

ZÁVĚREČNÁ STUDIJNÍ PRÁCE

dokumentace

Arduino Home-security alarm

Tomáš Hendrych

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

Třída: IT4

Školní rok: 2020/2021

Poděkování

Děkuji panu učiteli Ing. Petru Grussmannovi za cenné rady a panu učiteli Ing. Jiřímu Miekishovi za pomoc se součástkami.

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. 2020

podpis autora práce

ANOTACE

Projekt se zabývá tvorbou bezpečnostního systému v podobě jednoduchého alarmu. Skládá se z hardwarové a softwarové části. Základ hardwaru tvoří vývojová deska Arduino ESP32 D1 R32. Po zadání aktivačního PIN kódu se spustí bezpečnostní systém spolu se senzorem pohybu, ten po zaznamenání pohybu odešle signál do vývojové desky Arduina, která následně rozsvítí LE diodu a na LC display vypíše hlášku: „!Intruder inside!“ Po vypsání hlášky musí operátor zadat deaktivací PIN kód, aby alarm vypnul. Po úspěšné deaktivaci display vypíše hlášku: „SYSTEM DEACTIVATED“ a přepne výstražnou červenou LE diodu na zelenou. Pro opětovné aktivování alarmu se musí opět zadat aktivační PIN kód, který uvede do provozu senzor pohybu PIR. Jak už jsem zmínil, program je napsán v jazyce Arduino, což je kombinace jazyků C a C++.

OBSAH

ÚVOD.....	5
1 VÝROBA BEZPEČNOSTNÍHO SYSTÉMU CHYBA! ZÁLOŽKA NENÍ DEFINOVÁNA.	
2 VYUŽITÉ TECHNOLOGIE	9
2.1 Hardware	9
2.1.1 Seznam součástek	9
2.1.2 ESP32 D1 R32	9
2.1.3 PIR motion sensor HC-SR501	9
2.1.4 LC Display 16x2	10
2.1.5 Membránová číselná klávesnice 4x3	10
2.1.6 LE Diody	10
2.2 Napájení	11
2.3 Software	11
2.3.1 Jazyk Arduino	11
2.3.2 Visual Studio code	11
2.3.3 Platformio	11
2.3.4 Arduino IDE 1.18.3	11
3 ZPŮSOBY ŘEŠENÍ A POUŽITÉ POSTUPY	12
3.1 Zadání hesla	12
4 VÝSLEDKY ŘEŠENÍ	13
4.1 Podoba hardwarového zařízení	13
ZÁVĚR	14
SEZNAM POUŽITÝCH INFORMAČNÍCH ZDROJŮ	15
SEZNAM PŘÍLOH	16

ÚVOD

Rozhodl jsem se postavit a naprogramovat funkční model alarmu, protože mě tato problematika a obecně problematika bezpečnosti dosti zajímá, hlavně pak v posledním roce, tudíž jsem měl víceméně jasno v tom, co budu dělat za projekt. Chtěl jsem si vyzkoušet, jak to asi může vypadat při vývoji těchto bezpečnostních systémů, na jaké věci si dát pozor a jaké maličkosti dotáhnout k dokonalosti.

Hlavním cílem projektu bylo vytvoření funkčního modelu domácího alarmu, který může operátor kdykoliv aktivovat či deaktivovat zadáním PIN kódu, např. když jede do práce nebo na dovolenou, může si tak dopřát chvilky mimo domov s vědomím, že je jeho majetek alespoň trochu chráněn.

Protože se jednalo o můj první projekt, rozhodl jsem se využít alespoň trochu známé technologie, s vývojovými deskami Arduino jsem se setkal již v minulosti ve škole, a tak jsem trochu tušil, co mě bude čekat. S dalším hardwarem, jako je například display nebo číselná klávesnice Keypad, kterým se budu věnovat později v dokumentaci, jsem se setkal poprvé, a proto jsem se musel naučit, jak tyto zařízení fungují a komunikují s vývojovou deskou.

V této dokumentaci podrobně popisuji výrobu bezpečnostního systému, princip jeho fungování a zapojení součástek. Na začátku se zmiňuji o problémech, se kterými jsem se při vývoji setkal. Dále v práci popisuji použité součástky a technologie a následně se rozepisuji o samotném principu systému. V závěru se zabývám nedostatky a možnými vylepšeními do budoucna.

