

Gymnázium, Nový Jičín, příspěvková organizace  
Palackého 50, 741 01 Nový Jičín

# **DIY zabezpečení rodinného domu**

**maturitní práce**

**Radim Kaděra, 6. G**

**2019 - 2020**

Prohlašuji, že jsem maturitní práci zpracoval samostatně a uvedl v ní všechny prameny, literaturu a ostatní zdroje, které jsem použil.

Souhlasím s využitím své práce pro studijní, výukové a propagační účely Gymnázia v Novém Jičíně.

V Novém Jičíně 30. března 2020

-----  
Radim Kaděra

Chtěl bych poděkovat vedoucímu maturitní práce panu Ing. Michalovi Hlostovi za jeho rady a čas, který mi věnoval při řešení dané problematiky. Děkuji panu Ondřeji Kopeckému za pomoc při 3D tisku. Děkuji všem ostatním, kdo mi při tvorbě mé práce jakýmkoliv způsobem pomohli.

V Novém Jičíně 19. března 2020

-----  
Radim Kaděra

### ***Anotace***

*Tato maturitní práce pojednává o vytvoření DIY zabezpečení rodinného domu za pomoci kamery, PIR čidla a webové aplikace.*

*Práce je členěná na teoretickou a praktickou část. Teoretická část je zaměřená na obecné informace a znalosti potřebné ke zpracování tématu.*

*V praktické části následuje popis jednotlivých částí a jejich fungování.*

### ***Annotation***

*This graduation work deals with the creation of DIY security of a family house using a camera, PIR sensor and web application.*

*The work is divided into theoretical and practical part. The theoretical is focused on general information and knowledge needed to process the topic.*

*The practical part is followed by a description of the individual parts and their operation.*

***Klíčová slova***

*databáze, ESP32, ESP32-CAM, ESP8266, HTML, MySql, webová aplikace, PHP, PIR  
čidlo, Wiring, 3D tisk*

***Keywords***

*Database, ESP32, ESP32-CAM, ESP8266, HTML, MySql, Web Application, PHP, PIR  
Sensor, Wiring, 3D Printing*

# Obsah

Úvod .....	1
<b>1 Teoretická část .....</b>	<b>2</b>
1.1 Hardware .....	2
1.1.1 ESP8266 .....	2
1.1.2 ESP32 .....	3
1.1.3 ESP32-CAM .....	3
1.1.4 HC-SR501 PIR čidlo .....	4
1.1.5 Ostatní .....	5
1.2 Software .....	6
1.2.1 Webová aplikace .....	6
1.2.2 HTML .....	6
1.2.3 PHP .....	6
1.2.4 MySQL databáze .....	7
1.2.5 Wiring .....	7
<b>2 Praktická část .....</b>	<b>9</b>
2.1 Fungování systému .....	9
2.2 Vývojová deska .....	9
2.3 Webová aplikace .....	16
2.3.1 Šablona .....	16
2.3.2 Databáze .....	18
2.3.3 Zpracování dat z mikrokontroleru .....	19
2.3.4 Přihlášení .....	24
2.3.5 Zapomenuté heslo .....	26
2.3.6 Přehled .....	29
2.3.7 Změna hesla a emailu .....	31
2.3.8 Všichni uživatelé .....	33
2.3.9 Nový uživatel .....	34
2.3.10 Zapnutí a vypnutí systému .....	35
2.3.11 Odhlášení .....	36
2.4 3D tisk .....	36
<b>Závěr .....</b>	<b>39</b>
<b>Seznam obrázků .....</b>	<b>40</b>
<b>Seznam použitých zdrojů .....</b>	<b>41</b>
<b>Přílohy .....</b>	<b>44</b>

# Úvod

Práci na téma DIY zabezpečení rodinného domu jsem se rozhodl vypracovat zejména z důvodu mého zájmu o IoT<sup>1</sup> domácnost a následné zpracování dat do webové aplikace. Jeden z dalších důvodů byla rostoucí kriminalita v oblasti mého bydliště v době zadávání maturitní práce.

Mé zkušenosti z oblasti zabezpečení domu jsou velmi malé, proto se pokusím zaměřit na propojení vývojové desky s webovou aplikací a následným zpracováním dat. Ve své práci se tudíž nezabývám následnou instalací zabezpečovacího systému v domě.

Výstupem následně bude webová aplikace zpracovávající data z ESP desky s kamerou a připojeným PIRO senzorem.

---

<sup>1</sup> IOT (Internet of things) – internet věcí

# 1 Teoretická část

## 1.1 Hardware

Důležitou součástí je ESP vývojová deska, do systému je možné připojit některou z níže popsaných. U mého funkčního vzorku využívám ESP32-CAM zejména kvůli integrované kameře, avšak pro připojení pouze čidla je dostačující ESP8266, nebo ESP32. Jejich podstatná vlastnost je připojení k internetu.

Mimo jiné je možné použít i další čidla, která uživatel považuje za důležité, například sensor teploty, či CO<sub>2</sub>. Programování desek zajišťuje open-source program Arduino IDE. Ten však desky ve své základní verzi nenabízí, proto je potřeba jejich instalace.

Všechny desky (ESP8266, ESP32 a ESP32-CAM) podporují takzvaný deep sleep, čili mód při kterém se značně sníží počet procesů a spotřeba - přibližně na 20 uA.

### 1.1.1 ESP8266

Jedná se o mikrokontroler s čipem ESP8266 od společnosti Espressif. Jeho důležitou vlastností je možnost připojení k WiFi s frekvencí 2,4 GHz, což je ideální pro IoT a DIY<sup>2</sup> projekty, kdy je možné tuto desku připojit poměrně jednoduše do sítě. Deska navíc obsahuje 17 GPIO<sup>3</sup>, to je více než dostačující pro malé projekty. Procesor obsahuje jedno jádro.

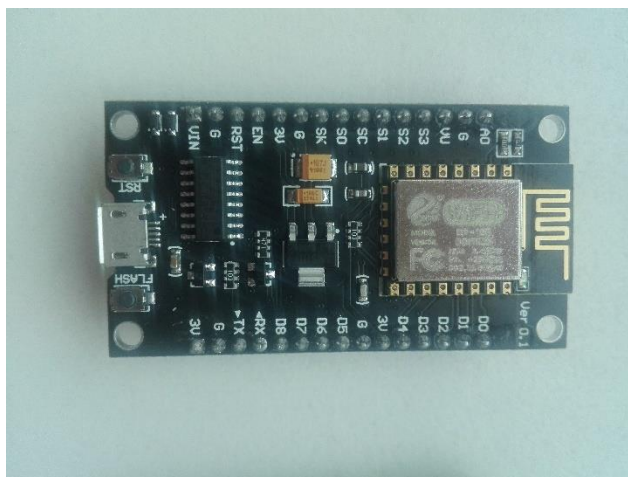
Programování a napájení je možné přes micro USB.

---

<sup>2</sup> DIY (Do It Yourself) - „udělej si sám“, vyrobení produktu samostatně, bez pokročilých odborných znalostí

<sup>3</sup> GPIO – vstupně-výstupní porty

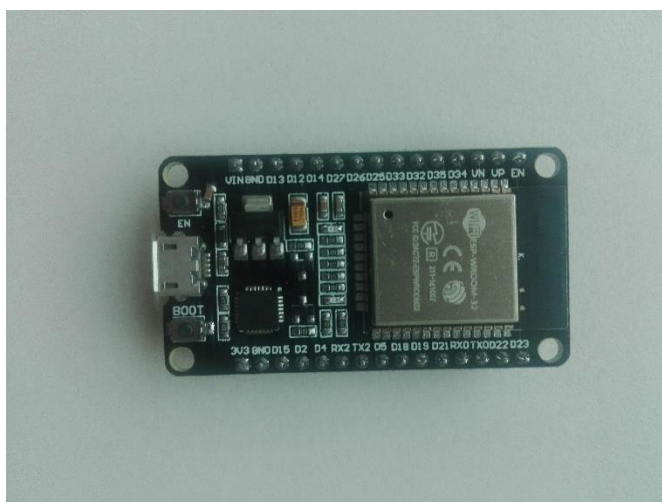




Obr. 1: Vývojová deska ESP8266

### 1.1.2 ESP32

Tato vývojová deska je nástupcem ESP8266. Disponuje novějším čipem ESP32 s lepším výkonem a novými funkcemi. Tou hlavní je Bluetooth 4,2 s podporou BLE.<sup>4</sup> Dalším rozdílem je procesor, který obsahuje dvě jádra a SRAM o velikosti 512 kB. Navýšen byl také počet GPIO a to na 36.



