Toc535166726)

[1.4.4 Trimr 9](#_Toc535166727)

[1.4.5 Tlačítko 10](#_Toc535166728)

[2 Postupy řešení 11](#_Toc535166729)

[2.1 Zapojení 11](#_Toc535166730)

[2.1.1 Schéma 11](#_Toc535166731)

[2.2 Ukázky kódu 13](#_Toc535166732)

[2.2.1 Přerušovaný signál 14](#_Toc535166733)

[3 ŘEŠENÍ PROBLÉMŮ 17](#_Toc535166734)

[3.1 Linux 17](#_Toc535166735)

[3.1.1 VirtualBox 17](#_Toc535166736)

[3.2 Nekvalitní čip 17](#_Toc535166737)

[3.2.1 LoLin NodeMCU 17](#_Toc535166738)

[3.3 Nezvládnuté postupy 18](#_Toc535166739)

[Odesílání dat 18](#_Toc535166740)

[4 Konečná úprava 19](#_Toc535166741)

[4.1 Vzhled 19](#_Toc535166742)

[Závěr 21](#_Toc535166743)

[Seznam použitýCH INFORMAČNÍCH ZDROJů 22](#_Toc535166744)

Úvod

Cílem tohoto projektu bylo vytvořit chráněný prostor za pomocí jednoduchých a levných součástek, které jsou běžně k sehnání. Například laser byl vyndán z malé propisovací tužky a následně lehce upraven. Ostatní součástky jsou rovněž velice levné. Jediný problém je se sehnáním kompárátoru, který se shání trochu hůře, bylo by teda lepší jej dopředu objednat přes internet.

Ze začátku bylo mou hlavní prioritou udělat jednoduchou závoru, která spustí alarm v případě narušení. Později pak přicházela postupná vylepšení, jako například možnost informovat vlastníka v případě narušení bezpečnosti či přerušovaný signál, který způsobuje, že jen a pouze jedna frekvence z vysílače je přijímána. Rovněž jsem byl nucen změnit mikročip za jiný, abych s pomocí konzole zjistil, jak fototranzistor běží a jak snímá světlo.

Ve své dokumentaci se hned ze začátku budu věnovat podrobněji součástkám a také softwaru. Následně detailně přiblížím své zapojení a některé zajímavé části mého kódu. Ve třetí části se budu věnovat řešeným problémům, následně vzhledu a v samotném závěru budu věnovat pozornost výsledkům mého snažení.

# 1 Využité součástky a software

## Klíčová komponenta

### ESP D1 Mini Pro



Obrázek : ESP D1 Mini Pro

Tento typ ESP umí na své rozměry spoustu věcí a funguje bezchybně a rychle. Pro spojení se zařízením stačí použít normální USB kabel. Má zde zabudovanou anténu a rovněž umí odeslat z jednoho pinu až 5V. Jinak však přijímá 3,3V jako drtivá většina všech ostatním mikročipů z řad ESP 8266.

## Vysílač a přijímač

Jako vysílač jsem použil laser vytažený z propisovací tužky, který stál asi 10Kč. Je napájený přímo z ESP na 3,3V. Přijímačem je fototranzistor typu LTR-3208, který bez problémů stačí s maximem až do 30V na využití u tohoto projektu.

## VS Code a PlatformIO

Visual Studio Code je volně dostupný neplacený editor, v kterém může uživatel programovat. Zajímavější je však vývojové prostředí, kterému říkáme PlatformIO. Je opět volně stažitelné na oficiálních stránkách PlatformIO a je podporované většinou operačních systémů. Je velmi výhodné, že je zde možnost kompilovat pro více druhů desek. Můžete také programovat v C++ a celé prostředí se chová velmi chytře, protože napovídá uživateli, co zřejmě chce zadat.

## Ostatní součástky

### Rezistory



Obrázek : Rezistor

Rezistory jsou zapotřebí snad úplně všude. Nejčastěji jsem zde používal 300Ω anebo 1 a 10 kΩ.

### Piezoelektrický bzučák



Obrázek : Piezoelektrický bzučák

Tuto součástku jsem použil jako výstražné zařízení, které upozorní na přerušení paprsku. V testovacích verzích je dobré si tuto součástku nahradit například LED diodou, protože tyto zvuky ze bzučáku nejsou příliš příjemné.