1 VÝROBA BEZPEČNOSTNÍHO SYSTÉMU

Pro začátek bylo potřeba vybrat vhodnou vývojovou desku, už před tím jsem byl rozhodnut pro Arduino, ale nevěděl jsem, která vývojová deska mi bude vyhovovat nejvíce. Nejprve mě napadlo využít Arduino UNO nebo ESP32, pan učitel Ing. Petr Grussmann mi doporučil ESP32 kvůli jeho možnosti WIFI připojení. Bohužel deska neměla dostatečné množství potřebných pinů, a tak jsem si vybral trochu jiný model, který měl o něco více pinů pro práci. Tím byl ESP32 D1 R32, jehož jádro tvoří procesor od firmy Espressif.

Jelikož jsem s většinou součástek pracoval poprvé, rozhodl jsem se postupovat po malých krocích. Postupně jsem tedy zapojoval, programoval a zkoušel jednu součástku za druhou. Pro všechny případy jsem si také dokoupil Arduino UNO, na které jsem testoval součástky, abych je nespálil a následně nečekal na dodání nových součástek.

Jak už jsem se zmínil, tak po odzkoušení a úspěšném zapojení součástky jsem ihned začal s jejím programováním, můžete tak na mém Githubu nalézt několik kódů různých součástek od senzoru pohybu, přes display, až po číselnou klávesnici a následné sloučení jejich kódů, kdy při stisknutí klávesy display zobrazí stisknutou klávesu.

Největším problémem byl asi samotný LC Display, kdy po úspěšném nahrání funkčního kódu často zobrazoval pouze nebo částečně cizí asijské znaky, když jsem se posléze na internetu dozvěděl, nebyl jsem jediný, kdo měl podobný problém. Někteří tvrdili, že jde o vadu systému či zbytkový kód z výroby, ale odpověď na řešení mi nikdo neposkytl. Naštěstí mi po několika zmáčknutích restartovacího tlačítka na vývojové desce začal display vypisovat postupně znaky v latině a následně i samotný text, který byl naprogramován.

Dalším velkým problémem, který stále přetrvává, jsou knihovny. Snažil jsem se kód programovat ve VS Code, ale přidělené knihovny odmítaly spolupracovat jak mezi sebou, tak s VS Code, proto jsem se uchýlil k Arduino IDE, kde se kód podařilo zkompileovat a konečně nahrát na vývojovou desku. Na VS Code jsem ale nezanevřel a po nějaké době jsem jej opět zkusil zprovoznit. Nejnovější verze nedokázala kompilovat kód, a tak jsem zkoušel starší verze, ale ani ty nešly. Poté jsem přešel znovu na nejnovější verzi a kód bylo možné kompilovat. Jenomže vyskočily nějaké chyby, ty se mi však podařilo obejít smazáním složky „pio“ a opětovnou kompilací, která byla již úspěšná. Následně jsem tedy postupoval s programováním ve VS Code.

Zlom opět nastal, když jsem začal používat knihovnu „password.h“, která opět nechtěla správně komunikovat se zbylými knihovnami. Na internetu jsem se nedopátral příčiny a myslel jsem si, že budu muset knihovnu přestat používat. Naštěstí mi knihovna správně fungovala v Arduino IDE, kde jsem kód zkompiloval. Později jsem ale narazil na další problém s knihovnou „password.h“, a to ten, že nedokázala komunikovat s dalšími knihovnami, jako jsou např. „SPIFFS.h“ a „ESPAsyncWebServer.h“, které byly potřebné pro komunikaci vývojové desky se serverem a k přístupu na internet. Tento problém jsem stále nedokázal vyřešit.

