IMP - projekt
Zabezpečovací systém

Antonín Jarolím
xjarol06

16. prosince 2022

Obsah

Úvod

Projekt se zabývá implementací zabezpečovacího systému, který po detekci pohybu ve střežené oblasti vyfotí narušitele. K řešení je využito ESP32 desky Wemos D1 R32[1], ke které je připojený detektor pohybu[2] na GPIO16 a digitální kamera ESP32-CAM[3], která fotí fotky a také vytváří AP. Také implementuje http server, který slouží pro zahájení focení fotky a jejich následné zpřístupnění.

Pro řešení projektu byla zvolena platforma esp-idf[4] a rozšíření vscode esp-idf.[5], což umožnilo pracovat na nejnižší vrstvě. Aplikace *Http server kamera*2.2 je ve složce */camera* a aplikace *Detektor pohybu*2.1 ve složce */detector*.

Pro inicializaci a používání desky bylo nutné využít knihovnu esp32-camera[6]. Tato knihovna je ve složce */camera/lib*. Řešení bylo dosaženo upravením příkladu *take_picture.c* v této knihovně. Bylo tedy využito struktury *camera_config* a funkcí *init_camera()*, *esp_camera_init* z tohoto příkladu. Druhá zmíněná funkce využívá API rozhraní knihovny - konkrétně funkce *cam_init(), camera_probe(), cam_config() a cam_start()*. Na desce byla zapnuta podpora pro externí RAM paměť připojenou přes SPI rozhraní, jelikož je k tomu vybízeno v úvodu zmíněného příkladu.

Obě zmíněné aplikace využívají z modulu *esp_wifi.h* tyto funkce.

- *esp_netif_init()* pro inicializaci síťového rozhraní
- *esp_event_loop_create_default()* pro vytvoření event loopu a funkci
- *esp_wifi_init()*

Aplikace *Http server kamera* slouží jako AP a využívá proto funkci *esp_netif_create_default_wifi_ap()*, která vytváří defaultní WIFI přístupový bod. Aplikace *Detektor pohybu* se připojuje na vytvořené AP za pomocí funkce *esp_netif_create_default_wifi_sta()*. Řešení práce s wifi bylo částečně přebráno a upraveno z příkladů v *wifi/getting_started* [7]. Také se registrují na různé eventy, které zpracovávají stavovým výpisem.

Další využitá knihovna je *"esp_http_server.h"*, kterou využívá aplikace aplikace *Http server kamera* pro vytvoření http serveru. Využívá funkce *httpd_resp_send()* pro odeslání odpovědi a funkce *httpd_resp_send_500()* pro signalizaci neúspěšné operace. Byl využit příklad *simple http server*[8].

Pro odesílání dotazu na server se vychází z příkladu *http_request_example_main* [9], který využívá schránky pro odeslání http dotazu.

Také byla vyžity příklady[10] ze stránek platformia, které popisují jak pomocí esp32-cam vytvořit snímek.

[1] Schéma desky Wemos D1 R32
[2] Detektor pohybu
[3] Digitální kamera ESP32-CAM
[4] Esp-idf dokumentace
[5] Esp-idf vscode rozšíření
[6] Github repozitář knihovny esp32-camera
[7] ESP32 příklady - /wifi/getting-started
[8] ESP příklady - /protocols/http_server/simple
[9] ESP příklady - /protocols/http_request/
[10] Platformio příklady využití esp32-cam

