

ZÁVĚREČNÁ STUDIJNÍ PRÁCE

dokumentace

Kamerový systém ZoneMinder

Martin Šimkovský



Obor: 18-20-M/01 INFORMAČNÍ TECHNOLOGIE
se zaměřením na počítačové sítě a programování

Třída: IT4

Školní rok: 2019/2020

Poděkování

Chtěl bych poděkovat panu učiteli Ing. Petru Grussmannovi za cenné rady a věcné připomínky. Dále bych také rád poděkoval i mým spolužákům za případné poznámky pro vylepšení projektu a následnou pomoc.

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 31. 12. 2019

podpis autora práce

ANOTACE

Projekt se zabývá tvorbou kamerového systému, jehož součástí je také detekce pohybu pomocí externího čidla. Hardwarovou část systému tvoří modul Ai-Thinker ESP32-CAM, který snímá obraz a zprostředkovává komunikaci se serverem ZoneMinder. Dalším komponentem je PIR senzor, který dokáže detekovat vlny infračerveného spektra vyzařované okolními objekty. V případě detekce dojde k odeslání požadavku o započetí nahrávání na úložiště serveru. Programová část je vytvořena v jazyce Arduino, což je kombinace jazyků C a C++. Cílem této práce tedy bylo napodobit kamerové systémy, které jsou běžně dostupné na komerčním trhu, avšak za poměrně astronomické ceny. Prioritou bylo dosažení stejné funkčnosti za co nejmenší cenu.

KLÍČOVÁ SLOVA

kamerový systém; PIR senzor; ESP32-CAM; ZoneMinder; Wi-Fi; Arduino

OBSAH

ÚVOD.....	5
1 VYUŽITÉ TECHNOLOGIE	6
1.1 HARDWARE	6
1.1.1 Ai-Thinker ESP32-CAM	6
1.1.2 Pasivní infračervené čidlo HC-SR501	7
1.1.3 PL2303HX USB TTL převodník	7
1.1.4 Raspberry Pi 4 Model B 4GB	8
1.2 SOFTWARE	9
1.2.1 PlatformIO.....	9
1.2.2 ZoneMinder.....	9
1.2.3 Jazyk Arduino	9
2 PRINCIP FUNGOVÁNÍ SYSTÉMU	10
2.1 RTSP	10
3 ZPŮSOBY ŘEŠENÍ A POUŽITÉ POSTUPY	11
3.1 PŮVODNÍ VIZE O REALIZACI A PŘEHODNOCENÍ	11
3.2 NAHRÁVÁNÍ FIRMWAREU DO PAMĚTI ESP32	11
3.3 TECHNOLOGIE STREAMOVÁNÍ OBRAZU	11
3.4 ZONEMINDER.....	12
3.4.1 Instalace a uživatelé	12
3.4.2 Přidání kamery	12
3.5 EXTERNÍ TRIGGER	14
3.5.1 PIR senzor	14
3.5.2 Způsob oznámení o detekci pohybu.....	14
3.5.2.1. Telnet server.....	14
3.5.2.2. HTTP server	14
3.5.2.3. Telnet client.....	15
4 VÝSLEDKY ŘEŠENÍ.....	16
ZÁVĚR	17
SEZNAM POUŽITÝCH INFORMAČNÍCH ZDROJŮ	18

ÚVOD

Často jsem přemýšlel o tom, jak by se dal zabezpečit nějaký objekt proti potenciálnímu vniknutí nebo krádeži. Po dlouhé době jsem se nakonec rozhodl vytvořit první část bezpečnostního systému - část kamerovou. Primárním cílem bylo propojit alespoň jednu kameru se zabezpečenou webovou aplikací a umožnit tak online sledování. Uživatel je po zadání IP adresy do vyhledávacího okna prohlížeče vyzván k přihlášení. Dalším úkolem tohoto zařízení je vytvářet videozáznam právě v okamžiku, kdy dojde k detekci pohybu pomocí PIR senzoru a k jeho uložení na server. Po celou dobu provozu vývojové desky je také možné sledovat aktuální obraz kamery online. Zařízení tak lze využít k monitorování jakýchkoliv nemovitostí, kde je dostupná síť Wi-Fi.

