Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Gašper Bavdek

Oddaljeni nadzor varnostnih sistemov z Raspberry Pi

diplomsko delo

univerzitetni študijski program
prve stopnje
računalništvo in informatika

Mentor: prof. dr. Branko Šter

Ljubljana, 2017

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Implementirajte oddaljen nadzor varnostnega sistema z uporabo računalnika na ploščici Raspberry Pi, ki se odlikuje z nizko ceno in precejšnjo prilagodljivostjo. Raspberry Pi naj deluje kot samostojni strežnik, do katerega lahko dostopamo oddaljeno preko varnih povezav. Njegova funkcija naj bo tudi pošiljanje obvestil odgovorni osebi o napaki ali okvari na varnostnem sistemu. Opišite sistem in preizkusite njegovo delovanje.

Zahvalil bi se rad mentorju dr. Branku Šteru za pomoč pri izdelavi diplomske naloge. Posebno pa bi se rad zahvalil vsej družini in vsem prijateljem, ki ste mi na kakršen koli način pomagali pri izdelavi diplomske naloge

Hvala vsem.

Kazalo

Seznam uporabljenih kratic

kratica	angleško	slovensko
TVO	technical protection of buildings	tehnično varovanje objektov
GPIO	general purpose input/output	splošnonamenski vhod/izhod
VPN	virtual private network	navidezno zasebno omrežje
FPA	fire panel	protipožarna centrala
ARM	Acorn RISC Machine	vrsta procesorjev
RAM	random access memory	bralno-pisalni pomnilnik
GPU	graphics processing unit	grafična procesna enota
HDMI	high-Definition Multimedia Interface	multimedijski vmesnik visoke ločljivosti
USB	universal Serial Bus	univerzalno serijsko vodilo
LAN	local area network	lokalno omrežje
SD	secure digital	pomnilniška kartica
NO	normally open	normalno odprti
NC	normally closed	normalno zaprti
COM	common	skupni
GND	ground	ozemljitev
SQL	structured query language	strukturiran povpraševalni jezik
DC	direct current	enosmerni tok
IP	internet protocol	internetni protokol
MAC	media access control	unikatna številka medija

Povzetek

Naslov: Oddaljeni nadzor varnostnih sistemov z Raspberry Pi

Avtor: Gašper Bavdek

Varnost je v današnjem svetu na prvem mestu. Eno od področij varnosti je tudi tehnično varovanje objektov. Varnostni sistemi, ki so vključeni v to panogo, so sistemi javljanja požara, kontrola pristopa, video nadzor in protivlomni sistem.

Problem, ki nastane pri izvajanju tehničnega varovanja, je nadzor centralnih naprav. Velikokrat pri okvari ali napaki na sistemu odgovorna oseba za samo napako ali okvare ni obveščena. To pomeni, da bi v primeru izrednega primera lahko sam sistem odpovedal in posledice so lahko zelo hude.

To poskušamo rešiti tako, da bomo varnostne sisteme nadzorovali oddaljeno z uporabo računalnika na ploščici Raspberry Pi. Raspberry Pi bomo uporabili predvsem zaradi ugodne cene in velike prilagodljivosti. Deloval bo deloval kot samostojni strežnik, do katerega bomo dostopali oddaljeno, preko varnih in zaščitenih povezav. Imel bo tudi funkcijo pošiljanja obvestil odgovorni osebi o napaki ali okvari na varnostnem sistemu.

Ključne besede: Raspberry Pi, varnostni sistemi, oddaljeni nadzor.

Abstract

Title: Remote control of security systems with Raspberry Pi

Author: Gašper Bavdek

Nowadays safety is of primary importance and one of its branches includes technical protection of facilities. Belonging to this branch are security systems, such as fire announcement systems, access control, video surveillance and security system.

The issue that arises in performing technical protection is the control of central installations. With breakdowns or malfunctions it is often the case that the person responsible is not informed about it, which may, in the event of an extreme situation, cause the system to fail where the consequences can be severe.

The resolution of this problem will involve controlling security systems remotely by means of the Raspberry Pi. Raspberry Pi will be used due to its reasonable price and great adaptability of the system. Raspberry Pi will function as an independent server, allowing remote access via safe and protected connections. In addition it will have the function of sending notifications regarding malfunctions or breakdowns in the security system to the person responsible.

Keywords: Raspberry Pi, security systems, remote control.

Poglavje 1

Uvod

V podjetju Iskra na oddelku TVO *(tehnično varovanje objektov)* se ukvarjamo s tehničnim varovanjem objektov. Varnostni sistemi, ki jih uporabljamo, so video nadzor, kontrola pristopa, protipožarni sistemi in protivlomni sistemi. Vsi sistemi potrebujejo zanesljiv nadzor, da v primeru napake naše ekipe hitro posredujejo in odpravijo napake na sistemu. Tako smo prišli na idejo, da bi varnostne sisteme nadzorovali oddaljeno, in sicer na sedežu našega podjetja.

S številnimi različnimi idejami in preizkusi smo ugotavljali, kako bi lahko ugodno, hitro in zanesljivo izvajali nadzor. Ugotovili smo, da bo najceneje in še vedno zanesljivo uporabljati računalnik na ploščici Raspberry Pi. Računalnik je cenovno ugoden, relativno enostaven za programiranje in nastavljanje konfiguracije, a vseeno dovolj majhen, da ga lahko vgradimo v samo ohišje varnostnega sistema, ki je po navadi varovano s stikalom. Ta v primeru nezakonitega vstopa v ohišje varnostnega sistema sproži alarm. Na sistem smo naložili strežnik Apache, ki ni potraten pri porabi računalniških virov. Tako lahko do samega sistema dostopamo s pomočjo spletnega brskalnika.

V času razvoja smo naleteli na težavo, kako ugotoviti, kdaj je sistem v napaki ali alarmu. Rešili smo jo tako, da smo na računalniku Raspberry Pi uporabili splošnonamenski vhod/izhod *(GPIO, general purpose input/o-*

utput), na varnostnem sistemu pa relejske izhode. Varnostnemu sistemu lahko nastavimo konfiguracijo, ki v primeru napake ali alarma preko relejskega modula pošlje signal na računalnik Raspberry Pi. Tudi Raspberry Pi lahko nastavimo tako, da v primeru prejema signal na določenem pinu GPIO ukaz sprejeme, ga prevede in izvrši ustrezno dodeljeno akcijo.

Največja težava je bila, kako vzpostaviti varno in zanesljivo povezavo z računalnikom Raspberry Pi. Odločili smo se, da uporabimo med Raspberry Pi in odjemalcem povezavo VPN *(virtual private network)*, saj imajo vsa večja podjetja, ki cenijo varnost povezav, že vzpostavljena omrežja VPN. Z vzpostavitvijo omrežja VPN zavarujemo podatke, do katerih nepooblaščene osebe ne smejo dostopati in tako priti do zaupnih podatkov. Glede varnosti moramo biti zelo pazljivi, navsezadnje gre za varnostne sisteme in v primeru nepooblaščenega vstopa v sistem so lahko posledice resne.