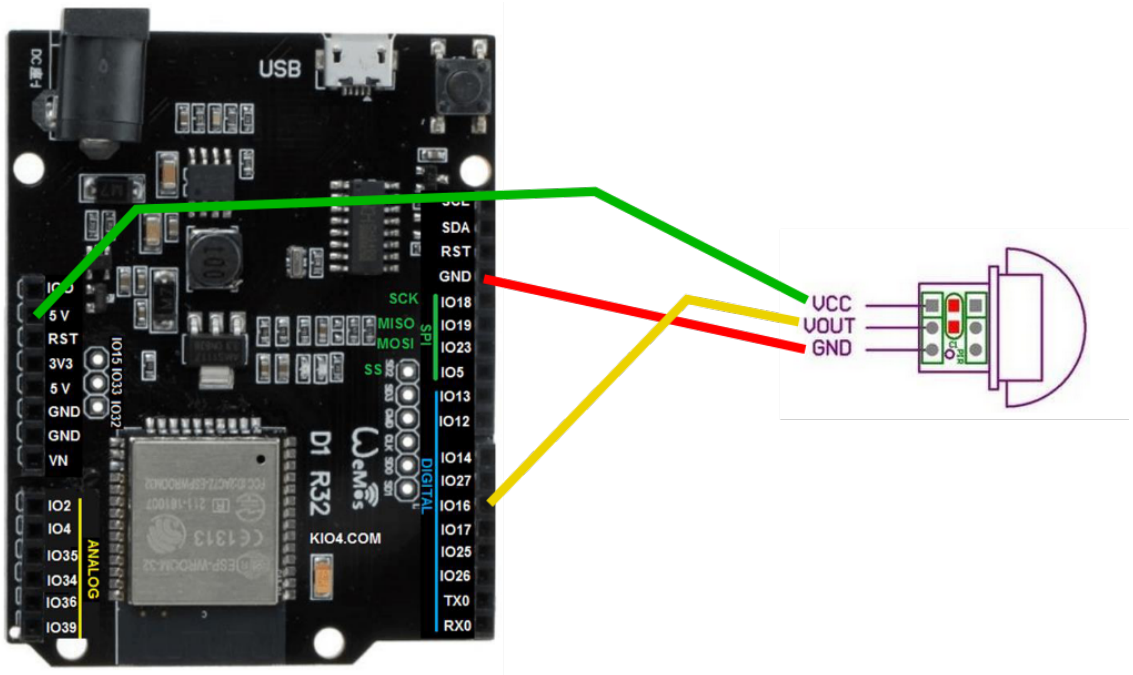
Implementace

Detektor pohybu

Pro detekci je využita deska ESP32, ke které je zapojený detektor pohybu na GPOI port 16. Napájecí nožka detektoru je připojena na 5V a GND je na desku uzemněno. Port GPOI 16 je propojen s VOUT a po zapnutí aplikace nastaven jako vstupní, což umožňuje kontrolu jeho stavu. Tato kontrola je následně prováděna v nekonečné smyčce s krátkou prodlevou. Také je inicializován port GPIO2, který je zapojen na desce k interní LED diodě. Tato dioda rozsvícením signalizuje, že čidlo detekovalo pohyb.

Pokud čidlo detekuje pohyb, tak se zavolá funkce *http_new_photo()*. Tato funkce slouží k odeslání http požadavku na server, využívá k tomu linux schránky. Pošle http dotaz na adresu 192.168.4.1/new_photo, port 80. Dotaz je specifikován pod identifikátorem *NEW_PHOTO_REQUEST*. Funkce také vypíše na výstup celý příchozí packet.

Aplikace se po spuštění připojí k wifi, za použití konfigurace uvedené v souboru *Kconfig.projbuild*, respektive souboru *sdkconfig*. Konfigurace je nastavena tak, aby se zařízení připojilo k AP, který vytvoří druhé zařízení. Inicializace wifi spojení proběhne voláním funkce *wifi_init()*. Tato funkce inicializuje nvs flash, poté se pokusí připojit ke specifikované wifi.



Obrázek 1: Schéma zapojení detektoru pohybu na ESP32 desku

Http server kamera

Po spuštění aplikace dojde k inicializaci nvs flash. Pokud proběhne úspěšně, tak se pokračuje na inicializaci wifi připojení. To proběhne ve funkci *wifi_init_softap()*. Tato funkce inicializuje síťové rozhraní a vytvoří event loop. Vytvoří wifi spojení s identifikátorem specifikovaným na začátku souboru makrem *EXAMPLE_ESP_WIFI_SSID* s heslem *EXAMPLE_ESP_WIFI_PASS* na kanálu 6.

Pokud proběhne wifi inicializace, následuje inicializace kamery ve funkci *init_camera()*. Následuje inicializace webového http serveru, což je provedeno ve funkci *start_webserver()*.

Webové rozhraní je definováno následovně.

- /bmp Vytvoření snímku ve formátu bmp a jeho odeslání v odpovědi na požadavek. Slouží jako ověření funkčnosti kamery a správnosti funkčnosti zařízení.
- /new_photo Vytvoření snímku a jeho uložení do RAM paměti.
- /get_photo Zaslání snímku předem vytvořeného pomocí předešlé funkce. Očekává jeden parametr, *at_index*, jehož hodnota je číslo specifikující index snímku, který má být zaslán.

Pokud přijde požadavek na adresu /new_photo, tak se zavolá funkce *new_photo_handler()*, která tuto událost ošetří. Tato funkce pozastaví plnění frame bufferu funkcí *esp_camera_fb_get()*, také tím získá odkaz na místo v paměti, kde je frame buffer uložen. Následně zkontroluje, zda tato akce proběhla v pořádku, pokud ne, fotka se neuloží a na dotaz je odpovězeno chybou 500. V případě úspěchu se volá funkce *frame2jpg_cb()*, která zpracovává frame buffer a převádí ho po částech na jpg formát. Tato funkce bere 4 parametry, první z nich je frame buffer, druhá je požadovaná kvalita snímku, následuje identifikátor callback funkce, která zpracovává části zpracovaného obrázku a jeden libovolný argument, který je této funkci předán. Předaná callback funkce na zpracování části obrázku se jmenuje *jpg_encode_stream*.