Jednalo se o můj první projekt, a proto jsem při jeho zpracovávání vyzkoušel mnoho verzí programu. Nakonec jsem došel k závěru, že hardwarovou část udělám co nejjednodušší a bude ji tvořit vývojová deska Ai-Thinker ESP32-Cam společně s pasivním infračerveným čidlem LHI778, které s 2 trimery (slouží k nastavení citlivosti snímání a prodlevě sepnutí) tvoří modul HC-SR501. Nejsem příliš programátorský typ, tudíž jsem se ani nepokoušel bádát v nabídce možných programovacích jazyků, ve kterých by šlo vše napsat, ale raději jsem bez dlouhého rozmýšlení sáhl po jazyce Arduino, kombinaci jazyků C a C++. Pro projekt mi byla doporučena open-sourcová aplikace ZoneMinder, která je velice komplexní a umožňuje jednoduché přidávání a konfigurování nových zařízení, správu uživatelů, ale i samotného serveru. U ZoneMinderu se jednalo pouze o instalaci a konfiguraci serveru nejdříve na virtuálním stroji, později na Raspberry PI4.

V úvodu dokumentace jsou zachyceny veškeré poznatky, které jsou důležité pro sestavení a zprovoznění systému. Dále pak vysvětlují postup při řešení jednotlivých úkolů, hledání a řešení vzniklých problémů. Popisují způsob komunikace mezi ESP a serverem. Závěrem uvádím, jak vypadá současná podoba systému a také hodnotím odvedenou práci.

1 VYUŽITÉ TECHNOLOGIE

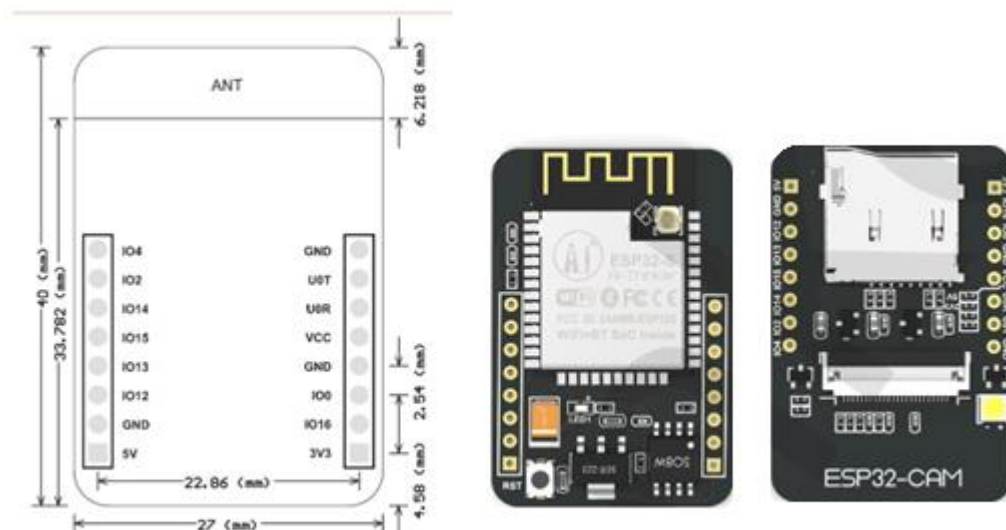
1.1 Hardware

1.1.1 Ai-Thinker ESP32-CAM

Základ projektu tvoří vývojová deska **Ai-Thinker ESP32-CAM**, která se skládá z hlavního modulu **ESP32-S** (z dílny společnosti Espressif Systems), jenž obstarává bezdrátovou komunikaci se serverem prostřednictvím Wi-Fi a samotné zpracování obrazových dat získaných z 2MPx snímače **OV2640**.