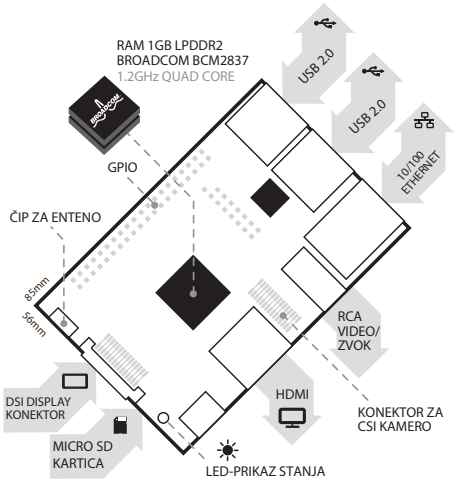
V drugem poglavju diplomskega dela smo opisali računalnik Raspberry Pi verzija 3 in njegove funkcionalnosti, ki so pomembne za naš projekt. V tretjem poglavju je opisana povezava med računalnikom Raspberry Pi in varnostnim sistemom. Kot primer smo vzeli protipožarno centralo Bosch FPA 5000 in delovanje sistema GPIO. V četrtem poglavju smo se posvetili opisu aplikacije. Zadnje poglavje pa je namenjeno varni povezavi in sistemu povezovanja VPN.

Poglavje 2

Raspberry Pi

2.1 Strojna oprema

Raspberry Pi 3 Model B je tretja generacija računalnikov na ploščici Raspberry Pi, izšel pa je leta 2016. Raspberry Pi je izdelek podjetja iz Velike Britanije. Raspberry Pi je v velikosti kreditne kartice in je uporaben na številnih področjih, kot so raziskovanje, avtomatizacija, učenje itd. Vsebuje veliko strojne opreme, ki jo lahko uporabimo v različne namene.



Slika 2.1: Strojna oprema Raspberry Pi 3

Raspberry Pi 3, prikazan na Sliki 2.1, poganja sistem na čipu Broadcom BCM2837 z integriranim procesorjem 1.2GHz quad-core ARM Cortex A53. Poleg procesorja ima še integriran pomnilnik v velikosti 1 GB LPDDR2. Procesor je 50 odstotkov hitrejši od njegovega predhodnika ARMv7 quad core, ki je bil vgrajen v Raspberry Pi 2.

Za grafični prikaz poskrbi GPU Dual core VideoCore IV, ki podpira Open GL ES 2.0, strojno pospešen OpenVG in slikovno kodiranje 1080p30 H.264.

Za priklop vhodno/izhodnih naprav vsebuje številne različne priklope in vmesnike. Do interneta lahko dostopamo s pomočjo brezžičnega vmesnika Wireless LAN 802.11 b/g/n ali Ethernet 10/100 BaseT. Za priklop zaslona ali katere koli druge naprave za prikaz slike vsebuje HDMI ali pa priključek DSI Display za priklop zaslona, občutljivega na dotik. Za prenos zvoka imamo na voljo priklop 3,5 mm jack. Zvok lahko prenašamo tudi preko priklopa HDMI ali USB, saj vsebuje kar štiri priklope USB 2.0. Vsebuje tudi GPIO, ki je za nas najbolj relevanten, saj z njim povezujemo nadzorni sistem na Raspberry Pi. Poleg tega imamo na voljo še 15-pinski priklop za kamero MIPI. Vsi podatki se shranjujejo na kartico SD, ki se nahaja na spodnjem delu Raspberry Pi.

2.1.1 GPIO

Za naš projekt je GPIO ena izmed glavnih stvari, ki jo uporabljamo na računalniku Raspberry Pi. Vsi signali, ki jih bomo dobili iz nadzornega sistema, bodo prejeti na GPIO in nato jih bo naša aplikacija ustrezno procesirala in izvedla ustrezen ukrep, glede na to kateri vhod bo aktiviran.

Na samem Raspberry Pi se nahaja 40 pinov, od tega jih je 26 GPIO, ostali pa so za napajanje 5V enosmerno (DC, direct current) ali 3V DC in ozemljitev. Razporeditev samih pinov vidimo na Sliki 2.2. Vsebuje pa tudi dva pina za ID EEPROM, ki sta namenjena programiranju trajnega pomnilnika. Same pine lahko preprosto sprogramiramo tako, da informacije pridobivamo ali pa jih pošiljamo. Vsak pin GPIO lahko sprogramiramo, da deluje kot vhod (input) ali pa izhod (output).

Če pin sprogramiramo kot izhod, deluje kot vir. Pinu lahko določimo vrednost HIGH ali LOW. Ko določimo vrednost HIGH, dobimo na pinu napetost 3,3V, ko pa mu določimo vrednost LOW, dobimo napetost 0V. Za pravilno delovanje priključene komponente moramo na samo komponento pripeljati tudi ozemljitev (ground).

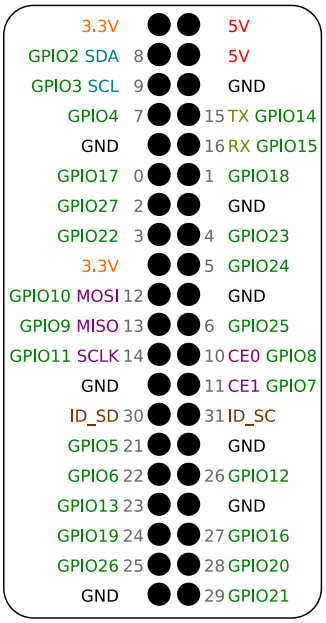
Če pin sprogramiramo kot vhod (input), pinu ne določamo vrednosti, ampak s pina odčitamo, ali je vrednost HIGH ali LOW. Tako pridobivamo podatke iz različnih senzorjev in komponent.

Za naš projekt smo določene pine sprogramirali, da delujejo kot vhod, pine oziroma releje na nadzornem sistemu pa tako, da delujejo kot izhod. Tako v primeru napake ali alarma nadzorni sistem pošlje izhod na računalnik Raspberry Pi, ki ga nato na vhodnem pinu sprejme.

