

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Microprocesorové a vestavěné systémy

Dokumentace k projektu

Zabezpečovací systém s detekcí pohybu

Autor: Jakub Hamadej

Datum: 13. 12. 2024

Obsah

1. Úvod.....	3
2. Prezentační video.....	3
3. Zapojení.....	4
3.1 Nastavení SD karty.....	4
4. Implementace.....	5
4.1 Rozdělení do souborů.....	5
4.1.1 ESP32_Security_Camera.....	5
4.1.2 ESP32_PIR_motion_module.....	5
4.2 Inicializace.....	5
4.2.1 Inicializace kamery.....	6
4.2.2 Inicializace SD karty.....	6
4.2.3 Inicializace WebServeru.....	6
4.3 Pořizování snímků.....	6
4.4 WebServer.....	6
4.5 Snímání čidla.....	6
5. Závěr.....	7
6. Zdroje.....	7
7. AutoEvaluace.....	7

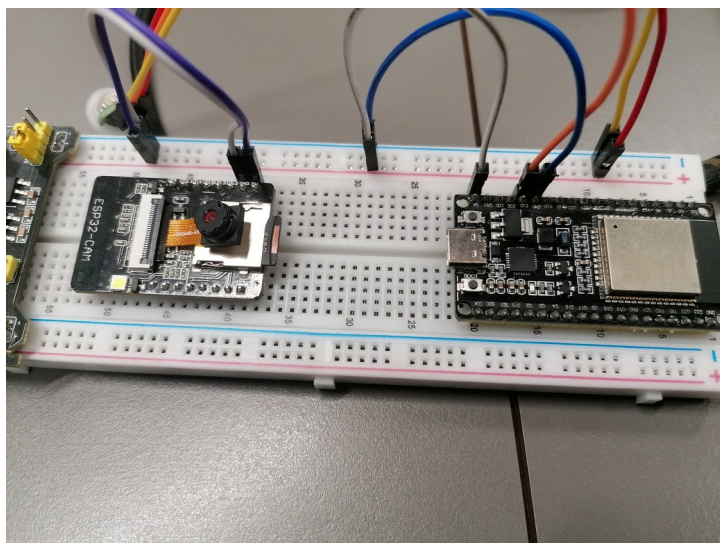
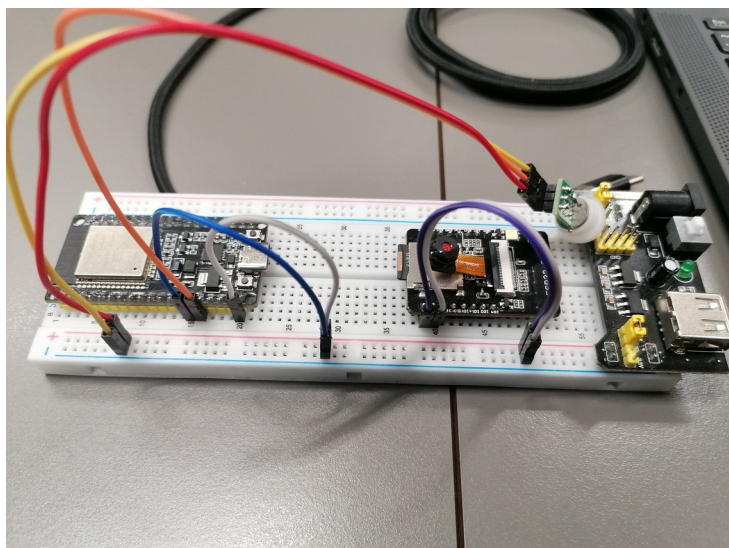
1. Úvod

Cílem projektu je naprogramovat “Zabezpečovací zařízení” pomocí ESP32-CAM, druhého ESP32 a pohybového čidla PIR, kdy bude možné na dálku se připojit k výslednému zařízení a prohlédnout si pomocí webového prohlížeče zachycené fotografie, případně pořídit novou fotografii.

2. Prezentační video

https://youtu.be/ySOQgiz_y8c

3. Zapojení



Na obrázcích můžeme vidět ESP32-CAM, ESP32 a čidlo detekce pohybu PIR, které jsou napájeny zdrojem, ESP32 jsou připojena na 5V a GND pinch, PIR čidlo je připojeno na 5V, GND a svůj senzorový výstup má připojený na ESP32 k GPIO13.

3.1 Nastavení SD karty

Kartu zformátujeme a nastavíme na formát FAT32. Poté připojíme do ESP32-CAM.

4. Implementace

K implementaci je použito frameworku Arduino a vyvíjeno v prostředí Platformio.

Při implementaci bylo využito k inspiraci již existujících projektů [2, 3] jako částečné základní kostry, které následně byli rozšířeny a upraveny pro potřeby projektu.

Je používáno dvou ESP32, kdy zařízení vybavené kamerou se chová jako server a druhé se chová jako klasický klient. Komunikace probíhá přes Wifi, kdy příkazy se posílají pomocí HTML hlavičky.