Této funkci jsou předávány data, tedy části obrázku, index, na který mají být data zapsána, délku těchto dat a předaný argument. Tato funkce tedy zapisuje postupně části dat do bufferu, který má velikost 20kb. Po zpracování snímku je funkce zavolána ještě jednou s délkou 0, což je signál pro uložení bufferu do paměti. Alokuje se tedy přesný počet potřebných bytů k uložení této fotky a na globální seznam *photos* je uložen na správný index nový snímek. Data nejsou ukládána na perzistentní úložiště, jsou tedy nenávratně smazána po odpojení zařízení od napájení.

Snímky jsou ukládány do struktury *photo_t*, která ukládá pointer na data a velikost těchto dat. Tato struktura je tedy perfektně připravena pro odeslání funkcí *httpd_resp_send()*, která bere pointer na data a velikost těchto dat. Je volána funkcí *send_photo_handler()*, která zpracovává požadavek na adrese */get_photo*. Zpracuje parametr at_index pomocí funkce *httpd_query_key_value* a dalších funkcí z modulu *http_server.h*.

Zaručení funkčnosti a signalizace chyb

Občas se stane, že inicializace kamery neproběhne úspěšně. Bylo proto zavedeno opatření, které po úspěšné inicializaci 3x zabliká s vestavěnou diodou na GPIO4. Takto je uživatel obeznámen o stavu zařízení bez toho, aniž by musel zařízení monitorovat a v případě neúspěchu může zařízení restartovat.

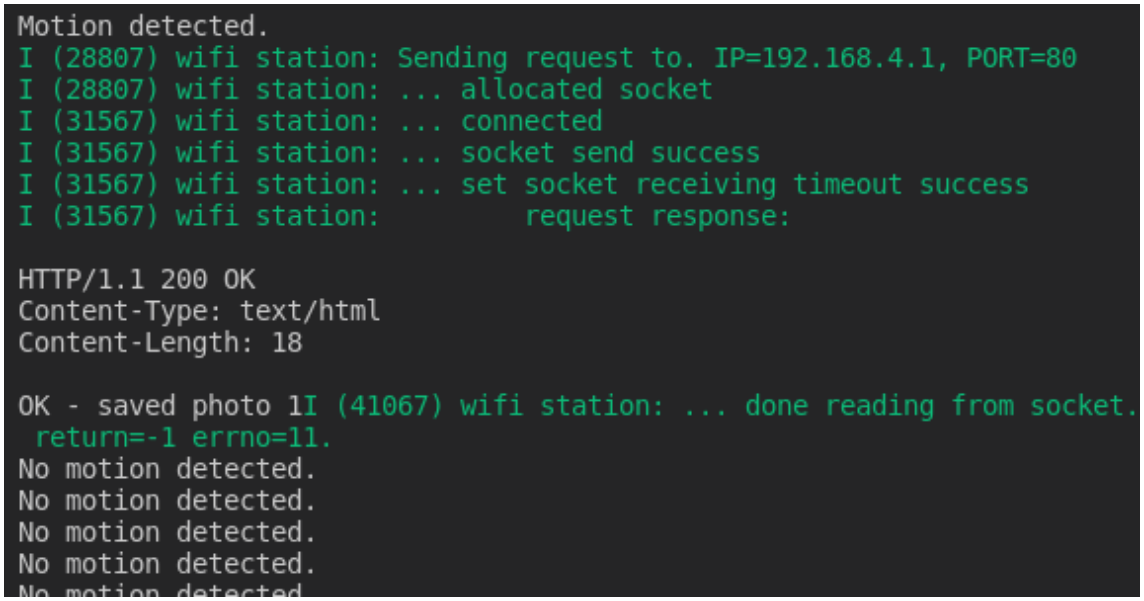
Detektor pohybu také rozsvítí zabudovanou ledku na portu GPIO2, pokud detekuje pohyb. Ledka takto slouží jako zpětná vazba. Pokud po detekci pohybu jakkoliv selže odeslání požadavku, není to fatální problém. Zařízení tedy packet neodešle a pokračuje dále ve své práci.

Je možné, že v případě zapisování fotky do bufferu bude výsledná fotka větší, než je stanovená velikost bufferu. V tomto případě se do bufferu přestane zapisovat, protože už je plný. Ukončující volání této funkce však zapíše celý buffer do nového snímku. Toto bylo zavedeno jako zabezpečení funkčnosti aplikace i v případě, že se podaří vyfotit neobvykle fotku.

Zařízení, které se připojuje k vytvořené wifi síti, se pokusí 5 krát připojit. Pokud se mu to nepovede ani na pátý pokus, vypíše tento neúspěch na výstup a následně přejde do chybového stavu, kde v nekonečné smyčce bliká ledkou, jako signál neúspěchu připojení k wifi síti.