Postopek za nastavitev pina kot vhod ali izhod:

```
import RPi.GPIO as GPIO
 # importiramo RPi.GPIO modul

GPIO.setmode(GPIO.BCM)
# izberemo BCM ali BOARD

GPIO.setup(port_or_pin, GPIO.IN)
# nastavimo pin kot vhod/input

GPIO.setup(port_or_pin, GPIO.OUT)
# nastavimo pin kot izhod/output

GPIO.output(port_or_pin, 1)
# nastavimo izhodno vrednost pina 1/HIGH/True

GPIO.output(port_or_pin, 0)
# nastavimo izhodno vrednost pina 0/LOW/False
```

```
i = GPIO.input(port_or_pin)
# preberemo vhodno vrednost pina in jo dodelimo spremenljivki i
```



Slika 2.2: GPIO pri Raspberry Pi 3

2.2 Programska oprema

Na Raspberry Pi lahko naložimo različne operacijske sisteme, kot so Raspbian, Ubuntu mate, Windows 10 IOT Core, OSMC, Libreelec, Pinet itd. Za naš projekt smo izbrali Raspbian, saj je po raziskavah najbolj ustrezal našemu projektu.

2.2.1 Raspbian

Raspbian je zastonjski operacijski sistem, ki temelji na sistemu Debian. Avtorji operacijskega sistema so razvijalci programske opreme, ki ni povezana s fundacijo Raspberry Pi. Raspbian je posebno optimiziran za programsko opremo Raspberry Pi.

Sestavljen je iz različnih osnovnih programov in pripomočkov, ki skrbijo za optimizirano delovanje Raspberry Pi. Poleg osnovnega sistema ponuja tudi do 35.000 različnih paketov, ki jih lahko namestimo poleg osnovnega sistema. Njegova posebnost je, da so njegove nastavitve prilagojene na optimizirano "hard float"programsko kodo. Izraz "hard float"pomeni, da aplikacija uporablja številne aritmetične operacije s plavajočo vejico.

Sama namestitev operacijskega sistema je izredno preprosta. Z uradne spletne strani si prenesemo sliko ISO operacijskega sistema, jo z ustreznim programskim orodjem zapišemo na mikro kartico SD in jo vstavimo v Raspberry Pi. Ob zagonu Raspberry Pi se mora sistem inicializirati/zagnati in sistem je pripravljen za delo.

Za naš projekt smo morali Raspbian najprej posodobiti in nanj nato namestiti določene programe in aplikacije, ki jih lahko nameščamo s pomočjo terminala, in sicer z ustreznimi ukazi.

Sprva posodobimo sistemsko listo paketov.

```
sudo apt-get update
```

Nato nadgradimo/posodobimo vse nameščene pakete na njihovo zadnjo verzijo.

```
sudo apt-get dist-upgrade
```

Za dostop do aplikacije iz oddaljenega računalnika moramo na Raspberry namestiti strežnik Apache 2.

```
sudo apt-get install apache2 -y
```

Zaradi varnega vpisa v sistemu bomo uporabili bazo sql, katero je potrebno namestiti v sistem.

```
sudo apt-get install mysql-server &&
sudo apt-get install mysql-client
```

Ker pa sem za svojo aplikacijo izbral spletni jezik PHP, smo potrebovali tudi ustrezen urejevalnik. Izbrali smo urejevalnik Bluefish.

```
sudo apt-get install bluefish
```

Zaradi lažjega programiranja in hitrejšega delovanja GPIO smo uporabili jezik Python. Python bo skrbel za nadzor nad GPIO in shranjeval podatke, medtem ko bo PHP poskrbel za prikaz teh podatkov.

Poglavje 3

Povezovanje Raspberry Pi s protipožarnim sistemom Bosch FPA 5000

Cilj diplomske naloge je bil uspešno povezati kateri koli nadzorni sistem, bodisi video nadzorni sistem, protivlomni sistem, protipožarni sistem, bodisi sistem kontrole pristopa z računalnikom Raspberry Pi 3.

Zaradi varnosti nadzornih sistemov se na njih ne bomo povezovali prek priklopa USB ali Ethernet. Z direktnim priklopom lahko konfiguriramo in programiramo nadzorni sistem, kar pomeni, da bi lahko nepooblaščena oseba spremenila nastavitve ali celo izklopila varnostni sistem. Priklop bo vezan na priklope, ki so zgolj vezani na rele, iz katerega lahko dobimo kontakt brez napetosti. Programiranje, konfiguracija in izklop sistema s pomočjo relejev niso mogoči, zaradi česar je sistem bolj varen.

V naslednjem podpoglavju je na kratko opisan protipožarni sistem Bosch FPA 5000, katerega smo uporabili kot testni nadzorni sistem. Opisana sta njegova osnovna zgradba in delovanje. Nato pa bomo opisali, kako sta povezana protipožarni sistem Bosch FPA 5000 in računalnik na ploščici Raspberry Pi 3.

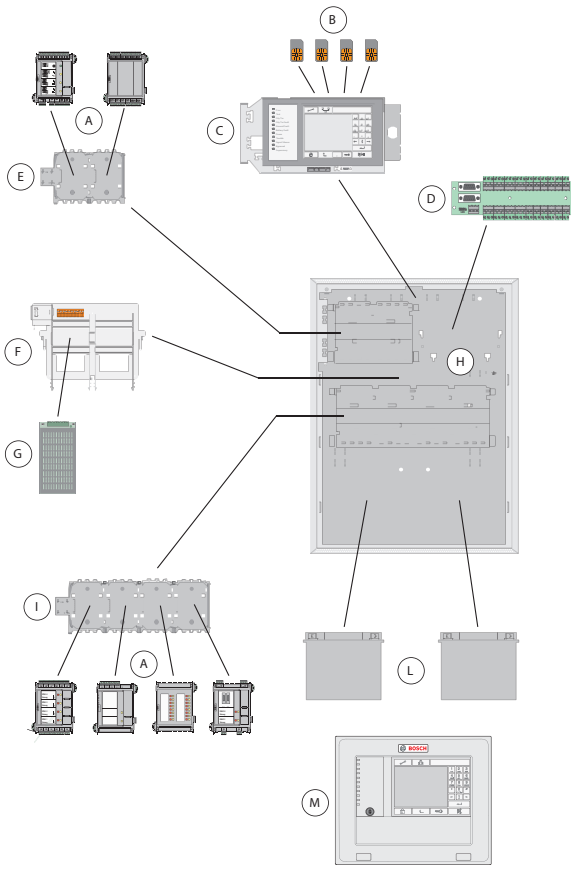
3.1 Bosch protipožarni sistem FPA5000

Za testni nadzorni sistem smo izbrali protipožarno centralo Bosch FPA 5000, ki je sestavljena iz različnih elementov. Ti elementi so požarna centrala, moduli, javljalniki požara, ročne tipke, sirene, najrazličnejši senzorji različnih plinov itd.

Bosch FPA-5000 je protipožarna centrala, Slika 3.1, namenjena uporabi v manjših in srednje velikih objektih. Omogoča priklop do 254 različnih elementov. Elementi pri tej centrali se povezujejo v krog ali tako imenovani "loop". Na centralo je mogoče priklopiti do 3 oddaljene tipkovnice, ki omogočajo lokalni nadzor nad centralo. Tako ni potrebno, da je centrala nameščena na mestu poleg pooblaščene osebe, ampak se za to uporabijo oddaljene tipkovnice. Za programiranje centrale moramo uporabljati poseben program, imenovan FSP-5000-RPS, ki je namenjen programiranju vseh požarnih central v tej seriji. Nova serija central omogoča avtomatsko detekcijo elementov, zaradi česar ni potrebno ročno vnesti vsakega elementa v sistem. Centrala je sestavljena modularno in glede na zahteve ji lahko dodajamo različne module.