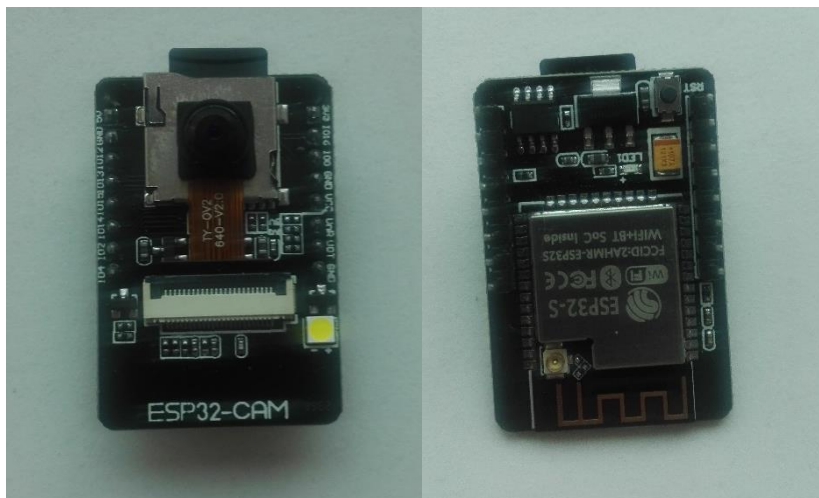
Obr. 2: Vývojová deska ESP32

### 1.1.3 ESP32-CAM

Jedná se o vývojovou desku postavenou na čipu ESP32, obsahuje tudíž také Wifi a Bluetooth. Hlavní předností je kamerový modul OV2640 s rozlišením 2MP a možností rozpoznávání obličeje. Fotografie či videa lze pak ukládat na microSD kartu díky čtečce.

<sup>4</sup> BLE (Bluetooth Low Energy) – bezdrátová Bluetooth komunikace s velmi malou spotřebou energie

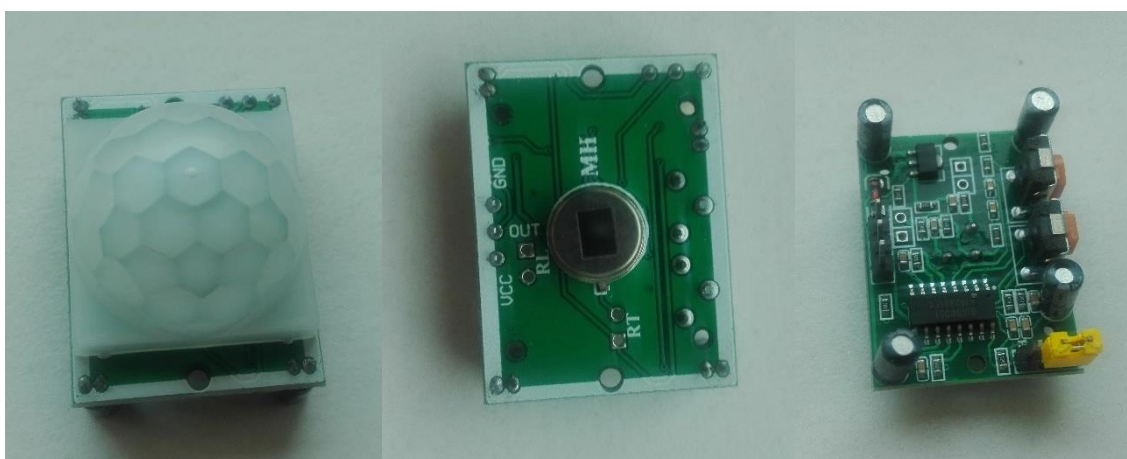
Dále deska obsahuje bílou přisvětlovací diodu. Nevýhodou je z mého pohledu nahrávání programu, pro které je potřeba využít USB-UART převodník, nebo například arduino.



Obr. 3: ESP32-CAM

#### 1.1.4 HC-SR501 PIR čidlo

Toto čidlo obsahuje pyroelektrický sensor, který funguje na principu pyroelektrické jevu. Dojde-li ve snímaném okolí k větší změně teploty, změní se také výstupní napětí čidla. Snímaný úhel je 160°. Dále je možné čidlo konfigurovat pomocí dvou trimrů, jeden ovlivňuje nejmenší snímanou vzdálenost a druhý čas po který zůstává změněné napětí na výstupním pinu. Některé typy umožňují změnu takzvané „trigger selection“, ta má dvě možnosti nastavení – každý další pohyb resetuje čas, po který je sensor sepnutý, a nebo mód, kdy jsou další detekce po dobu sepnutí blokovány.



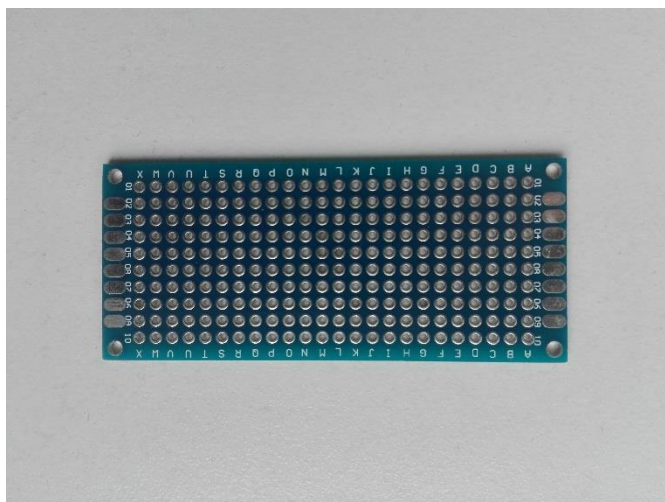
Obr. 4: PIR čidlo

### 1.1.5 Ostatní

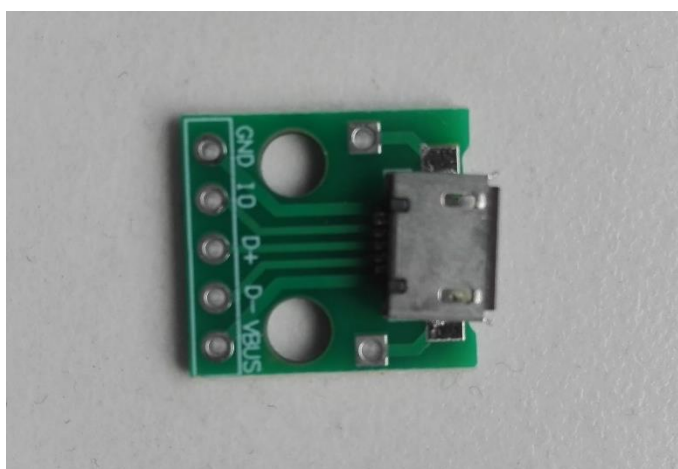
Dále v samotné konstrukci používám dva odpory hodnoty  $1\text{k}\Omega$ , pro propojení součástek pájivé pole a pro napájení adaptér microUSB na DIP.



Obr. 5: Odpor  $1\text{k}\Omega$



Obr. 6: Pájivé pole



Obr. 7: Adaptér microUSB na DIP

## 1.2 Software

### 1.2.1 Webová aplikace

Jedná se o webovou stránku, uloženou na webovém hostingu, zpracovávající data z bezpečnostního systému. Její součásti jsou přihlášení, uživatelská sekce, databáze a dále soubory, jenž zajišťují ověření uživatele, odeslání mailu a zapisování dat do MySQL databáze.

### 1.2.2 HTML

Značkový jazyk, sloužící k tvorbě základů webové stránky, poté se doplňuje zejména o kaskádové styly a php. Používají se takzvané tagy, ty jsou buď párové (<h1></h1>), nebo nepárové (<br>).

Základní šablona HTML dokumentu pak vypadá takto.

```
<html>

<head>

<title></title>

</head>

<body>

</body>

</html>
```

Do hlavičky (head) pak patří zejména informace pro prohlížeč, například jazyk stránky, autor, link na kaskádové styly a také tag title, do kterého se vkládá název stránky zobrazující se v záložce

Tělo stránky (body) obsahuje všechnu vlastní text, jenž se poté na stránce může zobrazovat a pomocí html formátovat.

### 1.2.3 PHP

Nezbytnou součástí každé dynamické webové stránky je skriptovací programovací jazyk php. Skripty jsou provedeny na straně serveru a poté putují k uživateli. Díky php je pak

možné se připojit například k MySQL databázi, pracovat se soubory, posílat emaily a jiné. Ukázka kódu:

```
<?php  
  
$pozdrav="Hello, world!";  
  
echo $pozdrav;  
  
?>
```

Zajímavostí je, že v php není potřeba při tvorbě proměnné udávat její datový typ.

#### 1.2.4 MySQL databáze

Databázový server, příkazy jsou prováděny jazykem SQL a po zprávu databáze se využívá například nástroj PhpMyAdmin. K databázi je pak možno přistupovat php kódem webové stránky, poté data vypisovat, či vkládat. Ukázka SQL příkazu pro vložení hodnot do vytvořené tabulky v databázi.

```
SELECT * FROM table_name;
```

#### 1.2.5 Wiring

Programovací jazyk sloužící k programování mikrokontrolerů jako například arduino uno, nebo již zmíněné ESP32. Je postavený na C++ a jeho základní členění je do dvou funkcí: setup() – ta se provede pouze při spuštění mikrokontroleru, loop() – cyklicky se opakující část. Ukázka:

```
void setup() {  
    pinMode(13, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(13, HIGH);  
    delay(1000);  
}
```

```
digitalWrite(13, LOW);  
delay(1000);  
}
```

V ukázce dojde k blikání LED připojené na pinu 13.