### Komparátor



Obrázek : Komparátor

Toto zařízení porovnává dvě hodnoty a vyhodnocuje přerušovaný signál a podle toho pozná, jestli na přijímač svítí správné světlo, či ne. Tato součástka, která nese označení typu LM393N, má dohromady 8 pinů a zdá se být složitá. Je tomu tak, protože v této součástce je možné komparátory vytvořit dva. V mém případě stačí pouze jeden. Využijete tedy pouze pět pinů a to: zemi, kladné napětí (obě jsou pro oba komparátory stejné) a následně dvě hodnoty, které chcete porovnávat a poslední vrací hodnoty do mikročipu.

### Trimr



Obrázek : Trimr

Trimr je rezistor s proměnným odporem, jednoduše nastavitelný rezistor. V mém případě jsem trimr 500Ω použil jako dělič napětí. Funguje tak, že šroubovákem jednoduše potočíte na vršku součástky.

### Tlačítko



Obrázek 6: Tlačítko

Nemohu zapomenout také na tlačítko, které se hodí v případě nějakého restartu či vypnutí alarmu.

Rovněž jsou zapotřebí propojovací vodiče, když pracujete na testech na nepájivém kontaktním poli.

# Postupy řešení

## Zapojení

### Schéma

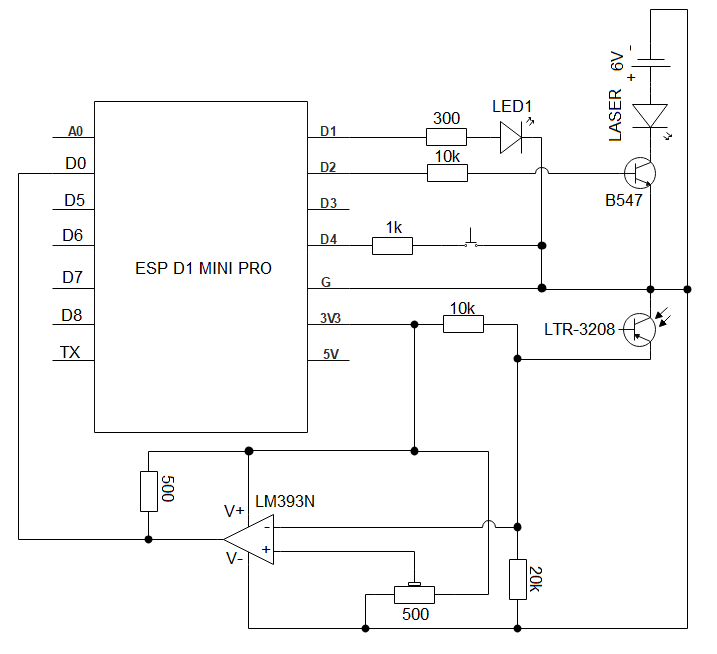


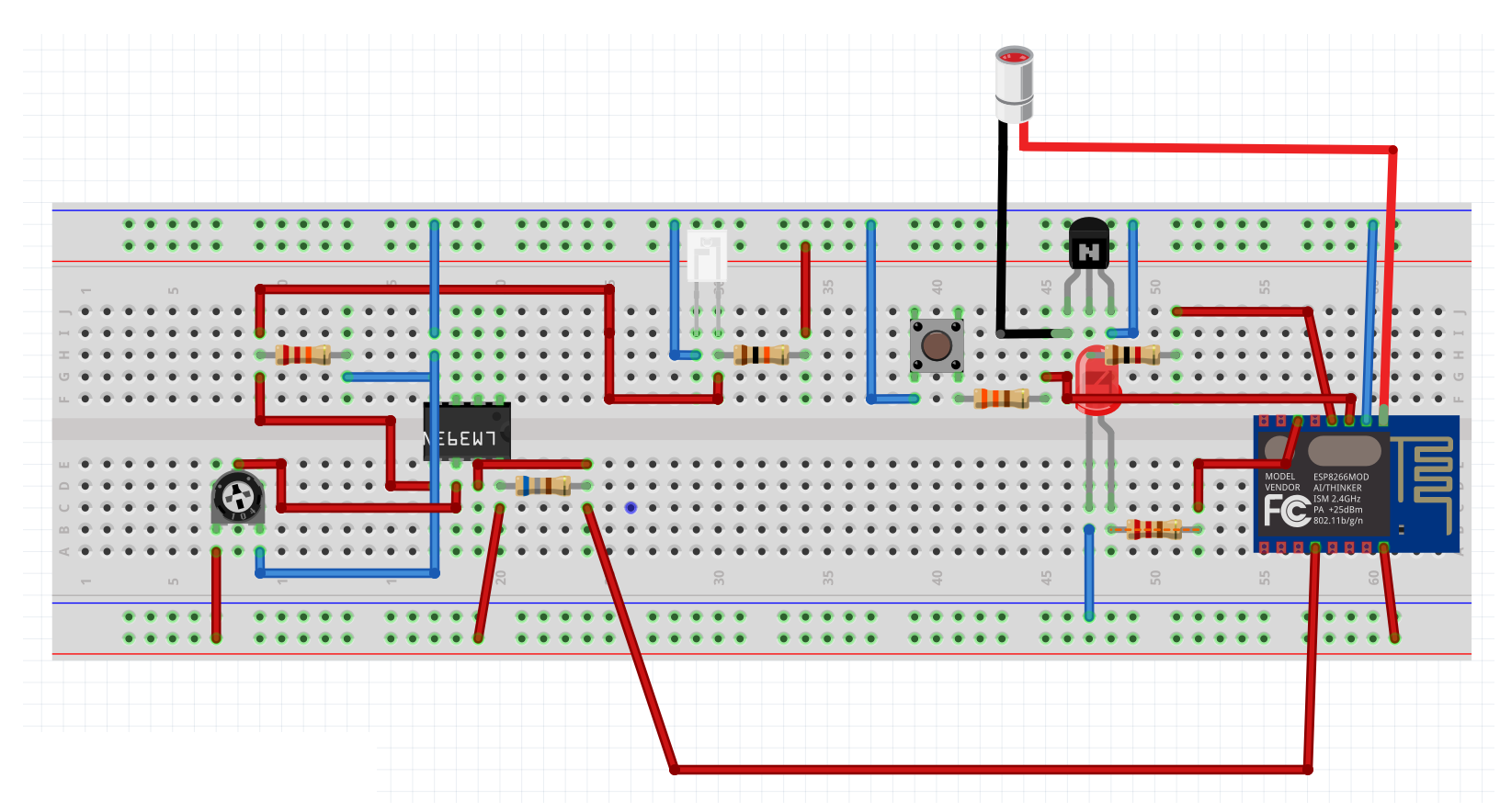
Schéma bylo nakresleno v programu EDraw Max.

LTR-3208 … Fototranzistor

LM393N … Komparátor

B547 … Tranzistor typu NPN

Pomocí tranzistoru je spínáno za pomoci externího zdroje napětí šesti voltů, kvůli dostatečnému napájení laseru. V konečném výsledku je LED1 nahrazena piezoelektrickým bzučákem.