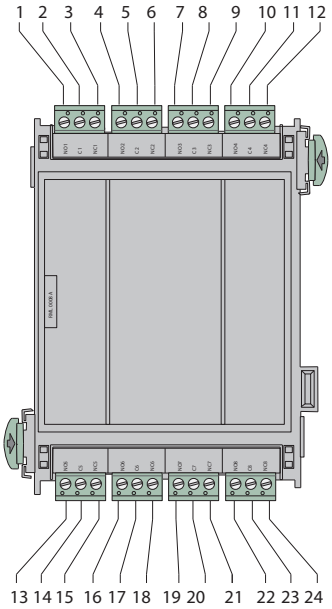
Za priklop na Raspberry Pi moramo v protipožarno centralo dodati nizkonapetostni relejski modul RML 0008 A, prikazan na Sliki 3.2. Modul ima osem kontaktnih relejev tipa C in jih lahko prosto programiramo.

Delovanje samega releja, kot prikazuje Slika 3.3, je zelo preprosto. Rele je elektromagnetno stikalo, ki ga upravljamo z napetostjo. Ima tri različne kontakte: NO *(normally open)* – normalno odprti kontakt, NC *(normally closed)* – normalno zaprti kontakt in skupni *(COM, common)*. Kontakti so v normalnem stanju, ko napetost ni prisotna, torej sta kontakta COM in NC sklenjena. Ko je napetost prisotna, tuljava ustvari elektromagnetno silo, ki sklene kontakt COM s kontaktom NO.



A	Moduli	G	Napajalna enota
B	Licenčni ključi	H	Ohišje (HCP 0006 A)
C	Panelni kontroler	I	Panelni nosilec - dolgi
D	Distributor	L	Baterije
E	Panelni nosilec	M	Oddaljena tipkovnica
F	Nosilec napajalne enote (Vgrajeno v ohišje)		

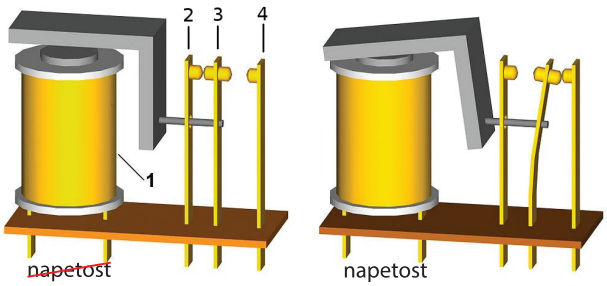
Slika 3.1: Bosch FPA 5000



Poz.	Opis	Konektor
1-3	NO1 / C1 / NC1	Rele 1
4-6	NO2 / C2 / NC2	Rele 2
7-9	NO3/C3/NC3	Rele 3
10-12	NO4/C4/NC4	Rele 4
13-15	NO5/C5/NC5	Rele 5
16-18	NO6/C6/NC6	Rele 6
19-21	NO7/C7/NC7	Rele 7
22-24	NO8/C8/NC8	Rele 8

NO - Normaly open - normalno odprti
NC - Normaly closed - normalno zaprti

Slika 3.2: Relejski modul RML 0008 A



1 - Elektromagnetna tuljava
2 - NC - normalno zaprti kontakt
3 - COM
4 - NO normalno odprti kontakt

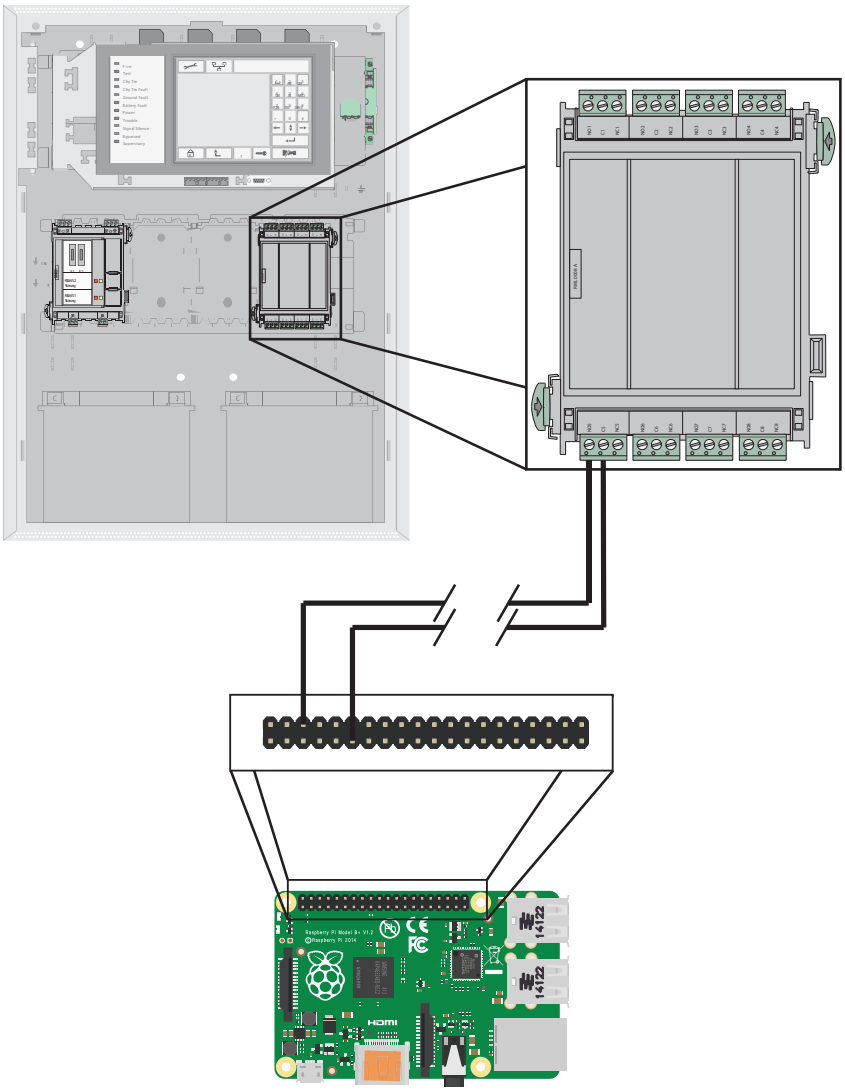
Slika 3.3: Rele

3.2 Povezava Bosch FPA5000 in Raspberry Pi 3

V prejšnjem poglavju smo razložili, kako moramo nastaviti varnostni sistem, da je pripravljen na povezavo z Raspberry Pi. Potrebno je tudi ustrezno nastaviti GPIO na računalniku Raspberry Pi. Za priklop smo v našem primeru izbrali pine GPIO 17, GPIO 24 in GPIO 27, ki ponazarjajo tri različna stanja, ki jih bo centrala poslala na Raspberry Pi. Za vsako različno stanje smo centrali dodelili različen relejski izhod.