2 VYUŽITÉ TECHNOLOGIE

2.1 Hardware

2.1.1 Seznam součástek

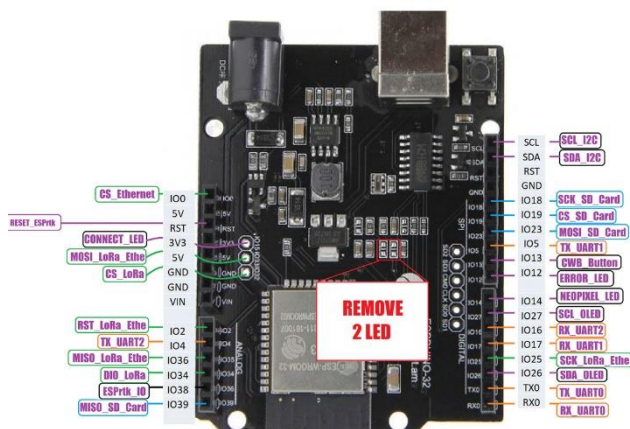
- Vývojová deska ESP32 D1 R32.
- PIR motion sensor HC-SR501.
- LC Display 16x2.
- Membránová číselná klávesnice Keypad 4x3.
- Červená LE Dioda.
- Zelená LE Dioda.
- 2x rezistor 330 Ω .
- Nepájivé pole.
- Drátky.

2.1.2 ESP32 D1 R32

Základ tvoří vývojová deska ESP32 D1 R32 (obrázek č. 1). Deska je založena na modelu ESP32, ale obsahuje několik pinů navíc a má možnost analogového napájení 5V-12V. Deska dále obsahuje 4 MB Flash paměti. Na desce se nachází až 15 digitálních pinů a 6 analogových pinů, dva 5V piny, jeden 3,3V a třikrát pin GND (obrázek č. 2).



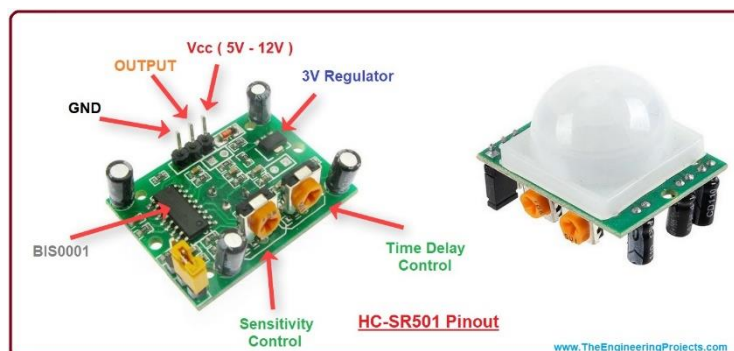
Obrázek č. 1 ESP32 D1 R32



Obrázek č. 2 Piny

2.1.3 PIR motion sensor HC-SR501

Systém funguje na principu změny světla v určité vzdálenosti. Čidlo snímá, zda-li dochází v určité vzdálenosti k nějakému pohybu. V případě že čidlo zaznamená pohyb, sepne. Senzor tvoří tři piny, pin pro kladný náboj (+), pro záporný (-, GND) a pin pro signál, který sepne, když senzor zaznamená pohyb (obrázek č. 3). Senzor dále obsahuje dva trimry, jeden pro regulaci vzdálenosti (až do 7m) a jeden pro regulaci délky signálu po sepnutí (0,3s-5min).



Obrázek č. 3 PIR pinout

2.1.4 LC Display 16x2

Jako vstupní zařízení jsem použil LC Displej 16x2, který dokáže zobrazit 32 znaků na dvou řádcích (16x2). Provoz displeje obstarává knihovna „LiquidCrystal.h“.

2.1.5 Membránová číselná klávesnice 4x3

Uživatelský vstup a ovládání je řešeno pomocí membránové klávesnice o dvanácti klávesách (4 řádky a 3 sloupce). Pro fungování klávesnice je potřeba sedm pinů. Fungování obstarává knihovna „Keypad.h“. Pro správné fungování je potřeba do dvourozměrného pole zapsat hodnoty kláves.

```
const byte ROWS = 4;
const byte COLS = 3;

char keys[ROWS][COLS] = {
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'},
  {'*','0','#'}
};
```

Piny, na kterých je klávesnice zapojena, na vývojové desce.

```
byte rowPins [ROWS] = {18, 19, 23, 5};
byte colPins [COLS] = {13, 12, 14};
```

2.1.6 LE Diody

Součástí systému jsou LE Diody, červená a zelená. Po aktivaci systému se rozsvítí červená LE Dioda a po deaktivaci se rozsvítí zelená LE Dioda.