Jednodušší zpracování zapojení na nepájivém poli navržené v programu Fritzing. Toto zapojení je téměř stejné až na to, že jsem se rozhodl zde vést laser přímo z ESP z 5V. Můj laser je sice zapojen na 6V z externího zdroje, ale to jen proto, že to je opravdu slabý laser. Normálně bude stačit 5V.

## Ukázky kódu

Tato část kódu je zaměřena na první pokusy s fototranzistorem, kdy ještě data nevyhodnocoval komparátor. Fototranzistor v mém případě funguje tak, že snímá světlo od hodnot 0 – 1024 (čím více světla, tím menší číslo). Na testech jsem zjistil, že v normálně osvětlené místnosti udává fototranzistor hodnoty okolo 850 – 950. (to znamená, že na přijímač skoro žádné světlo přímo nesvítí). Stanovil jsem teda jednoduchou podmínku, že když bude hodnota pod 600 (tj. laser výrazněji svítí na fototranzistor, vypíše se mi to do konzole a LED dioda zůstane vypnutá, neboť prostor není narušen. Když laser svítí přímo na fototranzistor, udává obvykle hodnoty okolo 50. Jakmile nesvítí nic, spustí se druhý blok příkazů, který se nachází pod příkazem „else“. Tam nechybí opět výpis do konzole, rozsvícení ledky a nastavení proměnné „trigger“ typu boolean na hodnotu false. Ta je zde kvůli druhé části kódu a tím je resetovací tlačítko. To zde funguje tak, že když laser na přijímač nesvítí, zapne se LED dioda. Poté odejmu překážku a laser opět svítí na fototranzistor, avšak LED dioda zůstane vypnutá. Proto je nutné stisknout tlačítko v moment, kdy už mezi vysílačem a přijímačem není žádná překážka a LED dioda se opět vypne.