Za povezavo relejskega izhod in pina GPIO potrebujemo dva električna prevodnika. Prvega povežemo na kontakt NO ali normalno odprti kontakt na releju in ustrezen pin GPIO. Drugega pa povežemo na skupni kontakt na releju in pin GND za ozemljitev na računalniku Raspberry Pi. Vezalna shema je prikazana na Sliki 3.4.

Če povzamemo: ko se zgodi nek dogodek v varnostnem sistemu, centrala ta dogodek zazna. Program v centrali ta dogodek ustrezno prevede in določi, katera akcija se mora sprožiti, v našem primeru pošlje podatek, kateri rele naj se sklene na relejskem modulu. Modul nato pošlje napetost na ustrezni rele, rele pa sklene ustrezna kontakta. To spremembo zazna računalnik Raspberry Pi, ki izvede ustrezen ukrep, ki smo ga sprogramirali.



Slika 3.4: Povezava med Raspberry Pi in FPA 5000

Poglavje 4

Aplikacija za nadzor varnostnega sistema

Z vzpostavitvijo uspešne povezave med računalnikom Raspberry Pi in varnostnim sistemom lahko začnemo nadzorovati varnostni sistem. V ta namen smo razvili posebno aplikacijo, ki omogoča vpogled v to. V aplikaciji je možno spremljati trenutno stanje sistema, omogoča pa tudi vpogled v pretekle dogodke, ki so se zgodili v sistemu. V prvem podpoglavju sledi razlaga, na kakšnih temeljih stoji aplikacija, in opis kode. V drugem podpoglavju sta opisana delovanje in uporaba aplikacije s priloženimi slikami.

4.1 Zgradba in kodiranje

Aplikacija je zgrajena iz dveh delov, in sicer iz dela, ki zbira podatke in jih shranjuje, in dela, ki je namenjen za prikazovanje podatkov. Del, ki shranjuje podatke, smo napisali v jeziku Python, saj zagotavlja lažje programiranje s sistemom GPIO. Za prikazovanje podatkov smo razvili spletno aplikacijo v jezikih HTML in PHP.

Razlog za ločitev je bil, da mora del kode, ki zbira podatke, delovati neprekinjeno, saj mora stalno nadzirati, kakšno stanje pošilja varnostni sistem.

Najprej si bomo ogledali, kako je koda za zbiranje podatkov spisana v programskem jeziku Python. V prvem delu kode smo uvozili vse potrebne knjižnice za delovanje GPIO ter knjižnico za pridobitev ure in datuma. Datum in uro potrebujemo za to, da v primeru dogodka lahko shranimo točne podatke, kdaj se je dogodek zgodil. Te zapišemo v datoteko, kot bomo videli v kasneje v kodi.

```
import RPi.GPIO as GPIO
import time
from datetime import datetime
```

Nato smo s funkcijo GPIO.setup določili, katere pine GPIO bomo uporabljali in jim določili funkcijo, da delujejo kot vhod:

```
GPIO.setmode(GPIO.BCM)
#BCM - stevilka pina GPIO - ?
GPIO.setup(17, GPIO.IN, pull_up_down=GPIO.PUD_UP)
#GPIO 17 vhod - input
GPIO.setup(27, GPIO.IN, pull_up_down=GPIO.PUD_UP)
#GPIO 27 vhod - input
GPIO.setup(22, GPIO.IN, pull_up_down=GPIO.PUD_UP)
#GPIO 22 vhod - input
```

Koda, ki sledi, skrbi za nenehno preverjanje stanja na pinih GPIO 17, 22 in 27. V primeru, da pin dobi ustrezno stanje, kar pomeni, da je rele sklenjen, ta izvrši ustrezno proceduro.

Program sprva preveri staro stanje gumba, kar smo naredili, ker rele v primeru napake vseskozi drži določeno stanje. Če program ne bi preveril prejšnjega stanja, bi tekom ohranjanja stanja vsako sekundo shranil trenutno stanje, tako pa to naredi samo ob spremembi. Nato se mora stanje resetirati, če želi shraniti novo stanje.

Nato program odpre tekstovno datoteko in v njo zapiše datum in stanje, ki je odvisno od tega, na kateri pin smo dobili kontakt.

Koda:

```
import RPi.GPIO as GPIO
import time
from datetime import datetime

stanj17=0
stanje27=0
stanje22=0

GPIO.setmode(GPIO.BCM)

GPIO.setup(17, GPIO.IN, pull_up_down=GPIO.PUD_UP)
GPIO.setup(27, GPIO.IN, pull_up_down=GPIO.PUD_UP)
GPIO.setup(22, GPIO.IN, pull_up_down=GPIO.PUD_UP)

try:
    while True:
        gpio17 = GPIO.input(17)
        gpio27 = GPIO.input(27)
        gpio22 = GPIO.input(22)

        if gpio17 == False:
            if stanje17==0:
                txt = open ("eventlog.txt","a")
                txt.write(datetime.now().strftime
                ('%Y-%m-%d %H:%M:%S')+"
                Napaka na sistemu\n")
                txt.close()
                time.sleep(1)
                stanje17=1
        else:
```

```
                stanje17=0

        if gpio27 == False:
            if stanje27==0:
                txt = open ("eventlog.txt","a")
                txt.write(datetime.now().strftime
                ('%Y-%m-%d %H:%M:%S')+"
                Opozorilo na sistemu\n")
                txt.close()
                time.sleep(1)
                stanje27=1
        else:
            stanje27=0

        if gpio22 == False:
            if stanje22==0:
                txt = open ("eventlog.txt","a")
                txt.write(datetime.now().strftime
                ('%Y-%m-%d %H:%M:%S')+"
                Potrebna zamenjava baterij\n")
                txt.close()
                time.sleep(1)
                stanje22=1
        else:
            stanje22=0
except:
    GPIO.cleanup()
```

Program, ki zbira podatke o sistemu, smo nastavili tako, da se zažene ob vsakem vklopu računalnika. V primeru ponovnega zagona računalnika programa ni potrebno vsakič ročno zagnati.

Vsi pretekli dogodki, ki so se zgodili, se shranjujejo v tekstovno datoteko, imenovano eventlog.txt. Podatki so zapisani v naslednji obliki: datum, ura in vrsta opozorila.

```
2018-02-02 14:33:29 Opozorilo na sistemu
2018-02-03 19:41:22 Potrebna zamenjava baterij
2018-02-09 23:02:50 Napaka na sistemu
```

Za sistem, ki skrbi za prikazovanje podatkov in temelji na programskih jezikih PHP in HTML, smo naložili strežnik Apache2, ki skrbi, da lahko oddaljeno dostopamo do vsebin.