Parametry:

- napájení 5V
- 9 IO pinů
- frekvence procesoru 160MHz
- paměť SRAM o velikosti 520KB
- podpora senzorů OV2640 a OV7670
- možnost nahrání firmwaru pomocí protokolu OTA
- rozměry 40,5mm*27mm
- cena 120Kč



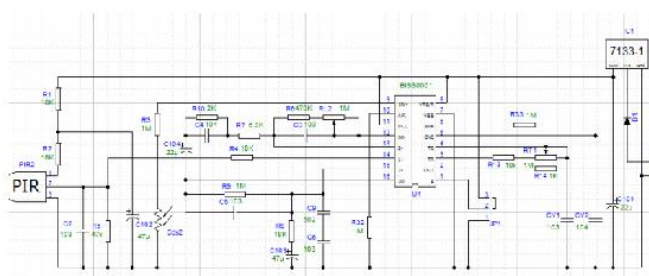
Obrázek č. 1 Ai-Thinker ESP32-CAM

1.1.2 Pasivní infračervené čidlo HC-SR501

Detekce pohybu funguje na pyroelektrickém principu. Senzor snímá své okolí pod úhlem $<120^\circ$. Pokud ve snímaném prostředí dojde ke větší změně teploty, pomocná elektronika na modulu pohybového čidla tuto akci zaznamená a oznámí ji uživateli změnou výstupního napětí. Modul HC-SR501 obsahuje také 2 trimry, kterými lze nastavit veličiny - citlivost snímání okolí a dobu sepnutí výstupního napětí. Nevýhodou tohoto modulu je blokování čidla po vypnutí senzoru na dobu 2,5 s z důvodu zamezení rušení senzoru indukční složkou.



Obrázek č. 2 HC-SR501



Obrázek č. 3 HC-SR501- schéma

1.1.3 PL2303HX USB TTL převodník

Jedná se o tzv. „programátor“ s vestavěným TTL COM PC-PL2303HX čipem, který slouží pro nahrávání firmwaru do desek, jež nemají pro sériovou komunikaci žádný typ USB zabudován. Z jedné strany se nachází standardní USB-A port, na straně druhé pak 4 pin konektory, konkrétně to jsou (GND, TXD, RXD, VCC).



Obrázek č. 4 PL2303HX USB TTL převodník



1.2 Software

1.2.1 PlatformIO

Jedná se o program, který kombinuje IDE (nástroj pro zápis zdrojového kódu), správce knihoven a nástroj pro překlad a nahrání výsledného firmwaru do paměti požadované desky. Největšími výhodami (oproti Arduino IDE) jsou - podpora téměř 700 MCU na 31 platformách, inteligentní doplňování kódu, správa knihoven, podpora GITu a spousta dalších zajímavých funkcí.

1.2.2 ZoneMinder

Jedná se o bezplatný a opensourcový software, vytvořený speciálně pro kamerové systémy. Byl vyvinut a přizpůsoben pro provoz pod systémy Linux a FreeBSD. Aplikaci lze využívat jak pro klasické kamery, propojené přes záznamové zařízení, tak pro USB či IP kamery.

Uživatel si může vybrat ze tří módů provozu aplikace:

- monitorování (bez nahrávání)
- nahrávání po detekci pohybu
- trvalé nahrávání

Aplikace nabízí širokou škálu nastavení zařízení, ale i sebe samu (správu uživatelů, nastavení přihlášení, upozornění, úložiště, atd.).

1.2.3 Jazyk Arduino

Program, který má za úkol ovládání ESP32-CAM, je napsán v jazyce Arduino. Je velice podobný jazyku C nebo C++. Jedná se o jazyk, který byl primárně vyvinut pro programování integrovaných obvodů.