void loop()

{

ldr\_value = analogRead(ldr); //čte hodnoty LDR

Serial.println(ldr\_value); //zobrazí hodnoty LDR na sériové lince

if (trigger){

if (ldr\_value < 600) {

Serial.println("mensi nez 600");

//digitalWrite(LED, LOW);

}

else {

Serial.println("vetsi nez 600");

trigger = false;

digitalWrite(LED, HIGH);

}

}

else{

if(digitalRead(BUTTON) == LOW){

trigger = true; //tlačítko je sepnuté

digitalWrite(LED, LOW);

Serial.println("zmackl jsi tlacitko");

}

else{

Serial.println("Nezmackl jsi tlacitko");

}

}

delay(500);

}

### Přerušovaný signál

ESP kontroluje laser. Laser se zapíná a vypíná na dobu 20 milisekund. Fototranzistor přijme signál z laseru, který dále pokračuje přes komparátor. Na komparátoru máme nastavené referenční napětí. Toto napětí nám určuje minimální intenzitu paprsku laseru a následné zajištění správného počítání (nahrazení funkce analogRead z předchozího kódu). Referenční napětí je nastaveno pomocí trimru, je nastaveno experimentálně. Mikročip umí vyhodnotit pouze dva druhy signálu (zapnuto/vypnuto) a tento signál zajišťuje komparátor.

Deklarace proměnných na dané hodnoty.

int LED = 5;

int BUZZ = 12;

int LASER = 4;

int BUTTON = 2;

volatile bool trigger = true;

const byte Comparator = 14;

volatile unsigned int period = 20, eror = 1;

volatile unsigned int time\_now, time\_interupt;

bool LASER\_ON = true;

volatile bool blink = true;

Nastavení vstupů a výstupů.

attachInterrupt je nastavení přerušení na daný pin a danou funkci(Interupt). Poslední dva řádky nastavují výchozí časy.

void setup()

{

Serial.begin(9600); //spuštění serioveho rozhrani

pinMode(LED, OUTPUT);

pinMode(BUZZ, OUTPUT);

pinMode(LASER, OUTPUT);

pinMode(BUTTON, INPUT);

pinMode(Comparator, INPUT\_PULLUP); //přerušení

digitalWrite(LASER, HIGH);

digitalWrite(LED, HIGH);

attachInterrupt(digitalPinToInterrupt(Comparator), Interupt , CHANGE);

time\_now = millis(); //vysílání laseru

time\_interupt = millis(); // přijímání z fototranzistoru

}

První podmínka se týká nastavení blikání laseru.

Druhá kontroluje, jestli není spojení mezi vysílačem a přijímačem narušeno nebo jestli do vysílače nesvítí jiné světlo.

void loop()

{

if (millis() >= time\_now + period){ //kontrola vysílače

//Serial.println(LASER\_ON ? "Zapnuto": "Vypnuto");

//Serial.println(millis() - time\_now);

time\_now = millis(); //nastavení výchozího času

LASER\_ON ? digitalWrite(LASER, LOW) : digitalWrite(LASER, HIGH); //vypnutí/zapnutí laseru

LASER\_ON = !LASER\_ON; //uložení aktuálního stavu do paměti

}

if (millis() >= time\_interupt + 10\*period){ //kontrola, že dlouho nepřišel signál (reaguje na změnu)

digitalWrite(LED, LOW); //vypnu ledky

digitalWrite(BUZZ, HIGH); //zapnu bzučák

trigger = false; //nezapezpečeno

time\_interupt = millis(); //posunutí času, aby se podmínka nespouštěla po každém průchodu, ale po každém 10ém. Kdyby se podmínka vyhodnocovala pořád, nikdy by se nevyhodnotila funkce Interupt z důvodu trigger = false

}

}

Funkce Interupt() slouží k zjištění zabezpečení a opětovné aktivaci systému v případě narušení. Kontrola zabezpečení se provádí měřením času mezi jednotlivými změnami signálu.