Sprva smo morali poskrbeti za varnost, zato smo oblikovali stran za prijavo v sistem. Tukaj uporabnik vpiše svoje uporabniško ime in geslo, s katerim si pridobi pravice za vpogled v sistem.

Ker smo hoteli varen sistem vpisa, smo uporabili bazo SQL, ki hrani podatke o uporabnikih, ki lahko dostopajo do sistema. Za bazo sql smo uporabili MariaDB. Za začetek sem ustvarili podatkovno bazo *varen_vpis*. V podatkovno bazo smo dodali novo tabelo *user* in v njo zapisali novega uporabnika.

Za dostop do baze lahko uporabimo uporabnika *root*, ampak zaradi varnosti smo ustvarili novega in mu dodelili pravice za dostop do baze.

```
CREATE DATABASE `varen_vpis`;

CREATE TABLE IF NOT EXISTS `users` (
`id` int(11) NOT NULL AUTO_INCREMENT,
`username` text NOT NULL,
`password` text NOT NULL
)

INSERT INTO users SET username='user', password='password'
```

Za dostop do baze z jezikom PHP smo sprva spisali konfiguracijsko datoteko, ki vsebuje potrebne podatke za uspešno povezavo z bazo *mysql*, po kateri lahko iščemo.

```
<?php
/* Podatki za povezavo na bazo
server with default setting (user 'root' with no password) */
define('DB_SERVER', 'localhost');
define('DB_USERNAME', '******');
define('DB_PASSWORD', '********');
define('DB_NAME', 'varen_vpis');

/*Poizkus povezave na podatkovno bazo */
$db = new mysqli(DB_SERVER, DB_USERNAME, DB_PASSWORD, DB_NAME);

// Preverimo povezavo
if($db === false){
    die("ERROR: Could not connect. " . mysqli_connect_error());
 }
?>
```

Nato sem oblikoval vpisno stran, kjer uporabnik vpiše svoje uporabniško ime in geslo. V primeru pravilnega uporabniškega imena in gesla ga preusmerimo na stran sistema, v nasprotnem primeru pa ga stran opozori, da vpisani podatki niso pravilni.

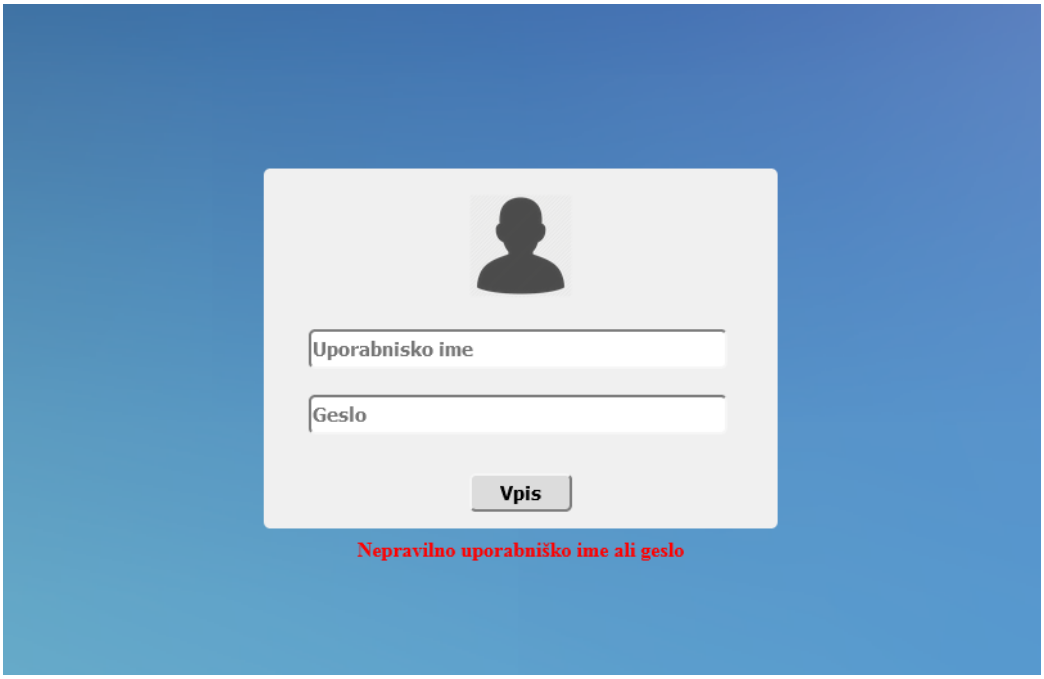
```
<?php
    /* vkljucimo konfiguracijsko datoteko z podatki
    include("config.php");
    session_start();
    /* ko uporabnik pritisne gumb vpis zacnemo
     z preverjanjem podatkov
    if($_SERVER["REQUEST_METHOD"] == "POST") {
       // username and password sent from form
```

```
        $myusername = mysqli_real_escape_string
        ($db,$_POST['user']);
        $mypassword = mysqli_real_escape_string
        ($db,$_POST['pass']);

        $sql = "SELECT username FROM users WHERE username =
        '$myusername' and password = '$mypassword'";
        $result = mysqli_query($db,$sql);
        $row = mysqli_fetch_array($result,MYSQLI_ASSOC);
        $active = $row['active'];

        $count  = mysqli_num_rows($result);
        if($count == 1) {
            header("location: home.php");
        }else {
            $error = "Napacno uproabnisko ime
            ali geslo";
        }
    }
?>
```

4.2 Uporabniški vmesnik aplikacije

Uporabniški vmesnik smo sestavili tako, da je čim bolj prijazen do uporabnika in čim bolj preprost zaradi omejenih računalniških virov.

Prva stran je vpisna stran, Slika 4.1, kjer mora uporabnik vnesti svoje uporabniško ime in geslo. V primeru nepravilnega gesla uporabnika sistem opozori in zahteva ponovni vpis. Dokler uporabnik ne vnese vseh parametrov pravilno, ne bo mogel dostopati do drugih strani. Namreč, ko se uporabnik uspešno vpiše, se ustvari seja, ki se tekom brskanja po spletni strani preverja. Kot smo že omenili, v ozadju poteka postopek preverjanja uporabnika z uporabo baze Maria DB SQL.