4.1 Rozdělení do souborů

4.1.1 ESP32_Security_Camera

ESP32_security_system.hpp

Obsahuje deklarace funkcí a externy globálních proměnných, používaných napříč programem.

handleClient.cpp

Obsahuje funkce pro komunikaci s uživatelem skrz webserver.

main.cpp

Obsahuje funkce **setup** a **loop** jako základ každého Arduino based programu.

pictureNumberManipulation.cpp

Obsahuje funkce pro načítání a ukládání počtu fotek uložených na SD kartě

setupCamera.cpp

Obsahuje funkce pro počáteční nastavení kamery

setupSDcard.cpp

obsahuje funkce pro počáteční nastavení SD karty a načtení počtu fotek

takeAPhoto.cpp

Obsahuje funkce pro pořízení fotografie a následné uložení na SD kartě

4.1.2 ESP32_PIR_motion_module

main.cpp

Obsahuje inicializaci wifi, připojení se k serveru jako klienta a snímání PIR senzoru

4.2 Inicializace

Při použití Arduino frameworku, inicializace desky probíhá ve funkci **setup** kde se postupně nastaví porty pro správnou komunikaci s periferiemi a případně dalšími zařízeními.

4.2.1 Inicializace kamery

Inicializace probíhá ve funkci **setupCamera**, kde se nastaví piny pro komunikaci s kamerou, následná samotná inicializace kamery probíhá pomocí knihovní funkce **esp_camera_init**. Pokud selže inicializace kamery, celý proces se ukončí.

4.2.2 Inicializace SD karty

Probíhá v **setupSDcard**, kde se ověří že je karta v zařízení, pokud není, celý proces inicializace se ukončí. Zároveň zde probíhá načtení počtu obrázků na SD kartě.

4.2.3 Inicializace WebServeru

WebServer je inicializován přímo v **setup** funkci, kde se nastaví statická ip adresa serveru, na kterém bude odpovídat a následně se spustí

4.3 Pořizování snímků

Probíhá ve funkci **takeAPhoto**, kde se volá knihovní funkce pro pořízení fotografie **esp_camera_fb_get**, která vrátí ukazatel na camera buffer, kde se nachází data ke snímku. Následně je snímek uložen na SD kartu. Z důvodu bugu v samotné knihovně vrací funkce **esp_camera_fb_get** buffer na snímek o jedna starší [5], a proto musí být volán 2x rychle za sebou.

4.4 WebServer

Poté, co se připojí zařízení a zadá předem danou IP adresu ve vyhledávači, vrátí ESP32 uživateli webovou stránku, kde lze prohlížet zaznamenané fotografie nebo nechat vyfotit novou fotografii.

O toto se stará funkce **handleClient**. Tato funkce zpracovává příchozí GET žádosti a odesílá zpět dané požadavky.

Jednou z nejdůležitějších funkcí které jsou volány je **sendPicture**, která odesílá klientovi obrázky, které načte na SD kartě.

4.5 Snímání čidla

Čidlo je snímáno přes přerušení sledovaném na GPIO13, kdy při zaznamenání pohybu pošle klient na server příkaz na pořízení fotografie.

5. Závěr

Původní záměr, aby na celý projekt stačilo jedno ESP32 se nezdařil. Problém nastal s možností využití PINů, kdy při komunikaci serveru se využívají i PINy, u kterých jsem neměl tušení že se bude něco dělat. Z toho důvodu nešlo připojit zařízení PIR ke kameře.

Z toho důvodu jsem se rozhodnul více přiblížit původnímu zadání se dvěma ESP32, kdy jsem druhé ESP32 zapojil jako dalšího klienta, který posílá ESP32-CAM příkaz na zachycení fotografie.

Toto se povedlo úspěšně.

6. Zdroje

- [1] Š - ESP32: Zabezpečovací systém s detekcí pohybu. Online. Dostupné z: https://www.vut.cz/studis/student.phtml?script_name=zadani_detail&apid=281143&zid=58236. [cit. 2024-12-12].
- [2] ESP32-CAM Take Photo and Save to MicroSD Card. Online. Dostupné z: <https://randomnerdtutorials.com/esp32-cam-take-photo-save-microsd-card/>. [cit. 2024-12-12].
- [3] How to Set an ESP32 Access Point (AP) for Web Server. Online. Dostupné z: <https://randomnerdtutorials.com/esp32-access-point-ap-web-server/>. [cit. 2024-12-12].
- [4] ESP32-CAM 2.4GHz WiFi+Bluetooth Modul. Online. Dostupné z: <https://www.laskakit.cz/ai-thinker-esp32-cam-2-4ghz-wifi-bluetooth-modul/#relatedFiles>. [cit. 2024-12-12].
- [5] Esp32 cam takes old image. Online. 2021. Dostupné z: <https://www.esp32.com/viewtopic.php?t=24566>. [cit. 2024-12-12].
- [6] PIR senzor pohybu AM312. Online. Dostupné z: <https://dratek.cz/docs/produkty/1/1051/1565175763.pdf>. [cit. 2024-12-13].

7. AutoEvaluace

E – 2b

F – 5b, plně splňuje zadání projektu

P – 2b, s prezentací se podle emailu nepočítá

Q – 2b

D – 3b