void Interupt()

{

if (trigger){ //zabezpečení ?

if (blink){ //první průchod - nastavení času

time\_interupt = millis(); //nastavení času

blink = !blink; //vypnout fuknci

}

else if (millis() <= time\_interupt + period + eror && millis() >= time\_interupt + period - eror){ //kontrola správného intervalu signálu

Serial.println(millis() - time\_interupt); //pomocný výpis ke kontrole periody

time\_interupt = millis(); //nastavení nového času

}

else{

Serial.println("Spatny cas");

Serial.println(millis() - time\_interupt); //výpis špatného času

digitalWrite(LED, LOW); //vypnutí ledek

digitalWrite(BUZZ, HIGH); //zapnutí alarmu

trigger = false; //prostor je narušen (podmínka trigger nebude probíhat)

}

}

else{

if(digitalRead(BUTTON) == LOW){ //je zmáčknuté tlačítko (Ground zapojen na pin = LOW místo HIGH)

blink = !blink; //synchronizace času (funkce blink proběhne znovu a nastaví se čas)

digitalWrite(LED, HIGH); //zapnutí ledek

digitalWrite(BUZZ, LOW); //vypnutí bzučáku

trigger = true; // prostor zabezpečen

Serial.println("Zmackl jsem tlacitko"); //ověření

}

}

}

# ŘEŠENÍ PROBLÉMŮ

## Linux

### VirtualBox

Ne každý má po ruce notebook s možností nahrát si na něj Linux a začít pracovat, tak jsem se rozhodl, že to zkusím nejprve ve VirtualBoxu. Zprvu to šlo hladce. Nainstaloval jsem editor a vývojové prostředí. Následně napsal jednoduchý kód, ale hlásil mi spoustu chyb. Některé z nich jsem vyřešil, jiné však nikoli. Po asi 14 dnech jsem své snažení vzdal a požádal o pomoc pana učitele. Ani po konzultacích jsem však nebyl úspěšný a od tohoto postupu jsem musel upustit.

## Nekvalitní čip

### LoLin NodeMCU



Obrázek 7: ESP LoLin NodeMCU

Původně jsem začínal s právě tímto typem ESP. Fungovalo správně a spoustu věcí jsem si na něm odzkoušel. Avšak po měsíci, kdy už jsem se dostal k fototranzistoru a měření hodnot, potřeboval jsem vidět výpis, co mi to hlásí. Stejný problém bych měl později, když bych měl složitější kód, tak je dobré si do každých částí psát nějaké hlášky, abyste přesně věděli, kde program ztroskotal. A právě tento mikročip to neuměl nebo jsme na to s panem učitelem nepřišli. Po několika týdnech jsem byl nucen začít pracovat s jiným mikročipem a pan učitel byl ochoten mi poskytnout jiný, právě již výše zmíněný D1 Mini Pro.