Slika 4.1: Vpis v sistem

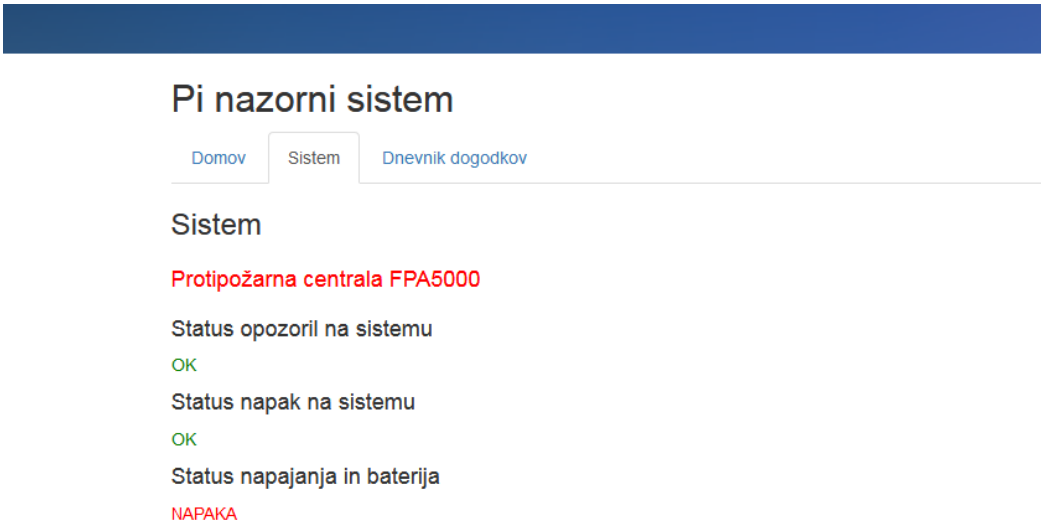
Po uspešni prijavi stran uporabnika preusmeri na domačo stran. Na domači strani se prikaže zavihek, kjer je naslov in objekt, v katerem se nahaja varnostni sistem. Naveden je tudi podatek, za kateri varnostni sistem gre in na katero strojno verzijo je sistem posodobljen. Na voljo imamo tudi še zavihek *sistem*, kjer je trenutno stanje sistema in pa zavihek *dnevnik dogodkov*, kjer je prikazan seznam vseh dogodkov.

Na prikazani Sliki 4.2 lahko vidimo, kakšna je domača stran, kjer je navedena lokacija varnostnega sistema in točno določen tip, kar je v našem primeru Bosch FPA5000.

Slika 4.2: Domača stran

Pod zavihkom *sistem* se odpre nov pogled, Slika 4.3, kjer lahko preverimo trenutno stanje varnostnega sistema. V primeru, da je določeno stanje brez opozorila ali napake, se pod statusom izpiše *OK*, ki je obarvan z zeleno barvo. V primeru napake pa se pod statusom izpiše *NAPAKA*, ki je obarvana z rdečo barvo.

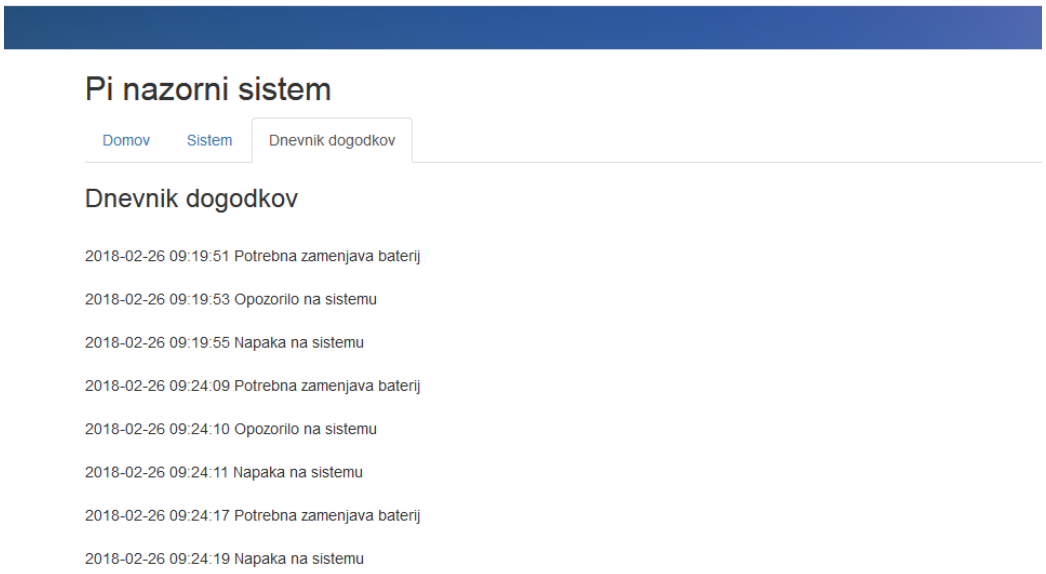
Na ta način lahko uporabnik vidi, ali je varnostni sistem še vedno v napaki ali ne, kajti iz dnevnika dogodkov ni mogoče razbrati, ali je napaka še vedno prisotna.



Slika 4.3: Prikaz stanja sistema

Pod *zavihkom dnevnik* dogodkov se odpre seznam vseh dogodkov, Slika 4.4, ki so se zgodili na varnostni centrali in so bili poslani na Raspberry Pi. Vsaka napaka ali opozorilo je zapisano v svoji vrstici. Zapisano je v naslednji obliki: najprej je naveden datum napake, nato ura napake in na koncu vrsta napake.

S tem ima uporabnik zagotovljen celovit pregled nad zgodovino dogodkov, ki jih je poslal varnostni sistem.



Slika 4.4: Prikaz dnevnika dogodkov

Poglavje 5

Varna povezava

Sistem za nadzor varnostnih sistemov potrebuje zanesljivo varnost, kajti podatki o stanju in preteklih dogodkih se prenašajo preko internetne povezave. V kolikor se podatki prenašajo preko interneta, so tako tudi posledično v nevarnosti, da jih prestreže druga nepooblaščena oseba. Taki podatki v nepravih rokah lahko privedejo do katastrofalnih posledic.

Moja raziskava je najprej tekla v smeri, da bi imeli na podjetju centralni strežnik. Sprva smo hoteli vzeti za strežnik kar Raspberry Pi, vendar smo po kratki analizi spoznali, da ima Raspberry Pi 3 premalo računalniških virov za opravljanje takega dela. Zato smo uporabili drugačen računalnik, ki ima veliko več prostih računalniških virov.

Začeli smo raziskovati, kako varno vzpostaviti povezavo med našim strežnikom in oddaljenim računalnikom Raspberry Pi. Začeli smo razmišljati, da bi podatke ustrezno kriptirali in jih ustrezno pretvorili nazaj. Ugotovili smo, da če bi hoteli izvesti ustrezno kriptiranje podatkov, bi porabili veliko računalniških virov in časa, zato smo to možnost opustili. Prav tako smo ugotovili, da nekateri protivirusni programi ponujajo možnost varnega povezovanja, vendar so te možnosti cenovno predrage za ta namen.