## 2 Praktická část

### 2.1 Fungování systému

Základ tvoří webová aplikace zpracovávající data. K té je možné připojit libovolný počet čidel, k tomu je zapotřebí některý z výše zmíněných mikrokontrolerů (ESP32, ESP8266, ESP32-CAM). Pro mou maturitní práci jsem zvolil ESP32-CAM. Deska při zachycení pohybu zachytí snímek a pošle jej webové aplikaci ke zpracování. Podrobnější popis fungování následuje v popisech jednotlivých částí.

### 2.2 Vývojová deska

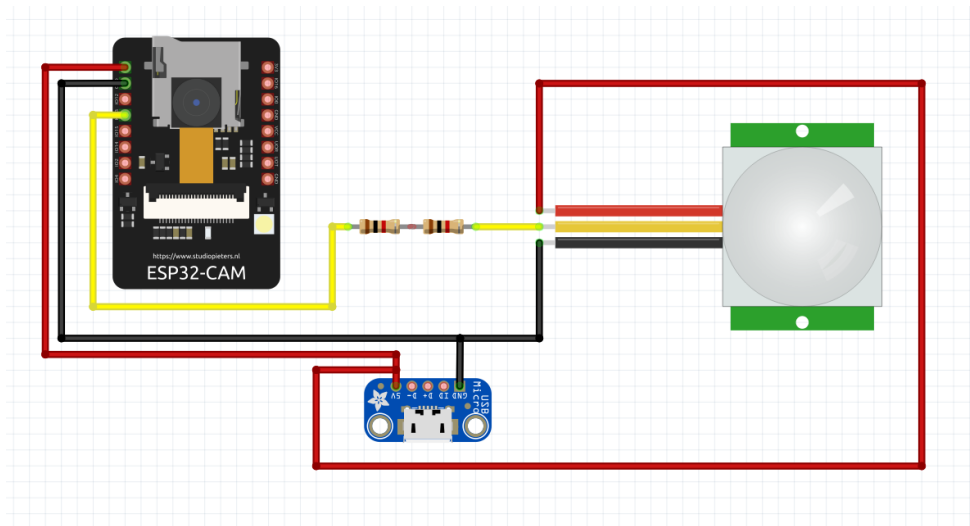
Základem pro mě byl již vytvořený projekt<sup>5</sup>, který mě na tento nápad přivedl. Ten nabízí zachycení pohybu čidlem PIR, pořízení snímku, jeho následné uložení do microSD karty a poté zaslání emailu uživateli. Tyto vlastnosti jsem pozměnil a například email klienta nevyužívám vůbec.

V původním zapojení počítalo s propojením výstupu PIR sensoru přes kondenzátor. Kondenzátor jsem nepoužil, následně pak bylo nutné v programu změnit hodnotu kdy se ESP32-CAM vzbudí z deep sleep u pinu 13 z 0 na 1.

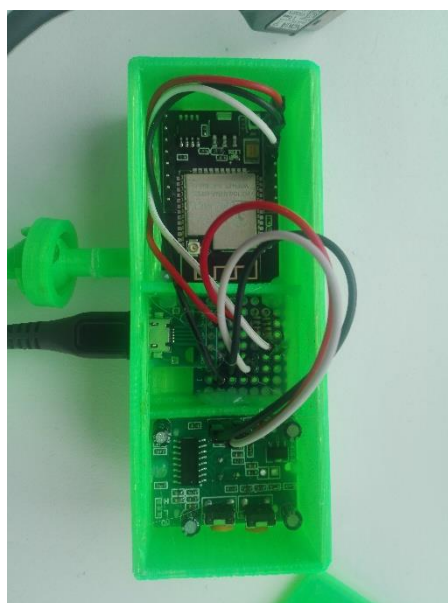
```
esp_sleep_enable_ext0_wakeup(GPIO_NUM_13, 1);
```

---

<sup>5</sup> Motion Triggered Image Capture and Email. *Instructables circuit* [online]. [cit. 2020-04-15]. Dostupné z: <https://www.instructables.com/id/Motion-Triggered-Image-Capture-and-Email/>



**Obr. 8: Schéma zapojení**



**Obr. 9: Zapojení**

Výchozí program bylo potřeba značně upravit. Přidal jsem odesílání obrázku na ftp server. Obrázky se pojmenovávají podle času, kdy byly zachyceny, což zajišťuje připojení mikrokontroleru k NTP serveru. Dále deska po probuzení z deep sleep pomocí http klienta pošle dotaz metodou POST (data tedy nejsou přímo v URL adrese, jako je tomu u metody GET) a zjistí, zda je systém spuštěný a má zachytit snímek, nebo přejít znovu do módu deep sleep.

V následující části popíšu důležité funkce programu a některé proměnné.



V ukázce dochází k nastavení údajů pro připojení k NTP serveru, lokální wifi síti a ftp serveru.

```
const char* ntpServer = "pool.ntp.org";

const long  gmtoffset_sec = 3600;

const int   daylightOffset_sec = 3600;

#define WIFI_SSID          "wifi_username"

#define WIFI_PASSWORD      "wifi_password"

char ftp_server[] = "files.000webhost.com";

char ftp_user[]   = "ftp_username";

char ftp_pass[]   = "ftp_password";
```

Následuje nastavení adresy pro http klienta a důležitá část, která obsahuje klíč, který je pak kontrolován webovou aplikací. Dále definuji umístění mikrokontroleru a typ čidla, o které se jedná v tomto případě kamera.

```
const char* serverName =

"http://home363.000webhostapp.com/post-esp-data.php";

String key = "tPmAT5Ab3j7F9";

String location = "Chodba";

String type = "camera";
```

Mikrokontroler se v dalším kroku připojí k wifi a ověří, zda je systém zapnutý či nikoliv. To proběhne pomocí http klienta, jak je vidět v následující části kódu, kdy klient otevře stránku webové aplikace, odešle potřebné data a výstupem je poté odpověď on, či off, podle které vyhodnotí, zda má dále pokračovat, nebo přejít do deep sleep.

```
HTTPClient http;
```

```

http.begin("http://home363.000webhostapp.com/on_of.php");
http.addHeader("Content-Type", "application/x-www-form-
urlencoded");
String httpRequestData = "key=" + key + "&type=on_off";

int httpResponseCode = http.POST(httpRequestData);
if (httpResponseCode>0) {
    Serial.print("HTTP Response code: ");
    Serial.println(httpResponseCode);
    String payload= http.getString();
    payload.trim();
    http.end();
    if(payload=="on"){
        //pokračování kódu
    }else{
        Serial.println("off");
    }
}
else {
    Serial.print("Error code: ");
    Serial.println(httpResponseCode);
}

pinMode(4, OUTPUT);
digitalWrite(4, LOW);
rtc_gpio_hold_en(GPIO_NUM_4);
Serial.println("Entering deep sleep mode");

```

```

Serial.flush();

esp_sleep_enable_ext0_wakeup(GPIO_NUM_13, 1);

delay(10000);

esp_deep_sleep_start();

```

Pokud je tedy systém spuštěný, následuje konfigurace kamery, nastavení kvality snímku a následné zachycení snímku. Z dalších částí bych rád popsal vytvoření názvu obrázku. Pro ten je potřeba, aby byl unikátní, získat čas vyfocení (rok, hodinu, minutu a sekundu). Ten nahrajeme z NTP serveru do konstanty a obrázek uložíme na microSD kartu.

```

configTime(gmtOffset_sec, daylightOffset_sec, ntpServer);

struct tm timeinfo;

if(!getLocalTime(&timeinfo)){

    Serial.println("Failed to obtain time");

    return;

}

char get_cas[11];

strftime(get_cas,11, "%Y%H%M%S", &timeinfo);

const char* const_cas=get_cas;

String cas= const_cas;

String path = "/public_html/img/IMG" + cas + ".jpg";

File file = fs.open(path.c_str(), FILE_WRITE);

```

Dále pomocí ftp klienta odešleme snímek na server. Cestu ke složce již máme uloženou v proměnné path.

```

ftp.OpenConnection();

const char *f_name = path.c_str();

ftp.ChangeWorkDir("/public_html/img/");

```

```

ftp.InitFile("Type I");

ftp.NewFile( f_name );

ftp.WriteData(fb->buf, fb->len);

ftp.CloseFile();

ftp.CloseConnection();

```

Nyní již zbývá poslat informace o snímku webové aplikaci, to zajišťuje opět http klient pomocí metody POST. Odesílaná data jsou lokace, typ čidla, cesta ke snímku a klíč sloužící k ověření, zda se ke stránce připojuje mikrokontroler či nějaký cizinec bez klíče.

```

HTTPClient http

http.begin(serverName);

http.addHeader("Content-Type", "application/x-www-form-
urlencoded");

String httpRequestData = "location=" + location + "&key=" +
key + "&type=" + type + "&url=" + path;

Serial.print("httpRequestData: ");

Serial.println(httpRequestData);

int httpResponseCode = http.POST(httpRequestData);

if (httpResponseCode>0) {

    Serial.print("HTTP Response code: ");

    Serial.println(httpResponseCode);

    String payload = http.getString();

    Serial.println(payload);

}else {

    Serial.print("Error code: ");