2.2 Napájení

Napájení je možné řešit pomocí dvou způsobů. První je napájení ze sériového portu USB, který slouží i pro nahrávání kódu na desku a umožňuje napětí až 5V, které pro provoz systému stačí. Druhý způsob je napájení pomocí analogového portu na desce, který umožňuje napětí od 5V až do 12V.

2.3 Software

2.3.1 Jazyk Arduino

Program, který slouží k ovládání součástek bezpečnostního systému jsem napsal v jazyce Arduino, který byl vytvořen k programování integrovaných obvodu a vývojových desek. Je principem velmi podobný programovacím jazykům C a C++.

2.3.2 Visual Studio Code

Zkráceně VS Code, je editor zdrojového kódu vyvíjený společností Microsoft pro operační systémy Windows, Linux a macOS. Jedná se o svobodný program volně ke stažení na oficiálních stránkách. Obsahuje podporu pro GIT a GitHub, zvýraznění syntaxí a instalaci vlastních jazyků a programů.

2.3.3 Platformio

Vývojové prostředí používané pro tvorbu Arduino kódů ve VS Code.

2.3.4 Arduino IDE 1.8.13

Jako hlavní vývojové prostředí jsem používal VS Code, ale po problémech, které jsem popisoval výše jsem se uchýlil i k tomuto vývojovému prostředí pro nahrávání Arduino programů do vývojové desky, které je k tomu přímo uzpůsobeno.

3 ZPŮSOBY ŘEŠENÍ A POUŽITÉ POSTUPY

3.1 Zadání hesla

Heslo je uloženo přímo v programu v proměnné „password“ („2345“) a jeho správu obstarává knihovna „password.h“, ta pomáhá volat funkce a dál s heslem pracovat, např. při ověřování hesla a při volání funkce password.reset().

Funkce, která se volá po zadání špatného PINu.

```
void invalidCode() { // zadal jste špatný PIN
    password.reset(); //zadaný PIN se vyresetuje pro další pokus
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("INVALID CODE!");
    digitalWrite(greenLed, HIGH);
    digitalWrite(redLed, LOW);
    delay(2000);
    digitalWrite(greenLed, LOW);
}
```

4 VÝSLEDKY ŘEŠENÍ

4.1 Podoba hardwarového zařízení

Systém je v současné době zcela funkční v základní podobě. Bezpečnostní zařízení jsem stavěl pouze jako model, a proto jsem se rozhodl k němu postavit malý model domu z kartonového papíru pro případnou prezentaci. Ten je v současné době nedostavěný. Po jeho dokončení vlepím součástky dovnitř, popřípadě nalepím na vnější stěnu domu pro jednodušší přístup k zadávacímu zařízení (membránová klávesnice). Do stěny plánuji zabudovat LC Display pro jednodušší čtení informace, zda je systém aktivní či ne.

Závěr

Cílem projektu bylo vytvořit funkční model domácího alarmu, který by uživatel spustil při opouštění domu, bytu, či kteréhokoliv objektu, při zadání aktivčního PINu a opět jej vypnul při zadání deaktivčního PINu po příchodu. Tento cíl byl splněn, bezpečnostní systém v této chvíli a v této podobě funguje bez problémů. Ačkoliv je zařízení hotové, napadá mě hned několik možných vylepšení. Rozhodně je možné zredukovat používané piny klávesnice ze sedmi digitálních na jeden analogový nebo využít technologie I2C pro LC Display, pro zredukování pouze na pár pinů. Dále možnost odesílání dat na internet, kdy při zaznamenání pohybu by se zapsal na server přesný čas spuštění i vypnutí alarmu a čas případného neo-právněného vstupu. Následně bych všechny součástky rád seskládal do nějaké přenositelné podoby, kdy by se několik pohybových senzorů pouze připojilo drátky a tím by alarm pokryl více místností domu. Dále bych chtěl vyřešit křehkost zařízení a upevnit drátky a potřebné spoje pinů, aby nevypadávaly a aby se nevypojovaly, toho by se dalo dosáhnout umístěním na plošný obvod místo na nepájivém poli.