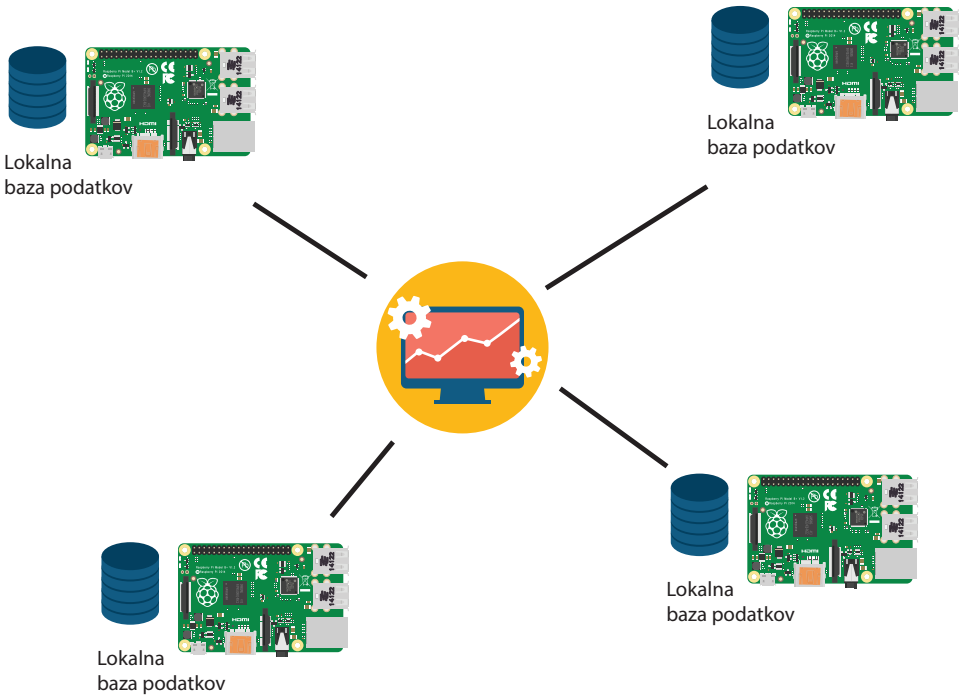
Kmalu smo ugotovili, da veliko naših podjetij uporablja varno povezavo VPN *(virtual private network)*. Zato smo se odločili, da jo preizkusimo. Za vzpostavljanje povezave VPN obstajajo tudi zastonjske verzije, ki so neko-

lio slabše kot plačljive. Torej bi morali za zelo dobro zaščito kupiti licence za namestitev aplikacije na strežnik in oddaljene Raspberry Pi. Veliko teh aplikacij je namenjenih bolj za uporabo na platformi Windows, Raspberry Pi pa, kot smo že omenili, deluje na sistemu Debian, ki je osnovan na platformi Linux. Do zapleta pride pri uporabi dveh različnih platform.

Sistem centralizacije smo opustili in izbrali porazdeljen sistem, kot prikazuje Slika 5.1. Porazdeljen sistem pomeni, da do vsakega računalnika Raspberry Pi dostopamo ločeno. Podatki se prenašajo samo takrat, ko želimo dostopati do podatkov, ne pa neprekinjeno, kar se zgodi pri centraliziranem sistemu. Zaradi tega je možnost za napad veliko manjša. Problem varne povezave pa kljub temu ostaja.

Seveda se tukaj poraja vprašanje, ali lahko kdo z napadom vstopi neposredno v sistem in s tem pridobi dostop do datotek s podatki. To bi lahko rešili z ustrezno konfiguracijo povezave VPN. Na primer, dostop do sistema samo z določenim IP-naslovom in MAC-naslovom, ustrezna zaščita z gesli itd.

Za fizični dostop je v podjetjih poskrbljeno. Komunikacijski prostori so ustrezno zaščiteni, tako fizično kot tudi z varnostnimi sistemi, zaradi česar nepooblaščena oseba zelo težko vstopi v komunikacijske prostore.



Slika 5.1: Porazdeljen sistem

Poglavje 6

Zaključek

Uspelo nam je razviti delujoč sistem za oddaljeni nadzor varnostnih sistemov, ki smo ga v podjetju že dolgo časa želeli razviti. Sistem deluje zadosti dobro, brez večjih napak, da ga začnemo uporabljati v našem poslu. Sam razvoj je bil vse prej kot enostaven. Veliko je bilo raziskovanja, kako naj razvijamo, centralizirano ali decentralizirano, veliko vprašanj pa se je porajalo tudi okoli varnosti.

Največje vprašanje pa je bilo seveda, kako povezati sistema med seboj, da bosta lahko komunicirala, saj različni sistemi uporabljajo različne protokole. Pri tem projektu smo ugotovili, da je to mogoče s pomočjo relejev. Vsi sistemi imajo neko možnost priklopa relejskega modula, s čimer si zagotovimo univerzalni sistem, ki ga lahko povežemo na skoraj vse varnostne sisteme.

Sistem smo zasnovali tako, da ga lahko razširimo. Dodali bi mu lahko tudi kake senzorje/tipala, ki merijo temperaturo in vlago, senzorje proti fizičnemu vdoru itd.

Sistem ima tako prednosti kot tudi slabosti v primerjavi z ostalimi izdelki, ki jih ponujajo proizvajalci varnostnih sistemov. Prednost je vsekakor ugodna cena in prosto programiranje, slabosti pa najdemo predvsem v manjšem številu funkcij, kot jih omogočajo ostalimi izdelki. Kljub temu menimo, da smo z razvojem uspešnega izdelka dosegli cilj diplomske naloge.

Slike

Literatura

[1] Bosch data sheets. Dosegljivo: https://emea.boschsecurity.com/en/products_3/firealarmsystemsen54_16/fpa5000modularfirepanel_15/fpa5000modularfirepanel_15_43#section_3869057931. [Dostopano: 10. 12. 2017].

[2] Marija DB - dokumentacija. Dosegljivo: https://mariadb.org/learn/. [Dostopano: 5. 1. 2018].

[3] PHP - dokumentacija. Dosegljivo: http://php.net/docs.php. [Dostopano: 14. 10. 2017].

[4] Python - dokumentacija. Dosegljivo: https://www.python.org/doc/. [Dostopano: 20. 11. 2017].

[5] Raspberry Pi 3 - dokumentacija. Dosegljivo: https://www.raspberrypi.org/documentation/. [Dostopano: 17. 11. 2017].

[6] VPN - How VPN works. Dosegljivo: https://technet.microsoft.com/en-us/library/cc779919(v=ws.10).aspx. [Dostopano: 11. 10. 2017].

[7] Russel Barnes. *Raspberry Pi projects book.* Liz Upton, 2015.

[8] A. Cox. *The Ultimate Raspberry Pi Handbook 2016.* 2016.