```

```

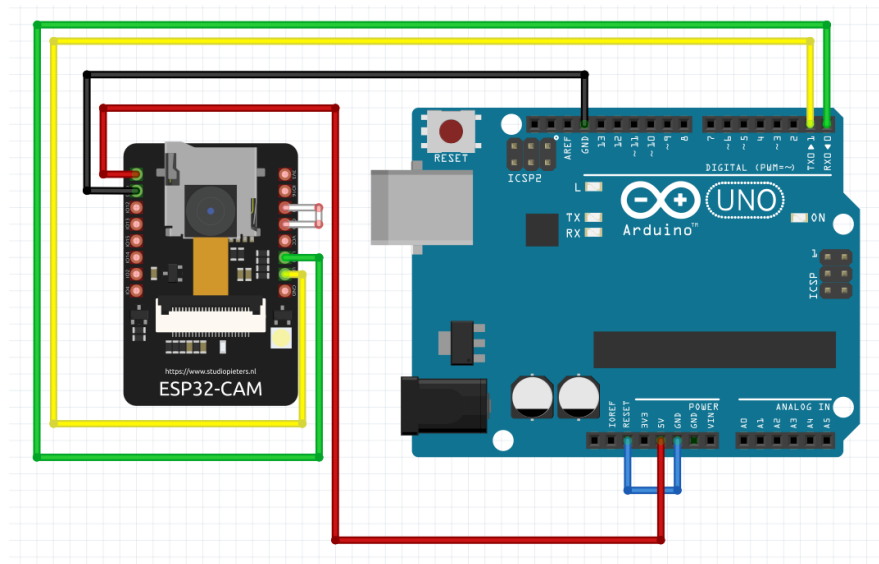
Serial.println(httpResponseCode);

}

http.end();

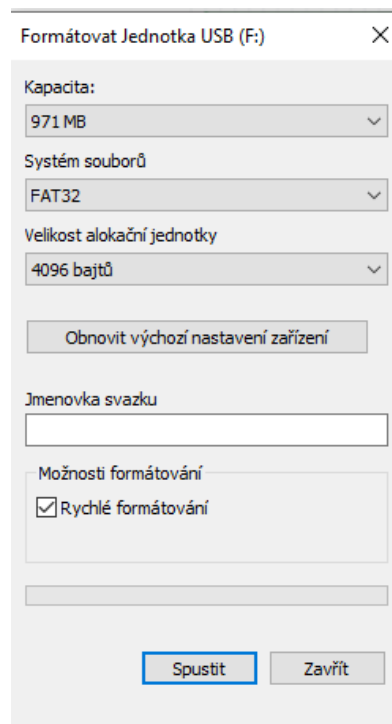
```

Pro nahrání kódu jsem využil arduino uno a následující zapojení.



Obr. 10: Schéma zapojení pro nahrání programu

Dalším nezbytným krokem je formátování microSD karty na systém souborů FAT32.



Obr. 11: Formátování microSD

## 2.3 Webová aplikace

Pro webovou aplikaci využívám hosting zdarma od 000webhost<sup>6</sup>. Zde mám doménu druhého řádu <https://home363.000webhostapp.com/>.

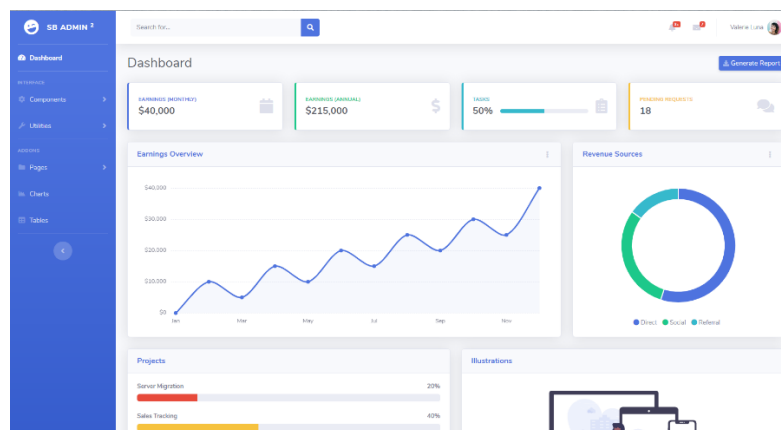
### 2.3.1 Šablona

Zvolil jsem možnost Bootstrap<sup>7</sup> šablony, tu jsem upravil a ponechal zejména základní vzhled a změnil jsem rozložení menu, upravil možnost odhlášení, přidal vypínání a zapínání aplikace, odstranil přebytečné vlastnosti jako bylo zobrazení uživatele s fotkou v pravém horním rohu a celkově šablonu přizpůsobil potřebám aplikace.

---


<sup>6</sup> 000webhost [online]. [cit. 2020-04-17]. Dostupné z: <https://www.000webhost.com/>

<sup>7</sup> SB Admin 2. *Start Bootstrap* [online]. [cit. 2020-04-17]. Dostupné z: <https://startbootstrap.com/themes/sb-admin-2/>



Obr. 12: Původní uživatelské rozhraní

The screenshot shows the BE SECURED user interface. The left sidebar contains navigation links: Způsob, Vyhledat, Ochrana, Přihlášení, Uživatelé, and Správa uživatelů. The main content area is titled 'Uživatel r\_kadera' and displays a table of filtered data.

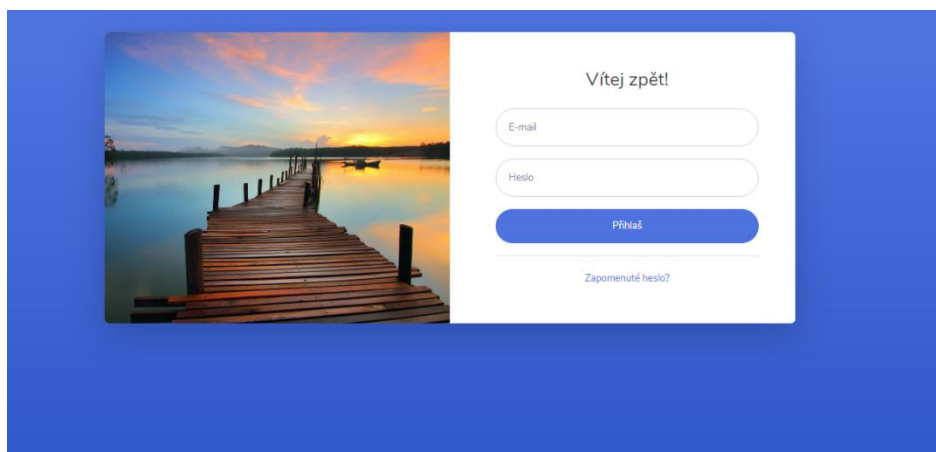
Čas	Místo	Typ senzoru	Foto	Smazat
02.05.2020 - 17:01	Chodba	camera		<a href="#">Smazat</a>
Čas	Místo	Typ senzoru	Foto	

Obr. 13: Upravené uživatelské rozhraní

The screenshot shows a login page with a blue background. On the left is a large image of a pug dog wearing a denim jacket. On the right is a white login form with the following elements:

- Welcome Back!** header
- Enter Email Address...
- Password
- ☐ Remember Me
- 
- 
- 
- [Forgot Password?](#)
- [Create an Account!](#)

Obr. 14: Původní přihlášení



Obr. 15: Upravené přihlášení

Šablona samotného uživatelského rozhraní je pak uložena zvlášť v souboru kostra.php. Tento soubor je pak načten do každé stránky, kde je tato šablona potřeba. To poskytuje výhodu, chceme-li upravit šablonu, stačí tak učinit pouze v tomto souboru.