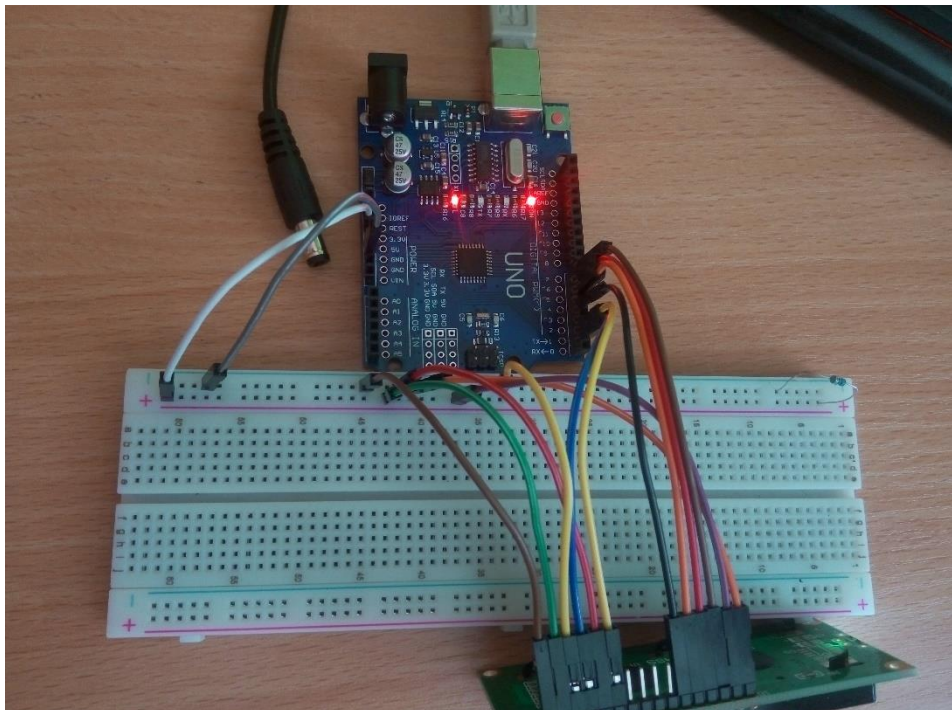
SEZNAM POUŽITÝCH INFORMAČNÍCH ZDROJŮ

- [1] Jak programovat Arduino [online]. [cit. 12. 7. 2019]. Dostupné z:
<https://www.youtube.com/watch?v=Lezw81Oph70>
- [2] How to program Arduino in VSCode (using Platform.io) [online]. [cit. 15. 7. 2019].
Dostupné z: https://www.youtube.com/watch?v=dany7ae_0ks
- [3] PIR Motion sensor with arduino [online]. [cit. 5. 7. 2015]. Dostupné z:
<https://www.youtube.com/watch?v=FxatDvs34mM>
- [4] Arduino LCD tutorial | How to control an LCD [online]. [cit. 9. 7. 2015]. Dostupné
z: <https://www.youtube.com/watch?v=dZZynJLmTn8>
- [5] Interface 4x3 & 4x4 Membrane Keypad with Arduino [online]. Dostupné z:
<https://lastminuteengineers.com/arduino-keypad-tutorial/>
- [6] Knihovna password.h [online]. [cit. 28. 9. 2020]. Dostupné z:
<https://github.com/trustcrypto/libraries/blob/master/password/password.h>
- [7] Knihovna password.h Arduino [online]. [cit. 5. 4. 2012]. Dostupné z: <https://playground.arduino.cc/Code/Password/>
- [8] Alarm security and alarm system project [online]. [cit. 30. 12. 2016]. Dostupné z:
https://www.youtube.com/watch?v=lTAo_H5eqsk
- [9] Web server using SPIFFS [online]. Dostupné z: <https://randomnerdtutorials.com/esp8266-web-server-spiiffs-nodemcu/>
- [10] Web interface IoTWebConf [online]. Dostupné z: <https://randomnerdtutorials.com/esp8266-web-server-spiiffs-nodemcu/>