## Nezvládnuté postupy

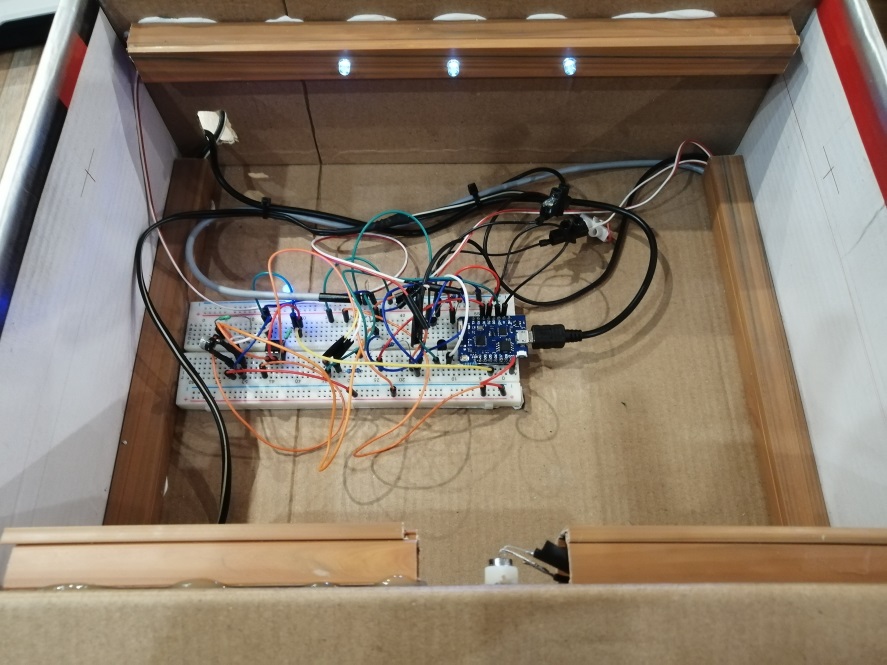
### Odesílání dat

Jako úkol navíc jsem dostal Homeassistant a odesílání dat na internet či na email v případě, že je prostor narušen. Z časových důvodů a neustálých trablí s PlatformIO a Linuxem jsem se k tomuto kroku zatím vůbec nedostal.

# Konečná úprava

## Vzhled

Jako místnost jsem použil krabici od bot, do ní jsem lehce zalepil nepájivé pole, udělal vývod pro externí zdroj a napájení mikročipu a také se mi podařilo schovat některé dráty s pomocí elektrikářských lišt, které jsem připevnil lepidlem zvaným Mamut. Krabice je zvenčí nastříkána sprejem na šedo.



Obrázek 8: Pohled dovnitř

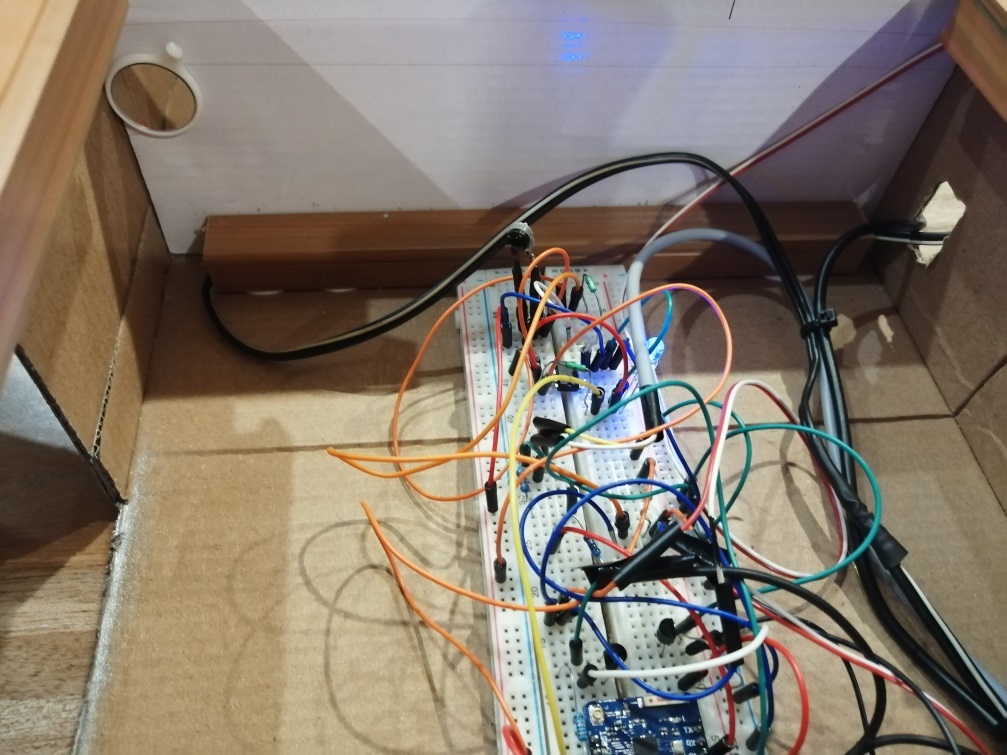


Obrázek 9: Pohled zvenčí

Velmi složitá část byla napasovat laser na odrazku tak, aby se odrazil přesně do fototranzistoru, který je opravdu velmi malý. Nakonec se to však podařilo, a když se s krabicí moc nehýbe, vše funguje tak, jak má.