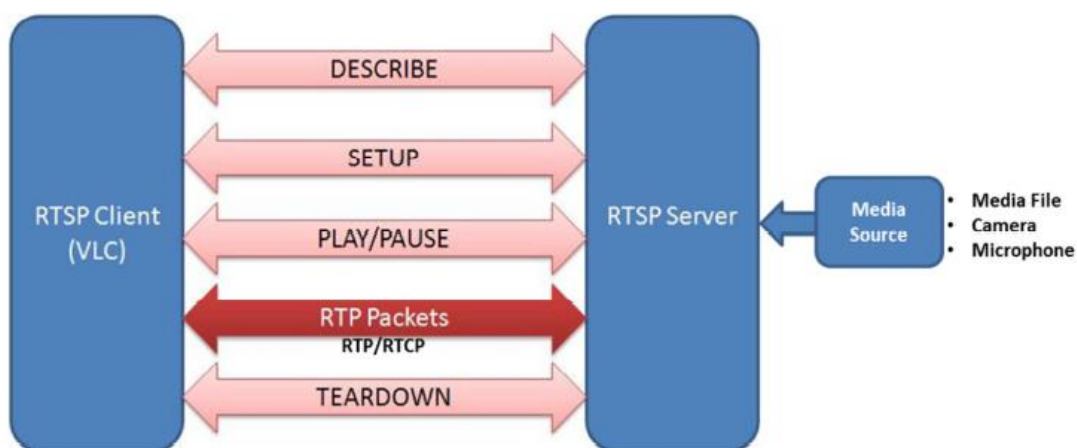
2 PRINCIP FUNGOVÁNÍ SYSTÉMU

Za předpokladu, že je současně zapnutý server ZoneMinder a kamera a zároveň jsou obě zařízení připojena ke stejné Wi-Fi síti, navážou mezi sebou spojení. Ai-Thinker ESP32-CAM začne pomocí svého snímače OV2640 zaznamenávat obraz. Pomocí RTS protokolu pak kontinuálně a bez ověřování úspěšnosti přenosu odesílá data. ZoneMinder je zpracovává a umožňuje je skrze svou webovou aplikaci zobrazovat. V momentu, kdy dojde k detekci pohybu externím PIR senzorem, ESP32 se připojí na ZoneMinder prostřednictvím telnetu a odešle požadavek o zahájení nahrávání. Pokud byl spuštěn alarm a zahájeno nahrávání, ihned po jeho dokončení lze vidět vytvořený event a video lze stáhnout nebo najít v úložišti serveru v adresáři:

```
/var/cache/zoneminder/events/název_kamery/datum/číslo_eventu.
```

2.1 RTSP

RTSP je zkratka pro Real Time Streaming Protocol. Byl vyvinut speciálně pro řízení doručování dat v reálném čase. Slouží k doručování obsahu formou datového proudu jednosměrným vysíláním (unicast). Jedná se o protokol, který má nějaký stav/směrnici, který určuje, o jaký druh požadavku se jedná. Směrnice můžou být například (OPTIONS, DESCRIBE, SETUP, PLAY, PAUSE, STOP, RECORD, SET_PARAMETER, ...).



Obrázek č. 6 RTSP komunikace

3 ZPŮSOBY ŘEŠENÍ A POUŽITÉ POSTUPY

3.1 Původní vize o realizaci a přehodnocení

Když jsem si zvolil okruh, o čem budu následující měsíce tvořit svou závěrečnou studijní práci, musel jsem se nejdříve rozhodnout, jak by zabezpečovací systém měl vlastně vypadat. Původní představa byla taková, že sestavím centrální jednotku tvořenou Raspberry Pi 4 s dotykovým displejem a k ní připojím několik kamer (výrobce Raspberry) a několik dalších přídavných čidel. Toto řešení mi však během konzultací vedoucí práce rozmluvil, protože se jednalo o finančně náročnou variantu, a doporučil mi vyzkoušet ESP32-CAM.

3.2 Nahrávání firmwaru do paměti ESP32

Za zmínku stojí také způsob flashování vývojové desky. Jelikož není osazena žádným USB konektorem, musí být k PC připojena TTL USB převodníkem pro sériovou komunikaci. Potřebná je také instalace příslušného ovladače, který však není oficiálně používán z důvodu jeho nefunkčnosti. Při každém cyklu nahrání nového programu je potřeba uvést desku do tzv. flashovacího módu zkratováním pinů GDN a IO 0.