## 2.3.2 Databáze

Databáze se skládá ze tří tabulek: system – kde je uložen název systému a jeho stav (on, off)

sensor – zde se nahrávají záznamy z čidel

users – ta obsahuje údaje o uživateli. Všechny tabulky obsahují id, které je unikátní.

	id	name	on_off
<input type="checkbox"/> Upravit  Kopírovat  Odstranit	1	Home	0

Obr. 16: Tabulka system

	id	location	type	time	url
<input type="checkbox"/> Upravit  Kopírovat  Odstranit	243	Chodba	camera	02.05.2020 - 17:01	img/IMG2020170123.jpg

Obr. 17: Tabulka sensor

	id	us_name	us_password	us_mail	send_mail	admin
<input type="checkbox"/> Upravit  Kopírovat  Odstranit	1	r_kadera	77cac6c6fbb3a38d8f6d8ff050934003	kaverave01@gmail.com	1	1
<input type="checkbox"/> Upravit  Kopírovat  Odstranit	25	okok	854bdefcf36b8c50d154d2f9d9f84f0c	radim.kadera@gnj.cz	0	0

Obr. 18: Tabulka users



### 2.3.3 Zpracování dat z mikrokontroleru

První moment, kdy dochází k propojení mikrokontroleru a aplikace, nastává ve chvíli kdy zjišťujeme zda-li je samotný bezpečnostní systém spuštěný. Při tomto požadavku se provede kód souboru on\_of.php. Ten se připojí k MySQL databázi, samotné připojení je v souboru db\_connect.php. Následuje kontrola klíče a zda přijatý dotaz obsahuje nějaká data poslaná metodou POST. Pokud je vše splněno, aplikace otestuje data zda neobsahují cizí kód a případně je upraví. Dále proměnná type uvádí, zda se jedná opravdu o kontrolu stavu aplikace, pokud ano (type=on\_off), aplikace zjistí, zda se v databázi v tabulce system rovná 0 (on), či 1. Tuto informaci poté vrací mikrokontroleru.

```
<?php

require_once("db_connect.php");

$key="tPmAT5Ab3j7F9";

if ($_SERVER["REQUEST_METHOD"] == "POST") {

    if($key==test_input($_POST["key"])){

        $type=test_input($_POST["type"]);

        if($type="on_off"){

            $sql="SELECT on_off FROM system WHERE id=1";

            $result=$conn->query($sql);

            $result=$result->fetch_assoc();

            if ($result["on_off"]==1){

                echo"on";

            }else{

                echo"off";

            }

        }

    }

}
```

```

        }

        }else{

            echo"error";

        }

    }else{

        echo "wrong key";

    }

}

else {

    echo "No data POSTed with HTTP POST.";

}

function test_input($data) {

    $data = trim($data);

    $data = stripslashes($data);

    $data = htmlspecialchars($data);

    return $data;

}

?>

```

V dalším propojení už aplikace přijímá data zachycená čidlem. K zpracování slouží soubor post-esp-data.php. Ten opět ověřuje klíč a data poslaná metodou POST. Následně pomocí funkce date() získávám datum a čas. Dále se tyto data uloží do databáze do tabulky sensor. Pokud vše proběhne úspěšně, tak aplikace využije knihovnu PHPMailer a pomocí ní pošle uživatelům, kteří mají povolené posílání emailů (v tabulce users), email. Tyto emailové adresy program načítá pomocí sql dotazu z tabulky users.

```

<?php

require 'mailer/src/Exception.php';

require 'mailer/src/PHPMailer.php';

require 'mailer/src/SMTP.php';

use PHPMailer\PHPMailer\PHPMailer;

use PHPMailer\PHPMailer\SMTP;

use PHPMailer\PHPMailer\Exception;

require_once("db_connect.php");


date_default_timezone_set("Europe/Paris");

$key="tPmAT5Ab3j7F9";


if ($_SERVER["REQUEST_METHOD"] == "POST") {

    if($key==test_input($_POST["key"])){

        $location = test_input($_POST["location"]);

        $type=test_input($_POST["type"]);

        $url=test_input($_POST["url"]);

        $url=str_replace("/public_html/", "", $url);

        $time=date("d.m.Y - G:i");
    }
}

```

```
$sql = "INSERT INTO sensor (location, type, url,
time) VALUES ('" . $location . "', '" . $type . "', '" .
$url . "', '" . $time . "' )";
```

```
if ($conn->query($sql) === TRUE) {
    echo "New record created successfully";

    $mail = new PHPMailer;

    $mail->isSMTP();

    $mail->Host = "smtp.gmail.com";

    $mail->SMTPAuth = true;

    $mail->Username = $mailuser;

    $mail->Password = $mailpassword;

    $mail->SMTPSecure = "tls";

    $mail->Port = 587;

    $mail->CharSet = "utf-8";

    $mail->isHTML();

    $mail->setFrom("espcam363@gmail.com", "Be
secured");
```

```
$sql2 = "SELECT us_mail FROM users WHERE
send_mail=1";
```

```
$result = $conn->query($sql2);
```

```
if ($result->num_rows > 0) {
```

```

        while($adress = $result->fetch_assoc())
        {
            $mail->
>addAddress($adress["us_mail"]);
        }
    } else {
        echo "0 results";
    }

    $mail->addAttachment($url);
    $mail->Subject = "Upozornění";
    $mail->Body = "Zachycen pohyb v místě : " .
$location;

    if($mail->send()){
        if($p>1){
            echo"Emaily byly odeslány($p)";
        }else{
            echo "Email byl odeslán";
        }
    }else{
        echo $mail->ErrorInfo;
    }
}
else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}

```

```

    }

    $conn->close();

} else {

    echo "Wrong key";

}

}

else {

    echo "No data POSTed with HTTP POST.";

}

function test_input($data) {

    $data = trim($data);

    $data = stripslashes($data);

    $data = htmlspecialchars($data);

    return $data;

}

?>

```

### 2.3.4 Přihlášení

Součástí přihlášení jsou dva soubory – index.php a prihlas.php. Index.php obsahuje formulář pro uživatele, zde vloží svůj email a heslo. Dále je zde odkaz pro obnovení hesla.

Po odeslání formuláře se data pomocí metody POST pošlou do souboru prihlas.php ke zpracování. Zde se zkontrolují vstupy dat, uloží se do proměnné a ověří se, zda v databázi existuje uživatel s použitým emailem a heslem. Heslo je šifrováno pomocí md5(). Pokud uživatelské jméno a heslo souhlasí s databází, je uživatel přesměrován do uživatelské

sekce a jeho údaje jsou uloženy do session – způsob ukládání do proměnné mezi více webovými stránkami. Při každém pokusu pro připojení je poté ověřeno, zda je session nastavena, pokud ne, není uživatel přihlášen a je přesměrován zpět na index.php. Na začátku je spuštěna funkce ob\_start(), ta zajišťuje správné fungování přesměrování, neboť pokud je zavolána funkce header() tak znemožňuje stránce vypisovat informace, ty by totiž mohli přerušit přesměrování.

```
<?php

session_start();

ob_start();

require_once("db_connect.php");

$log_mail=htmlspecialchars($_POST["log_mail"]);

$log_password=md5(htmlspecialchars($_POST["log_password"]))

;

$sql="SELECT * FROM users where us_mail ='$log_mail' and
us_password = '$log_password'";

$result = $conn->query($sql);

if ($result->num_rows == 1) {
    while ($user= $result->fetch_assoc()){
        $_SESSION['us_id']= $user ["id"];
        $_SESSION['us_name']= $user ["us_name"];
        $_SESSION['us_mail']= $user ["us_mail"];
        $_SESSION['us_password']= $user ["us_password"];
        $_SESSION['admin']= $user ["admin"];
```

```

    }

    header ("Location: user.php");

}

else{

    $_SESSION ["chyba"]= 1;

    header ("Location: index.php");

}

ob_end_flush();

?>

```

### 2.3.5 Zapomenuté heslo

Pokud uživatel zapomene heslo, stačí pod přihlášením zvolit odkaz k tomu určený. Bude přesměrován na stránku sloužící k jeho obnovení forgot-password.php. Ve formuláři pak stačí zadat uživatelský email a odeslat jej. Zpracování poté provede forgot-password-zpracuj.php. Ten opět ověří vstup dat a poté v databázi vyhledá uživatele a následně mu zašle email s novým náhodně vygenerovaným heslem, současně toto heslo uloží do databáze. Pokud nastane chyba, tak je uložena do session a vypsána uživateli.

```

<?php

require 'mailer/src/Exception.php';

require 'mailer/src/PHPMailer.php';

require 'mailer/src/SMTP.php';


use PHPMailer\PHPMailer\PHPMailer;

use PHPMailer\PHPMailer\SMTP;

use PHPMailer\PHPMailer\Exception;

session_start();

```



```

ob_start();

require_once("db_connect.php");

$sus_mail=htmlspecialchars($_POST["mail"]);

$sql="SELECT * FROM users WHERE us_mail='$sus_mail'";

$result=$conn->query($sql);

if ($result->num_rows == 1) {

    $new_password=rand();

    $new_password_hash=md5($new_password);

    $sql="UPDATE users SET us_password='$new_password_hash'

    WHERE us_mail='$sus_mail'";

    if ($conn->query($sql) === TRUE) {

        $mail = new PHPMailer;

        $mail->isSMTP();

        $mail->Host = "smtp.gmail.com";

        $mail->SMTPAuth = true;

        $mail->Username = $mailuser;

        $mail->Password = $mailpassword;

        $mail->SMTPSecure = "tls";

        $mail->Port = 587;

        $mail->CharSet = "utf-8";

        $mail->isHTML();

        $mail->setFrom($mailuser, "Be secured");

        $mail->addAddress($sus_mail);
    }
}