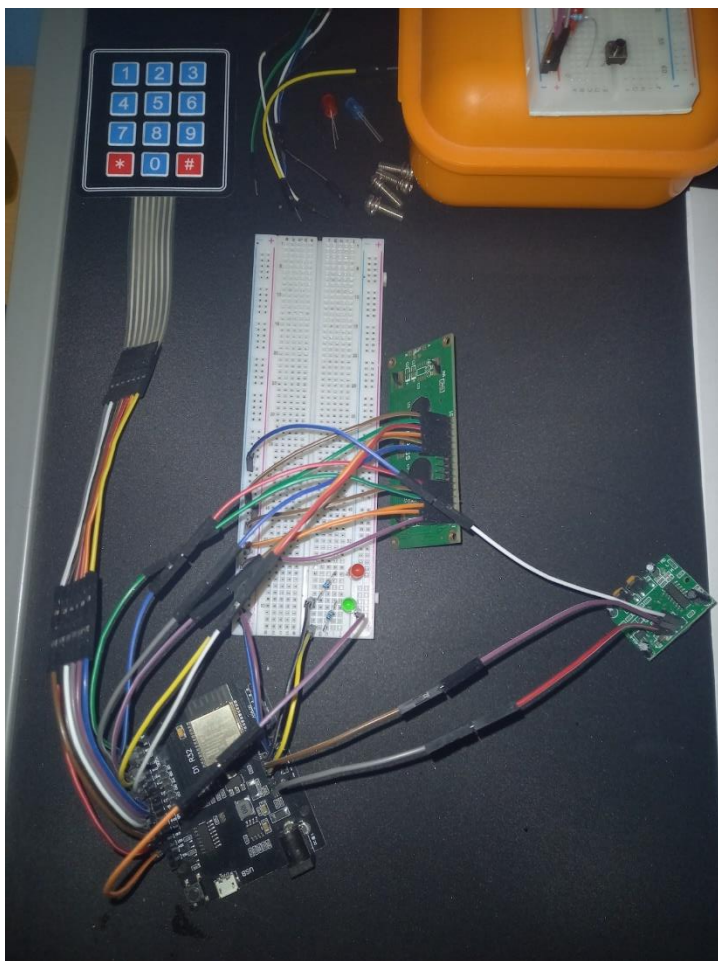
SEZNAM PŘÍLOH

č. 1 Fotodokumentace

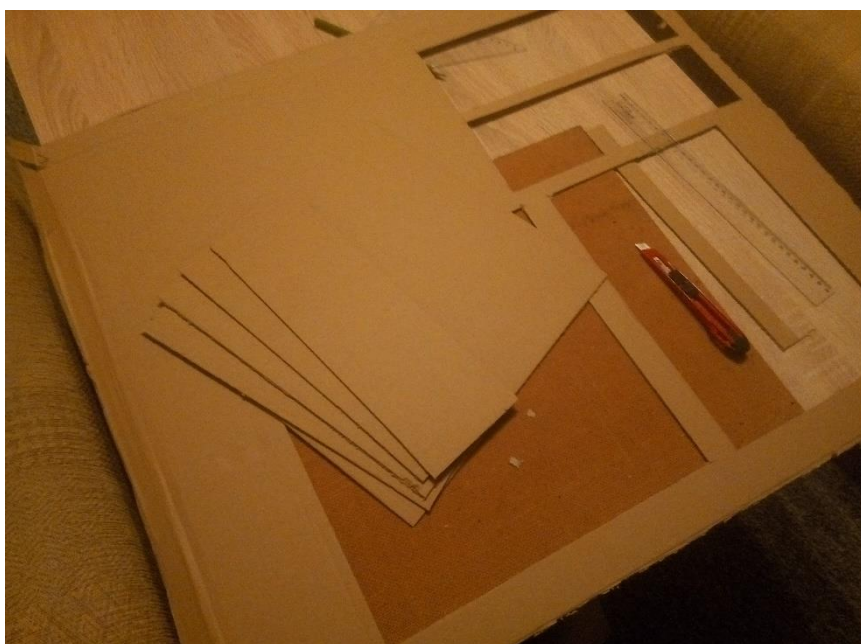
Příloha č. 1: Fotodokumentace



První verze displeje



Finální verze systému se všemi součástkami



Výroba modelu domu