3.3 Technologie streamování obrazu

Po rozhodnutí, že mou vývojovou deskou bude ESP32, jsem začal studovat, jakým protokolem lze bezdrátově přenášet obraz. Jako první jsem vyzkoušel HTTP JPG stream. Přenos fungoval dle mých představ a já se již začal zamýšlet nad webovou aplikací, která umožňuje správu jednotlivých zařízení a čidel, vytvoření logického systému pro vyhodnocování spouštění alarmu, započetí nahrávání a upozornění uživatele. V tuto dobu přišel další zásah vedoucího práce. Doporučení znělo jasně: přejít od protokolu HTTP na RTSP, nedělat si zbytečnou práci vytvářením vlastní webové aplikace, ale raději využít již předpřipravenou aplikaci pro tyto účely - ZoneMinder.

Ukázka kódu- http JPG stream

```
1. void handle_jpg_stream(void)
2. {
3.     WiFiClient client = server.client();
4.     String response = "HTTP/1.1 200 OK\r\n";
5.     response += "Content-Type: multipart/x-mixed-
   replace; boundary=frame\r\n\r\n";
6.     server.setContent(response);
7.
8.     while (1)
9.     {
10.        cam.run();
11.        if (!client.connected())
12.            break;
13.        response = "--frame\r\n";
14.        response += "Content-Type: image/jpeg\r\n\r\n";
15.        server.setContent(response);
16.
17.        client.write((char *)cam.getfb(), cam.getSize());
18.        server.setContent("\r\n");
19.        if (!client.connected())
20.            break;
21.    }
```

3.4 ZoneMinder

3.4.1 Instalace a uživatelé

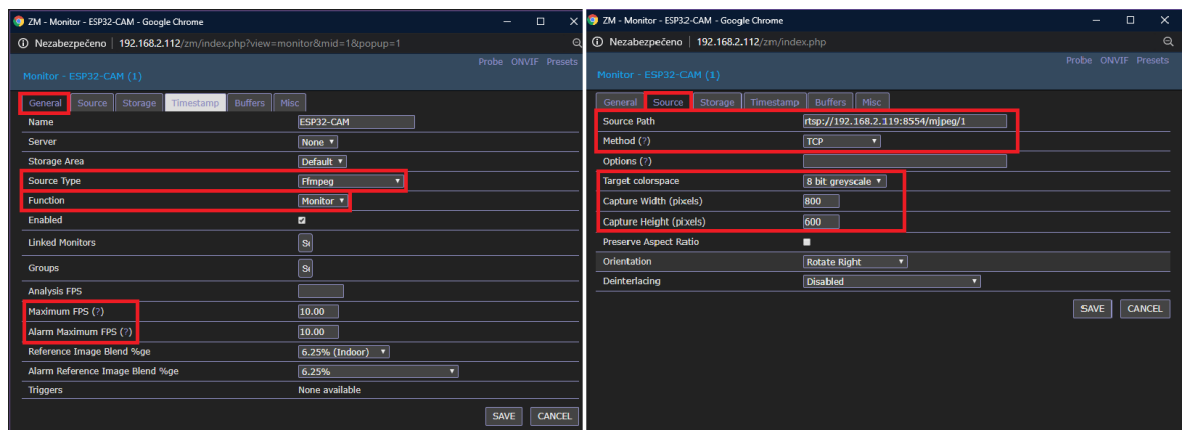
K instalaci samotné služby ZoneMinder je nejdříve nutné nainstalovat téměř jakýkoliv server Linux a dále několik závislostí, např. MySQL, Apache, PHP a Perl. Jakmile je služba ZM správně nainstalována, je pro uživatele ve webovém prohlížeči na url: IP_adresa/zm dostupná úvodní stránka, na které se zařízení ihned po nakonfigurování objeví. Jedním z prvních nastavení bylo požadování přihlášení Options->System->OPT_USE_AUTH a následné přidání dvou uživatelů, a to konkrétně admin (veškeré pravomoci) a spectator (nemá možnost nastavení serveru ani jednotlivých zařízení upravovat, jediné co může, je sledování videopřenosu a detaily jednotlivých eventů).