Shrnutí

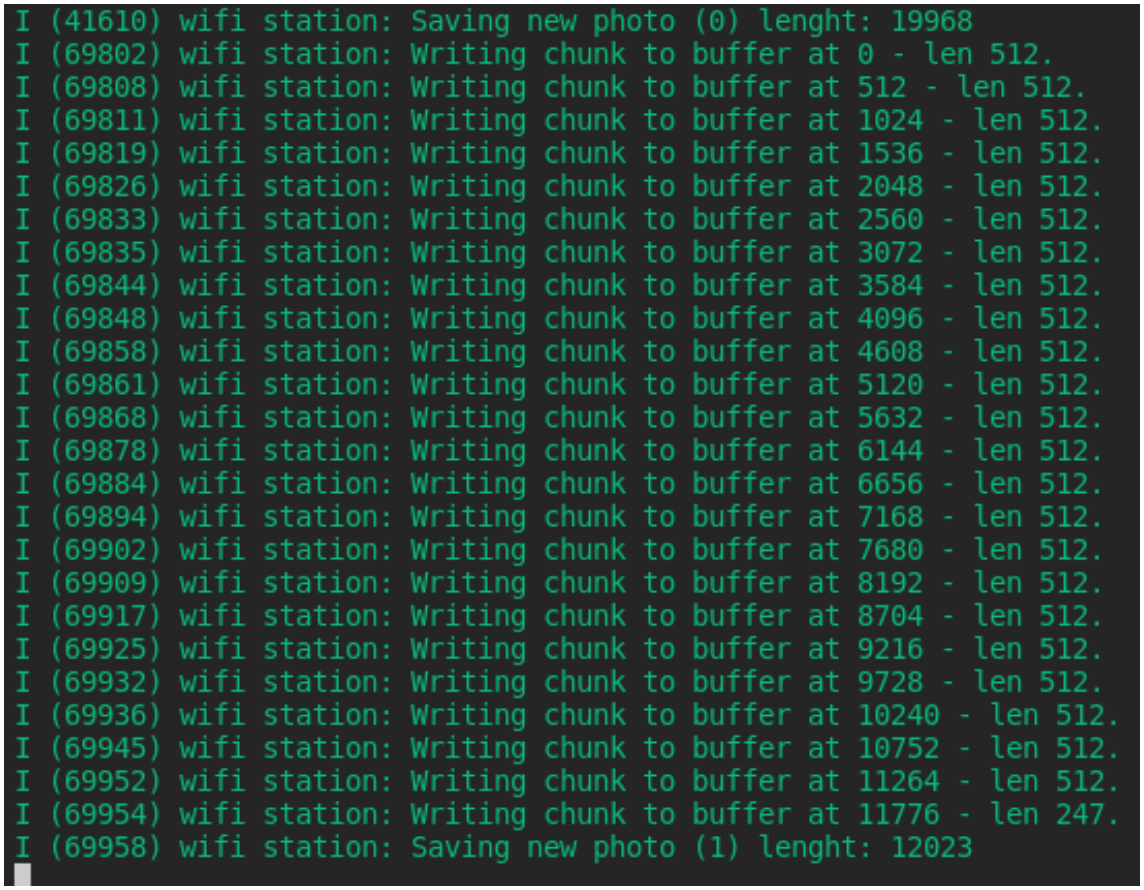
Funkčnost zařízení můžeme vidět na následujících snímcích a také na videu, které je ke dostupné na platformě youtube na tomto odkaze[11].



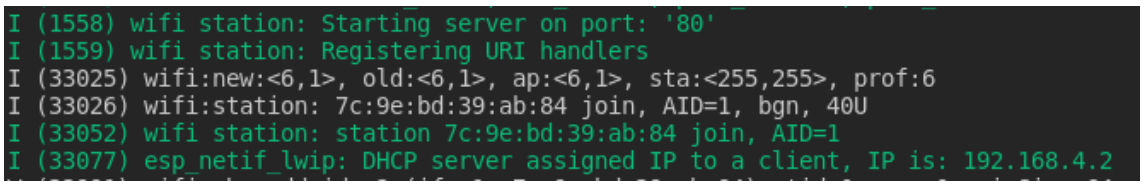
Obrázek 2: Záznam detekce pohybu, vypsání odpovědi, která přišla ze serveru a následného znovu spuštění detekce pohybu.

Řešení považuji za úplné a korektní a dokumentaci za propracovanou. Proto bych se ohodnotil plným počtem bodů.

[11] Odkaz na video je také v přiloženém souboru *video.txt*



Obrázek 3: Záznam z ukládání části dat do bufferu a následného uložení nového snímku.



Obrázek 4: Inicializace http serveru a přidělení ip adresy klientovi.