```

```

$mail->Subject = "Zapomenuté heslo";

$mail->Body = "Vaše nové heslo: " . $new_password;

if($mail->send()){

    $_SESSION['error']="Heslo zasláno";

    header("Location: forgot-password.php");

}else{

    $_SESSION['error']="Problém s odesláním emailu";

    header("Location: forgot-password.php");

}

F

}else{

    $_SESSION['error']="Chyba";

    header("Location: forgot-password.php");

}

}else{

    $_SESSION['error']="E-mail nenalezen";

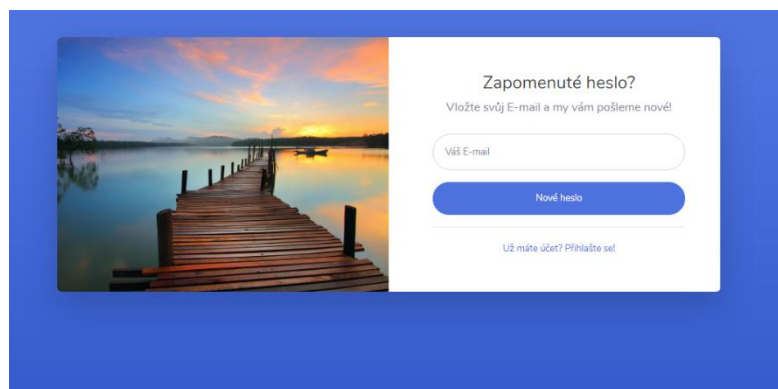
    header("Location: forgot-password.php");

}

ob_end_flush();

?>

```



Obr. 19: Zapomenuté heslo

### 2.3.6 Přehled

Přehled slouží k zobrazení záznamů, které zachytily čidla, soubor je pak pojmenován user.php. Tyto záznamy jsou načteny z databáze do tabulky, jelikož pracuji s kamerou, tak jsou doplněny o miniaturu obrázku, ten lze díky Lightbox knihovně rozkliknout a zobrazit. Další funkcí je odstranění záznamu, pokud ho uživatel chce odstranit, otevře se mu okno žádající ho o potvrzení tohoto kroku.

```
<?php

$sql = "SELECT * FROM sensor ORDER BY id DESC";

$result = $conn->query($sql);

if ($result->num_rows > 0) {

    while($row = $result->fetch_assoc()) {

        echo "<tr><td>" .

        $row["time"] . "</td><td>" .

        $row["location"] . "</td> <td> " .

        $row["type"] . "</td><td>

        <a class='ikona-link' href='" . $row["url"] . "' data-
        lightbox='roadtrip'><img class='ikona' src='" .

        $row["url"] . "' ></a></td>";
```

```

<td><a href="delete.php?url=<?php echo
$row["url"];?>&id=<?php echo $row["id"]; ?>"
onclick="return confirm('Odstranit
záznam?')">Smazat</a></td></tr>

<?php
    }
} else {
    echo "0 results";
}
$conn->close();
?>

```

Po potvrzení smazání je proveden kód souboru delete.php. Ten pomocí metody GET přijímá id záznamu a url obrázku, následně z databáze vymaže záznam a z ftp serveru obrázek.

```

<?php
session_start();
ob_start();
if (isset($_SESSION['us_name']) AND
isset($_SESSION['us_password'])) {

    require_once("db_connect.php");

    $id=htmlspecialchars($_GET["id"]);
    $url=htmlspecialchars($_GET["url"]);

    $sql= "DELETE FROM sensor WHERE id=$id";

```

```

        if ($conn->query($sql) === TRUE) {

            $_SESSION['error']="Smazáno";

            unlink($url);

            header("Location: user.php");

        } else {

            $_SESSION['error']="Chyba";

            header("Location: user.php");

        }

    }else{

        header ("Location: index.php");

    }

    ob_end_flush();

?>

```

### 2.3.7 Změna hesla a emailu

Změna emailu a hesla probíhá podobně, jsou k ní vyčleněné stránky `change_email.php` a `change_pswd.php`, data jsou poté zpracovávána v `change_email_zpracuj.php` a v `change_pswd_zpracuj.php`.

Ke změně emailu je potřeba zadat starou adresu, dvakrát novou a heslo. Tyto údaje poté aplikace porovná, pokud je vše v pořádku tak dojde ke změně a zaslání email s oznámením.

U změny hesla je potřeba staré heslo, pro potvrzení dvakrát nové a emailová adresa. Aplikace opět data zpracuje, zapíše do databáze a oznámení zašle uživateli emailem.

Kód už je velmi podobný s předešlými, proto ho nevkládám. Místo toho vkládám kód ukazující výpis případné chyby, ten je využit v celé aplikaci u většiny aplikací. Každá chyba je uložena do session s názvem error, jak už jsem se v textu zmínil dříve. Tato session, pokud existuje, je vypisována nad formulářem a ihned poté smazána, což zamezuje zobrazování chyb nesouvisejících z daným formulářem.

```
<?php

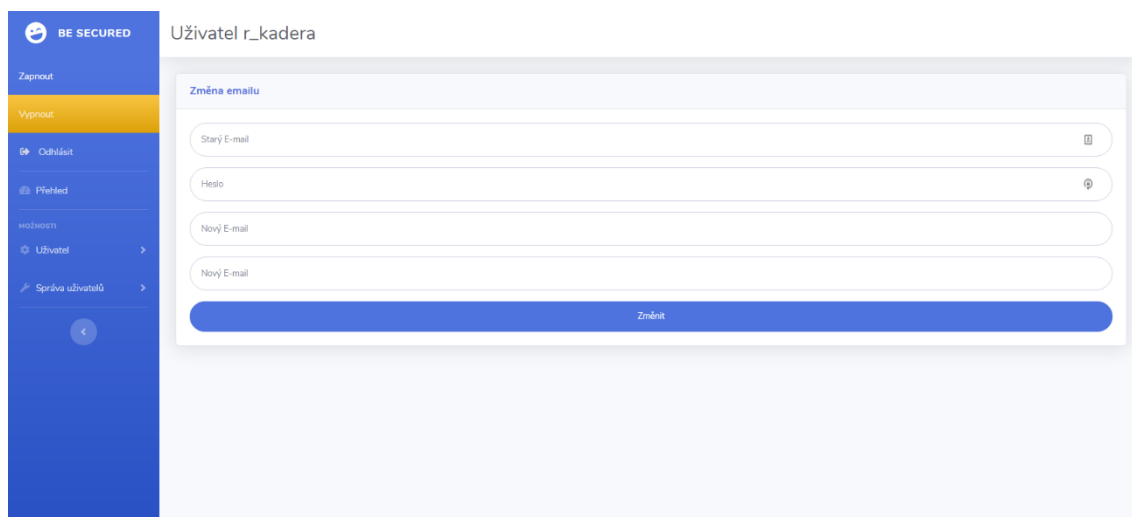
if(isset($_SESSION['error'])) {

    echo $_SESSION['error'];

    unset($_SESSION['error']);

}

?>
```



The screenshot shows a web application interface. On the left is a blue sidebar with a logo and the text "BE SECURED". Below the logo are several menu items: "Zapnout", "Vypnout", "Odházet", "Přihled", "možnosti", "Uživatel", and "Správa uživatelů". The main content area is titled "Uživatel r\_kadera" and contains a form titled "Změna emailu". The form has four input fields: "Starý E-mail", "Heslo", "Nový E-mail", and another "Nový E-mail". At the bottom of the form is a blue button labeled "Změnit".

Obr. 20: Změna emailu

Obr. 21: Změna emailu

### 2.3.8 Všichni uživatelé

V aplikaci jsou uživatelé děleni na administrátory a uživatele. K zobrazení všech uživatelů a vytvoření nového uživatele je potřeba administrátorský účet. Pokud uživatel není administrátor, není možné se do této sekce dostat, opět je tomu zabráněno pomocí kontroly session.

Administrátor má pak možnost zobrazit všechny uživatele a následně měnit jejich oprávnění, zasílání emailů s upozorněními o nových záznamech z čidla, či uživatele ze systému smazat.