3.4.2 Přidání kamery

Na úvodní stránce (záložka Console) stisknutím tlačítka ADD začneme konfigurovat kameru. V záložce General jsou nejdůležitější body Source Type (nastavíme Ffmpeg) a Function, kde máme na výběr hned ze šesti možností:

- None - funkce pro vypnutí kamery, obraz nepřenáší ani nenahrává
- Monitor - slouží pouze pro živý přenos, nepodporuje analýzu obrazu, nahrávání, alarmy a eventy
- Modect (motion detection) - veškeré snímky jsou analyzovány a při detekci pohybu v zóně je vytvořen event a spuštěno nahrávání
- Record - permanentní nahrávání obrazu
- Mocord - permanentní nahrávání obrazu s výjimkou eventu při detekci pohybu
- Nodect - speciální mód pro využití s externími čidly, která ovlivňují vytvoření eventu a spuštění nahrávání. Nejprve se spustí speciální script zmtrigger.pl, který naslouchá telnet stream na portu 6802 a čeká na řetězec. Konkrétně může vypadat třeba takto: „1|on+20|1|External Motion|External Motion“. Tato syntaxe spustí nahrávání kamery ID 1 na 20 vteřin s názvem a příčinou External Motion.

V záložce Source jsem nastavil zdroj a protokol vysílání, popřípadě barvu a rozlišení obrazu. Po uložení by mělo dojít k propojení Ai-Thinker ESP32-CAM s ZoneMinder serverem.



Obrázek č. 7 ukázka přidání kamery ve webové aplikaci ZoneMinder

3.5 Externí trigger

3.5.1 PIR senzor

K detekci pohybu jsem si vybral modul HC-SR501, který svou funkci zvládá dokonale. Jelikož se jedná o hotový modul, má také velice jednoduché zapojení 3 pinů - +5V, GND a výstupní pin, kde při detekci naměříme 3,3V.

Ukázka kódu - přerušovací funkce pro detekci pohybu

```
1. static void IRAM_ATTR detectsMovement(void *arg)
2. {
3.     Serial.println("MOTION DETECTED!!!!");
4.     //streamchar = "1|on+20|1|External Motion|External Motion";
5.     trStatus = true;
6. }
7. void setup(){
8.     esp_err_t gpio_install_isr_service(int intr_alloc_flags);
9.     gpio_install_isr_service(ESP_INTR_FLAG_IRAM);
10.
11.     err = gpio_isr_handler_add(GPIO_NUM_13, &detectsMovement, (void *)13)
12.     ;
13.     if (err != ESP_OK){
14.         Serial.printf("handler add failed w error 0x%x \r\n", err);
15.     }
16.     err = gpio_set_intr_type(GPIO_NUM_13, GPIO_INTR_POSEDGE);
17.     if (err != ESP_OK){
18.         Serial.printf("set intr type failed w error 0x%x \r\n", err);
19.     }
```

3.5.2 Způsob oznámení o detekci pohybu

3.5.2.1. Telnet server

První způsob, o který jsem se pokusil, bylo využití telnet serveru na ESP32, kde při vyčtení logické jedničky z jednoho z IO pinů dojde k zaslání požadavku o začátek nahrávání záznamu do paměti. Jak se ale později ukázalo, telnet server je pro ZoneMinder trigger skript nepoužitelný.

3.5.2.2. HTTP server

Princip funkčnosti byl stejný jako u telnetu, ale s tím rozdílem, že při detekci se požadavek pošle prostřednictvím HTTP protokolu. Při testování pomocí konzo-

lového příkazu CURL jsem se dočkal pozitivní odezvy a při každé detekci pohybu došlo k zaslání požadavku o nahrávání. Později při implementaci na ZoneMinder jsem přišel na jednu zásadní chybu. ESP32 nezvládal dvě odlišné služby najednou. Pokusil jsem se tedy o vytvoření dvou asynchronních serverů, což se mi však nepodařilo dotáhnout do konce. Nastalo tedy obrovské dilema, jak problém vyřešit. Rozhodoval jsem se mezi nastavením kamery na ZoneMinder serveru do módu MODECT a mezi analýzou obrazu přímo na úrovni ESP. Nakonec jsem se rozhodl spíše pro možnost první, tj. analyzovat obraz na ZoneMinder serveru, disponujícím čtyřjádrovým procesorem a 4 GB operační pamětí, oproti ESP32, které je už dosti HW zatíženo samotným snímáním obrazu a odesíláním prostřednictvím Wi-Fi. V ZoneMinderu lze nastavovat senzitivitu změn obrazu. Nejeefektivnější je však detekce pohybu externím senzorem PIR.