**

Obrázek 10: Detailní pohled na Laser (dole) a fototranzistor (nahoře)



Obrázek 11: Detailní pohled na odrazku

# **Závěr**

Cílem projektu bylo vytvořit zabezpečovací systém pro místnost za levné peníze, který by v případě narušení bezpečnosti zapnul alarm. Rovněž je zde pojistka, že se přes alarm nedostanete tak snadno díky přerušovanému signálu. Bohužel jsem nestihl dokončit odesílání dat na internet.

Do budoucna mám rozhodně plány tento projekt zcela dokončit. Začal bych rozhodně již zmíněným odesíláním dat na internet. Jednoduché základy už v tom mám, protože jsem to ve škole několikrát zkoušel. Chtěl bych, aby mi data přicházela na email. Následně bych vše zapájel na dvě desky, ať to taky nějak vypadá z blízky a ať se s celým zařízením lépe manipuluje. Rovněž bych zaměnil krabici za nějaký dřevěný výrobek, protože jak mám nešikovně umístěné tlačítko, tak v případě stisknutí pro reset se krabice lehce promáčkne a tím pádem se posune laser. V manipulaci s tím musím být opravdu opatrný. Poté bych se vrhl do opravdu reálného využití a to tak, že bych koupil silnější laser, upravil napájení a alarm bych mohl umístit například do garáže. V současné době se domnívám, že by to tento laser a ještě k tomu s daným slabším napájením nezvládl.

Na svém githubu mám rovněž pokus o plošný spoj udělaný v programu Fritzing. Do budoucna bych jej chtěl trochu vylepšit a zpřehlednit. Také tam najdete kódy, kde mám své pokusy, které kvůli nefunkčnosti nebyly v samotném kódu použity. Hlavní kód, který je na mém githubu ve složce „testtt“ se zabývá právě přerušovaným signálem.

Odkaz zde: https://github.com/it1512/projekt

Seznam použitýCH INFORMAČNÍCH ZDROJů

[] Arduino s fotorezistorem (světelný sensor). [online] [cit. 2019-01-06].

<http://arduinonavody.cz/arduino-s-fotorezistorem-svetelny-senzor/>

[2] Why is apt-get always using proxy [online] [cit. 2019-01-06].

<<https://askubuntu.com/questions/344802/why-is-apt-get-always-using-proxy-although-no-proxy-is-configured>>

[3] Homie dual relay [online] [cit. 2019-01-06].

<https://github.com/petrgru/homie-dual-relay>

[4] Laser data transfer [online] [cit. 2019-01-06].

<https://www.instructables.com/id/Laser-data-transfer-part2-Binary-transmitting//>

[5] Laserová závora – nová verze [online] [cit. 2019-01-06].

<https://www.gme.cz/data/attachments/orn.760-492.1.pdf>

[6] Úvodní obrázek - laser [online] [cit. 2019-01-06].

<https://i.ytimg.com/vi/xS3ipW3-cMU/hqdefault.jpg>

[7] Obrázek 1 [online] [cit. 2019-01-06].

< http://www.neven.cz/user/7262/upload/stuff/17892536.jpg>

[8] Obrázek 2 [online] [cit. 2019-01-06].

< https://www.gme.cz/data/product/1024\_1024/pctdetail.110-097.1.jpg

[9] Obrázek 3 - laser [online] [cit. 2019-01-06].

< <https://www.vpcentrum.eu/image/cache/data/prods/prod7/290677-piezobzucak-13-8mm-fy-14-napajeni-3-18v-7ma-1-500x500.jpg>>

[10] Obrázek 4 – LM393N [online] [cit. 2019-01-06].

< https://www.ges.cz/cz/lm393n-GES05001247.html>

[11] Obrázek 5 – Uhlíkový trimr PT10MVE500 [online] [cit. 2019-01-06].

< https://www.gme.cz/pt10mve500>

[12] Obrázek 6 – Mikrospínač 6x6x6 mm tlačítko [online] [cit. 2019-01-06].

< https://arduino-shop.cz/arduino/1253-mikrospinac-6x6x6-mm-tlacitko-1459621295.html?gclid=EAIaIQobChMI6ev7x9zZ3wIVAud3Ch0\_twKPEAQYCSABEgI-MPD\_BwE>

[13] Obrázek 7 [online] [cit. 2019-01-06].

< <https://www.aerial.net/shop/imageslarge/IOT-Lolin_main.jpg>>

[14] Github – it1512 - projekt [online] [cit. 2019-01-06].

< https://github.com/it1512/projekt>