Výpis uživatelů funguje obdobně jako u výpisu záznamu čidel. Jen dalšími možnostmi k smazání je také změna přijímání mailů a oprávnění. O to se stará soubor `change_user.php`, který přijme z odkazu (metoda GET), zda se jedná o změnu mailování, nebo oprávnění a tu poté provede, viz příklad změny oprávnění.

```
if ($change=="admin") {
    $admin=$from==1?0:1;;
    $sql="UPDATE users SET admin='$admin' WHERE
id='$id'";
    if ($conn->query($sql) === TRUE) {
        $_SESSION['error']="Změněno";
```

```

        header("Location: all_users.php");

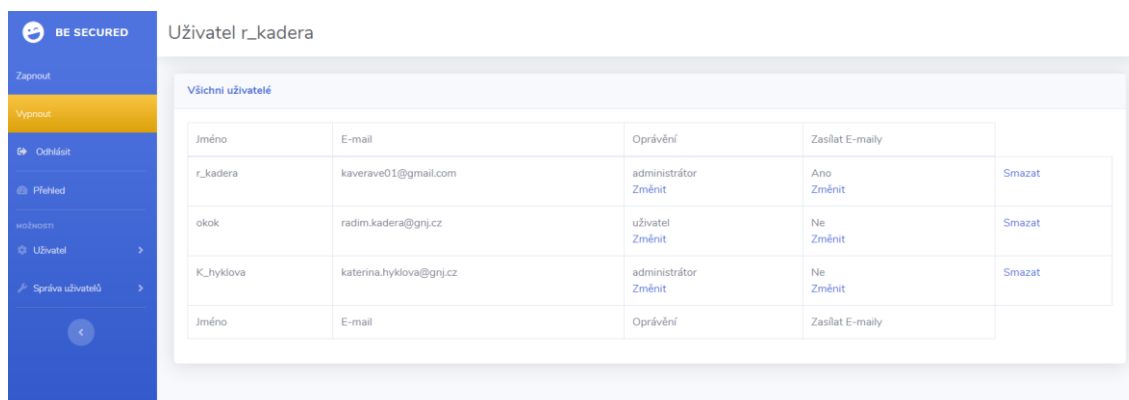
    }else{

        $_SESSION['error']="Chyba";

        header("Location: all_users.php");

    }

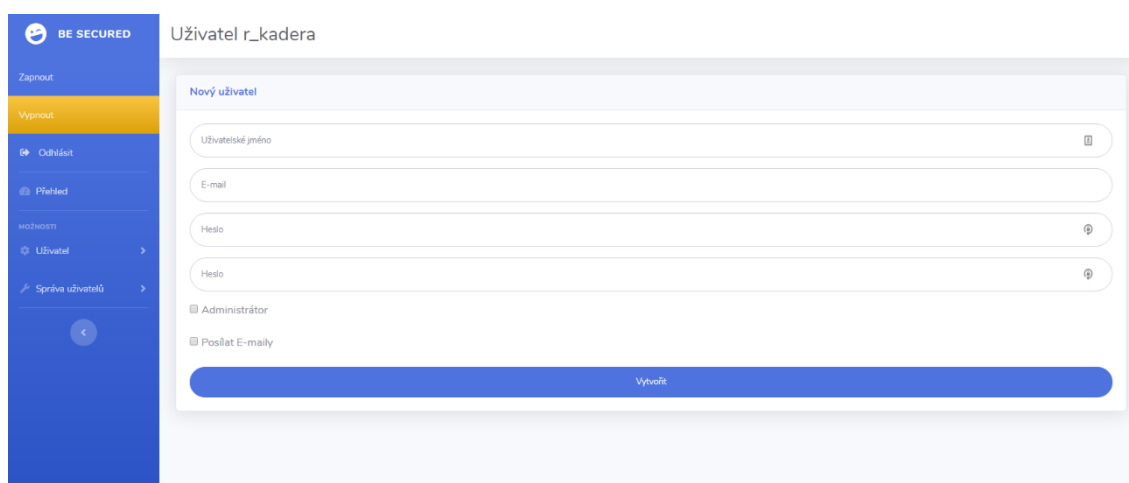
```



Obr. 22: Všichni uživatelé

### 2.3.9 Nový uživatel

K vytvoření nového uživatele je také potřeba být administrátorem. Po vyplnění údajů nového uživatele – uživatelské jméno, email, heslo, oprávnění, zasílání emailů, jsou data zpracovány v souboru new\_user\_zpracuj.php. Zde jsou data zkontrolována (zda mají délku větší než 6 znaků, zda se hesla shodují, zda uživatel již není zaregistrovaný), následně uložena do databáze a novému uživateli je zaslán email s přihlašovacími údaji.



Obr. 23: Nový uživatel



### 2.3.10 Zapnutí a vypnutí systému

O tento důležitý krok se stará soubor turn\_on\_off.php, který je spuštěn vždy po kliknutí na odkaz zapnout, či vypnout. Ten přijme z adresy (metodou GET) informaci, zda má aplikaci zapnout, nebo vypnout a tu následně uloží do databáze a do session. V kostra.php se v menu podle session označí, zda je aplikace zapnutá či vypnutá.

```
<?php

session_start();

ob_start();

if (isset($_SESSION['us_name']) AND
    isset($_SESSION['us_password'])) {

    require_once("db_connect.php");

    $change=htmlspecialchars($_GET["change"]);

    $sql="UPDATE system SET on_off='$change' WHERE id=1";

    if ($conn->query($sql) === TRUE) {

        $_SESSION['on_off']=$change;

        header("Location: user.php");

    } else {

        header("Location: user.php");

    }

} else {

    header ("Location: index.php");

}

ob_end_flush();

?>
```

### 2.3.11 Odhlášení

Odhlášení probíhá odstraněním všech vzniklých session v logout.php a následným přesměrováním na index.php.

```
<?php

session_start();

unset($_SESSION['us_id']);

unset($_SESSION['us_name']);

unset($_SESSION['us_password']);

unset($_SESSION['admin']);

unset($_SESSION['error']);

unset($_SESSION['us_mail']);

unset($_SESSION['on_off']);

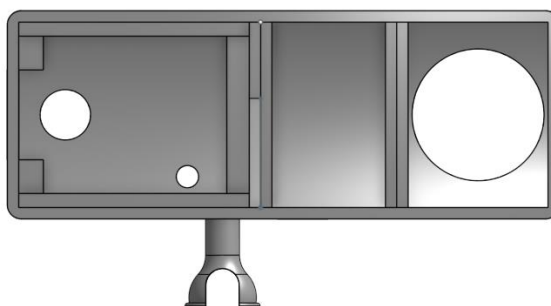
header ("Location: index.php");

?>
```

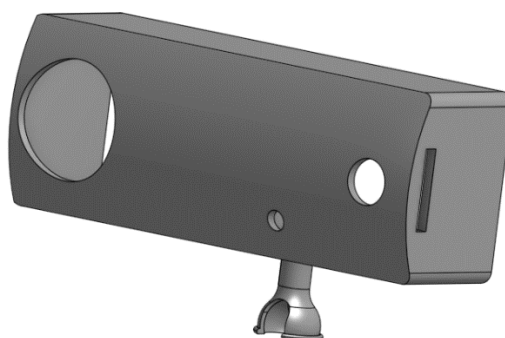
## 2.4 3D tisk

Pro samotnou kameru jsem vytvořil stojan a krabičku v online programu Onshape. Samotný výrobek se pak skládá ze čtyř částí – krabičky na komponenty, krytky, stojanu a kroužku na uchycení. V krabičce se nachází příčka pro zvládnutí přemostění při tisku

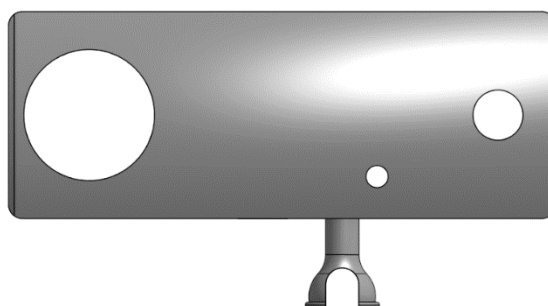
V krabičce je pak vytvořen otvor pro kameru, PIR čidlo, napájení, microSD kartu a diodu.



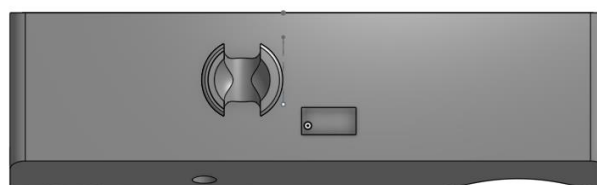
**Obr. 24: Krabička zezadu**



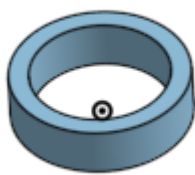
**Obr. 25: Krabička zboku**



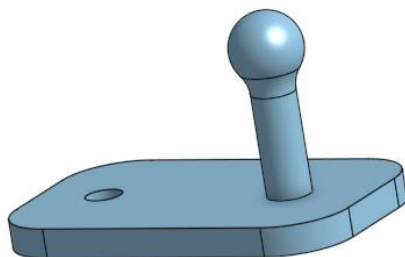
**Obr. 26: Krabička zepředu**



**Obr. 27: Krabička zespodu**



**Obr. 28: Kroužek**



**Obr. 29: Stojan**

Po výtisku je potřeba vypilovat otvor pro napájení, ten není možné kvůli přemostění vytisknout rovnou. Tisk proběhl na tiskárně pana Ondřeje Kopeckého.



**Obr. 30: Finální výrobek**

## Závěr

Ve své maturitní práci jsem se pokusil vytvořit aplikaci, která je schopná zabezpečit rodinný dům a propojit množství čidel. K dalšímu připojení čidla stačí jen upravit program pro ESP mikrokontroler. V práci jsem se potýkal s mnoha úskalími a jedním, z největších byl problém s http klientem, ten byl zapříčiněn špatným web hostingem a byl jsem nucen ho změnit. Před tímto zjištěním jsem prověřoval několik dnů všechny možnosti, kde může být chyba a čerpal z mnoha diskusních fór a stránek pojednávajících o tomto tématu.