3.5.2.3. Telnet client

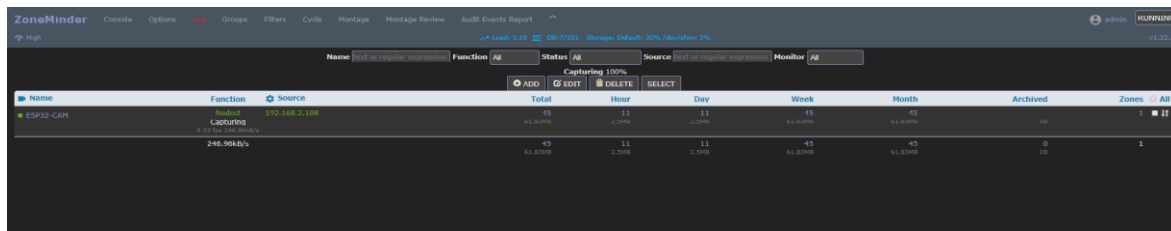
Před termínem odevzdání práce jsem pročítal dokumentaci ZoneMinder serveru a došel jsem k zásadnímu zjištění. Skript nefungoval na principu připojení ZM na ESP, ale zcela opačně. V případě detekce pohybu pomocí PIR senzoru se spustí interrupt funkce, která nastaví status detekce na `true` a později ve funkci `loop()` dojde k připojení na adresu ZoneMinder serveru prostřednictvím telnetu a dojde k zaslání požadavku o sepnutí nahrávání obrazu.

Ukázka kódu - zaslání požadavku o nahrávání přes telnet

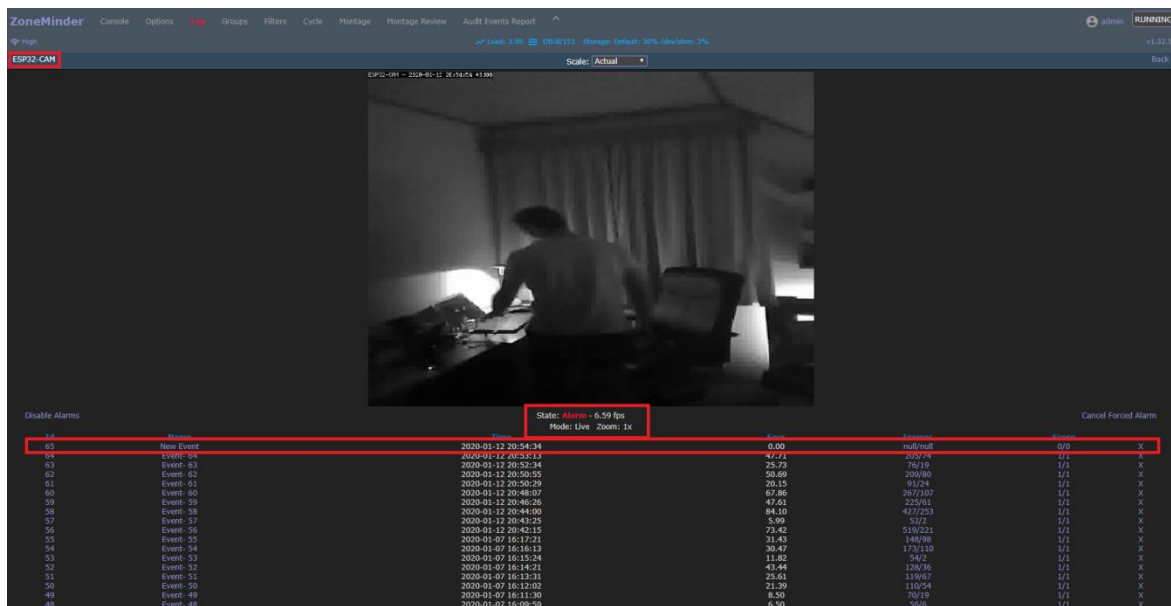
```
1. static void IRAM_ATTR detectsMovement(void *arg)
2. {
3.     Serial.println("MOTION DETECTED!!!!");
4.     trStatus = true;
5. }
6.
7. void loop()
8. {
9.     if (trStatus == true){
10.         telClient.connect(ZM,6802);
11.         telClient.println("1|on+20|1|External Motion|External Motion");
12.         telClient.stop();
13.         trStatus = false;
14.     }
15. }
```

4 VÝSLEDKY ŘEŠENÍ

Aktuální podoba kamerového systému nabízí uživateli funkční webovou aplikaci, kdy má uživatel po přihlášení dostupné své kamery. Chod každé z nich může libovolně upravovat podle jeho administračních práv. Každou kameru lze sledovat zvlášť nebo si je dohromady uspořádat do panelu. Má tak přehled o dění na více místech najednou. Součástí aktuální verze je rozpoznávání pohybu pomocí externího senzoru, kde v případě detekce ESP pošle požadavek na server a započne nahrávání. Po dokončení se vytvoří příslušný záznam, který uživatel může kdykoliv stáhnout do svého zařízení a zhlédnout, nebo najít přímo v úložišti serveru. Pod rolí administrátora lze také spravovat samotné nastavení serveru, uživatele...



Obrázek č. 8 hlavní menu ZoneMinder



Obrázek č. 9 ukázka alarmu

ZÁVĚR

Původním cílem projektu bylo vytvoření komplexního bezpečnostního systému domácnosti. Vedoucí mé práce mi však doporučil, že zajímavější a účelnější by bylo vytvořit raději bezpečnostní kamerový systém. Vytyčené cíle byly v podstatě splněny.

Během psaní dokumentace mě ovšem napadla ještě spousta vylepšení, jako např. přídavné osvětlení pro viditelnost ve tmě, upozornění uživatele prostřednictvím emailu a podobně. Na jejich realizaci už mi ale bohužel nezbyl čas. Na druhou stranu ale musím říci, že i když má práce ještě není úplně dokonalá a aby mohla být využitelná v praxi, bylo by třeba na ní ještě déle pracovat, v průběhu její tvorby jsem se naučil spoustu nových věcí a získal mnoho cenných zkušeností.

SEZNAM POUŽITÝCH INFORMAČNÍCH ZDROJŮ

- [1] geeksville. TenDollarWebcam [online]. In: . [cit. 2019-12-30]. Dostupné z: <https://github.com/geeksville/TenDollarWebcam>
- [2] raphaelbs. Esp32-cam-ai-thinker: product specification [online]. In: . [cit. 2019-12-30]. Dostupné z: https://github.com/raphaelbs/esp32-cam-ai-thinker/blob/master/assets/ESP32-CAM_Product_Specification.pdf
- [3] HC-SR501 PIR MOTION DETECTOR: Product Discription [online]. In: . [cit. 2019-12-30]. Dostupné z: <https://www.mpja.com/download/31227sc.pdf>
- [4] ZoneMinder: User Guide [online]. In: . [cit. 2019-12-30]. Dostupné z: <https://zoneminder.readthedocs.io/en/stable/userguide/index.html>
- [5] Debian: Installation Guide [online]. In: . [cit. 2019-12-30]. Dostupné z: <https://zoneminder.readthedocs.io/en/stable/installationguide/debian.html>
- [6] How to use your external camera's motion detection with ZM [online]. In: . [cit. 2019-12-30]. Dostupné z: https://wiki.zoneminder.com/How_to_use_your_external_camera%27s_motion_detection_with_ZM
- [7] espressif. WiFiTelnetToSerial [online]. In: . [cit. 2019-12-30]. Dostupné z: <https://github.com/espressif/arduinoesp32/blob/master/libraries/WiFi/examples/WiFiTelnetToSerial/WiFiTelnetToSerial.ino>