Další úskalí nastalo například při 3D tisku, kde byl problém s dlouhým přemostěním, ten jsem vyřešil pomocnou příčkou. S tvorbou aplikace mi značně pomohlo absolvování kurzu tvorby webových stránek.

V projektu se nabízí několik možných vylepšení, to už jak o navýšení počtu čidel, nebo připojení dalšího systému (například zabezpečení chaty), databáze je na tento krok z části připravena v tabulce system, dalším krokem by bylo upravení tabulky sensor, kde by se záznamům přidělilo id systému, ke kterému spadají. Poté by stačilo v aplikaci rozdělit výpis záznamů na dva a přidat vypínání druhého systému. Dalším výrazným vylepšením by byl placený hosting a email klient. Nyní je počet emailů odeslaných za den limitován a není možné v nich posílat odkazy.

## Seznam obrázků

Obr. 1: Vývojová deska ESP8266.....	3
Obr. 2: Vývojová deska ESP32.....	3
Obr. 3: ESP32-CAM.....	4
Obr. 4: PIR čidlo .....	4
Obr. 5: Odpor 1k $\Omega$ .....	5
Obr. 6: Pájivé pole .....	5
Obr. 7: Adaptér microUSB na DIP .....	5
Obr. 8: Schéma zapojení .....	10
Obr. 9: Zapojení .....	10
Obr. 10: Schéma zapojení pro nahrání programu .....	15
Obr. 11: Formátování microSD .....	16
Obr. 12: Původní uživatelské rozhraní.....	17
Obr. 13: Upravené uživatelské rozhraní .....	17
Obr. 14: Původní přihlášení .....	17
Obr. 15: Upravené přihlášení .....	18
Obr. 16: Tabulka system .....	18
Obr. 17: Tabulka sensor .....	18
Obr. 18: Tabulka users .....	18
Obr. 19: Zapomenuté heslo .....	29
Obr. 20: Změna emailu .....	32
Obr. 21: Změna emailu .....	33
Obr. 22: Všichni uživatelé .....	34
Obr. 23: Nový uživatel.....	34
Obr. 24: Krabíčka zezadu.....	37
Obr. 25: Krabíčka zboku.....	37
Obr. 26: Krabíčka zepředu .....	37
Obr. 27: Krabíčka zespodu.....	37
Obr. 28: Kroužek.....	38
Obr. 29: Stojan .....	38
Obr. 30: Finální výrobek.....	38

## Seznam použitých zdrojů

M., Luboš. Arduino návody [online]. 2019 [cit. 2020-04-17]. Dostupné z:  
<https://navody.arduino-shop.cz/navody-k-produktum/vyvojova-deska-esp32-cam.html>

ESP32-CAM Troubleshooting Guide: Most Common Problems Fixed. RANDOM NERD TUTORIAL [online]. [cit. 2020-04-15]. Dostupné z:  
<https://randomnerdtutorials.com/esp32-cam-troubleshooting-guide/>

Build an ESP8266 Web Server – Code and Schematics (NodeMCU). RANDOM NERD TUTORIAL [online]. [cit. 2020-04-15]. Dostupné z:  
<https://randomnerdtutorials.com/esp8266-web-server/> Začátek formuláře

Esp8266 Arduino Send Data To Web Server Tutorial With Example. Microcontroller Projects [online]. [cit. 2020-04-15]. Dostupné z:  
<http://microcontrollerproject.com/esp8266-arduino-send-data-to-web-server-tutorial-with-example/>

ESP32/ESP8266 Insert Data into MySQL Database using PHP and Arduino IDE. RANDOM NERD TUTORIAL [online]. [cit. 2020-04-15]. Dostupné z:  
<https://randomnerdtutorials.com/esp32-esp8266-mysql-database-php/>

Motion Triggered Image Capture and Email. Instructables circuit [online]. [cit. 2020-04-15]. Dostupné z: <https://www.instructables.com/id/Motion-Triggered-Image-Capture-and-Email/>

Control ESP32 and ESP8266 GPIOs from Anywhere in the World. RANDOM NERD TUTORIAL [online]. [cit. 2020-04-15]. Dostupné z:  
<https://randomnerdtutorials.com/control-esp32-esp8266-gpios-from-anywhere/>

How To: HTTP GET and POST requests with ESP8266. ESP8266 Shop [online]. 2018 [cit. 2020-04-15]. Dostupné z: <https://esp8266-shop.com/blog/how-to-http-get-and-post-requests-with-esp8266/>

Esp32\_ftpclient. PlatformIO [online]. 2018 [cit. 2020-04-15]. Dostupné z:  
[https://platformio.org/lib/show/6543/esp32\\_ftpclient](https://platformio.org/lib/show/6543/esp32_ftpclient)

Getting Date and Time with ESP32 on Arduino IDE (NTP Client). RANDOM NERD TUTORIAL [online]. [cit. 2020-04-15]. Dostupné z:  
<https://randomnerdtutorials.com/esp32-ntp-client-date-time-arduino-ide/>

ESP32 NTP Client-Server: Get Date and Time (Arduino IDE). RANDOM NERD TUTORIAL [online]. [cit. 2020-04-15]. Dostupné z:  
<https://randomnerdtutorials.com/esp32-date-time-ntp-client-server-arduino/>

SQL WHERE Clause. W3schools.com [online]. [cit. 2020-04-15]. Dostupné z:  
[https://www.w3schools.com/sql/sql\\_where.asp](https://www.w3schools.com/sql/sql_where.asp)

ESP8266 Deep Sleep with Arduino IDE (NodeMCU). *RANDOM NERD TUTORIALS* [online]. [cit. 2020-04-17]. Dostupné z:  
<https://randomnerdtutorials.com/esp8266-deep-sleep-with-arduino-ide/>

ESP32 vs ESP8266 - klady a zápory. *Kuongshun* [online]. 2018 [cit. 2020-04-17]. Dostupné z: <http://cz.szks-kuongshun.com/info/esp32-vs-esp8266-pros-and-cons-29253681.html>

SANTOS, Sara. ESP32 vs ESP8266 – Pros and Cons. *Kuongshun* [online]. 2019 [cit. 2020-04-17]. Dostupné z: <https://makeradvisor.com/esp32-vs-esp8266/>

ESP32 – WROOM. *OctopusLAB* [online]. [cit. 2020-04-17]. Dostupné z:  
<https://www.octopuslab.cz/esp32/>

*HC-SR501 Passive Infrared (PIR) Motion Sensor* [online]. [cit. 2020-04-12]. Dostupné z: <https://www.epitran.it/ebayDrive/datasheet/44.pdf>

Pohybové čidlo HC-SR501. *Arduino návody* [online]. [cit. 2020-04-12]. Dostupné z: <https://navody.arduino-shop.cz/navody-k-produktum/pohybove-cidlo-hc-sr501.html>



PIR detektor: skvělý sluha, ale zlý pán. *Vyvoj.hw.cz* [online]. [cit. 2020-04-12].  
Dostupné z: <https://vyvoj.hw.cz/automatizace/pir-cidlo-skvely-sluha-ale-zly-pan.html>

*W3schools.com* [online]. [cit. 2020-04-12]. Dostupné z: <https://www.w3schools.com/>

Seaport During Daytime. *Pexels* [online]. [cit. 2020-04-10]. Dostupné z:  
<https://www.pexels.com/photo/seaport-during-daytime-132037/>

# Přílohy

## Příloha A – Posudek vedoucího maturitní práce

Gymnázium, Nový Jičín, příspěvková organizace  
Palackého 50, 741 01 Nový Jičín

jarní zkušební období roku 2020

### POSUDEK VEDOUcíHO MATURITNí PRÁCE S OBHAJOBOU Z ODBORNÝCH PŘEDMĚTŮ

Jméno žáka:

Název práce:

Jméno vedoucího práce:

Jméno oponenta práce:

Hodnocení zpracování tématu, struktury a obsahu (max 25 b) (minimálně 15 bodů v součtu od vedoucího i oponenta pro úspěšné vykonání MZ)		Udělené body (0-5)
1. Analýza tématu ( <i>přístup k problému, jeho analýza, vytýčení cíle</i> )		
2. Použití zvolené metody při práci, její vhodnost a náročnost		
3. Náročnost zpracování práce a přístup žáka k vypracování práce		
4. Způsob zpracování práce		
5. Naplnění vytýčeného cíle práce		
Hodnocení písemné části (max 10 b) (minimálně 6 bodů v součtu od vedoucího i oponenta pro úspěšné vykonání MZ)		Udělené body (0-5)
1. Formální úprava		
2. Jazyková úroveň		
Kritéria pro hodnocení obhajoby (max 15 b) (minimálně 9 bodů v součtu od vedoucího i oponenta pro úspěšné vykonání MZ)		Udělené body (0-5)
1. Příprava obhajoby (prezentace, podklady)		
2. Kvalita přednesu obhajoby		
3. Příprava a schopnost reakce na otázky		
<b>Body celkem</b>		
<b>Hodnocení</b>		

Otázky k obhajobě:

Žák/žákyně hodnotícím kritériím vyhověl(a) - nevyhověl(a).

Datum:

Podpis vedoucího maturitní práce