



Fakultet elektrotehnike i računarstva, Sveučilišta u Zagrebu
Zavod za elektroničke sustave i obradu informacija
Sveučilište u Zagrebu

Sustav za videonadzor



- △ Tekst je namijenjen svima
- △ Osnove poznavanja rada URS-a
- △ Razvoj programske podrške na Beaglebone Black Linux razvojnoj platformi

△

Sažetak

Projekt pametne kuće predstavlja jedinstven skup projekatnih zadataka studenata FER-a koji su se odlučili slušati i položiti kolegij Sustavi za praćenje i vođenje procesa. Pod idejom pametne kuće objedinjuju se razni projekti čija je osnovna ideja poboljšanje i unaprijeđenje života ukućana, ali i rješavanje problema svih onih kojima je potrebna pomoć pri u svakodnevnom životu.

U ovom radu se razmatra projekt koji svoju kontribuciju u cijelovit projekt pametne kuće daje sa aspekta sigurnosti. Radi se o sustavu videonadzora, kodnog imena *G.O.O.M.H. project* (Get Out Of My House).

Sadržaj

1. UVOD.....	3
2. REALIZACIJA SUSTAVA.....	4
2.1. Generalni opis sustava.....	4
2.2. Hardware.....	5
2.2.1. BeagleBone Black.....	5
2.2.2. Motori i upravljanje motorima.....	7
2.2.3. Sirena.....	9
2.3. Software.....	10
2.3.1. C++ aplikacija.....	11
2.3.2. Node JS aplikacija.....	12
2.4. Realizacija sustava i poboljšanja.....	14
3. ZAKLJUČAK.....	16
4. LITERATURA.....	17
5. POJMOVNIK.....	18

Ovaj seminarski rad je izrađen u okviru predmeta „Sustavi za praćenje i vođenje procesa“ na Zavodu za elektroničke sustave i obradbu informacija, Fakulteta elektrotehnike i računarstva, Sveučilišta u Zagrebu.

Sadržaj ovog rada može se slobodno koristiti, umnožavati i distribuirati djelomično ili u cijelosti, uz uvjet da je uvijek naveden izvor dokumenta i autor, te da se time ne ostvaruje materijalna korist, a rezultirajuće djelo daje na korištenje pod istim ili sličnim ovakvim uvjetima.

1. Uvod

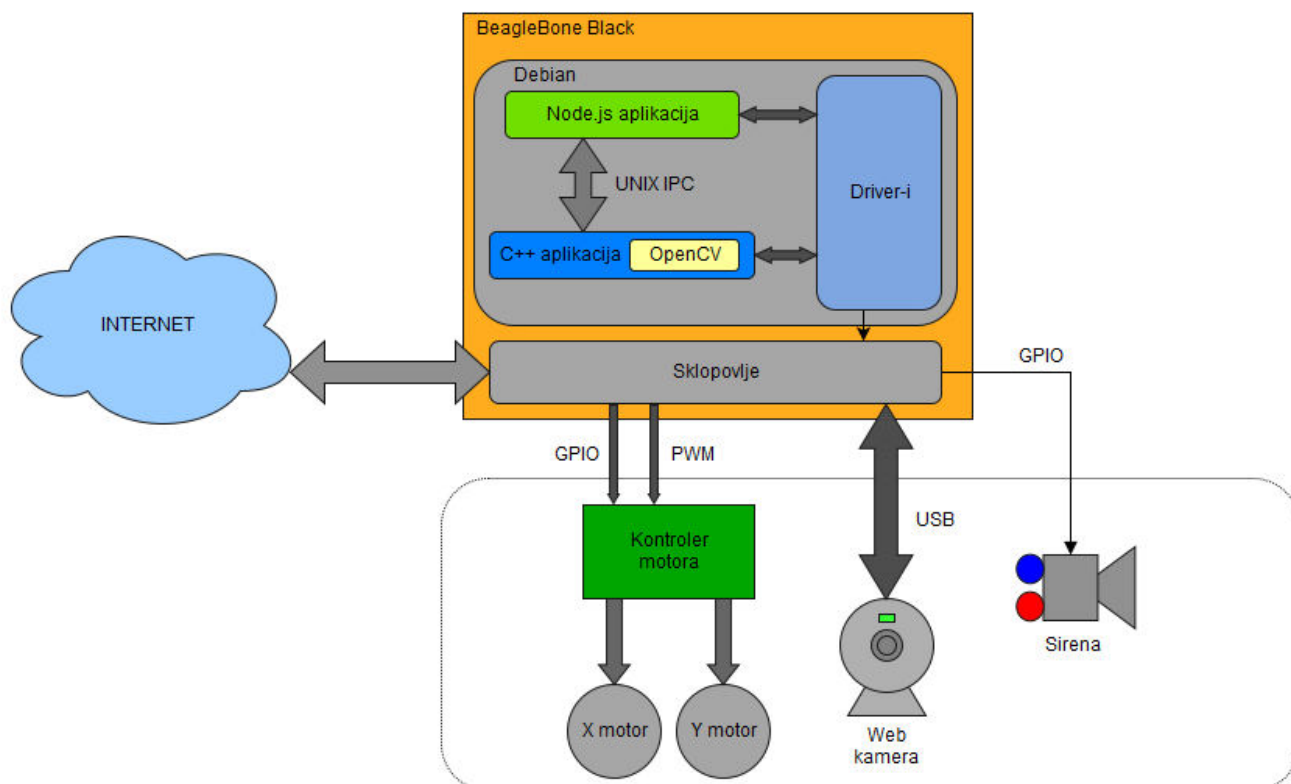
Jedan od povijesno nametnutih nagona koji se javlja u svakodnevnom ljudskom djelovanju jest nepovjerenje koje je pojedinac u stanju iskazati prema drugima, ali i šteta koju je spreman nanijeti drugima. U ovakvom svijetu sigurnost i zaštita pojedinca i njegovih materijalnih dobara od krađe i štete predstavlja jedinstvenu inženjersku problematiku (i nadasve veoma unosnu granu ljudske, a ne samo inženjerske djelatnosti). Jedan od pristupa na kojemu se posljednjih 30-ak godina zasnivaju *state of the art* sigurnosni sustavi jest videonadzor objekata koji se žele zaštititi. Ovo je pak u početku zahtjevalo posebne sobe i osoblje koje bi pratilo što se događa sa video kamerama, pa je ovo bilo namijenjeno zaštititi državnih i korporativnih objekata ili pak osobama čije su djelatnosti dovele do pojačane paranoje i u istu ruku mu priuštile ovakve mjere sigurnosti. Pojavom i razvojem internetskih tehnologija kontrola i nadzor nad ovakvim sustavima počeo se vršiti *remotely*, sa udaljenih mjesta. Činjenica je ostala dan danas da ćemo za ovakve sustave trebati izdvojiti pozamašnu količinu novaca.

Ovdje je predstavljen sustav koji bi trebao, korištenjem i povezivanjem već poznatih i testiranih tehnologija, vrlo jeftino omogućiti videonadzor bilo kojeg objekta, te integraciju više ovakvih sustava u jednu cijelinu kompletno upravljivu i konfigurabilnu sa udaljenih mjesta. Fokus je na niskoj cijeni i skalabilnosti sustava što će omogućiti korištenje sustava širem sloju pojedinaca.

2. Realizacija sustava

2.1. Generalni opis sustava

Na slici 1 je prikazana idejna blok shema sustava za videonadzor.



Slika 1: Blok shema sustava

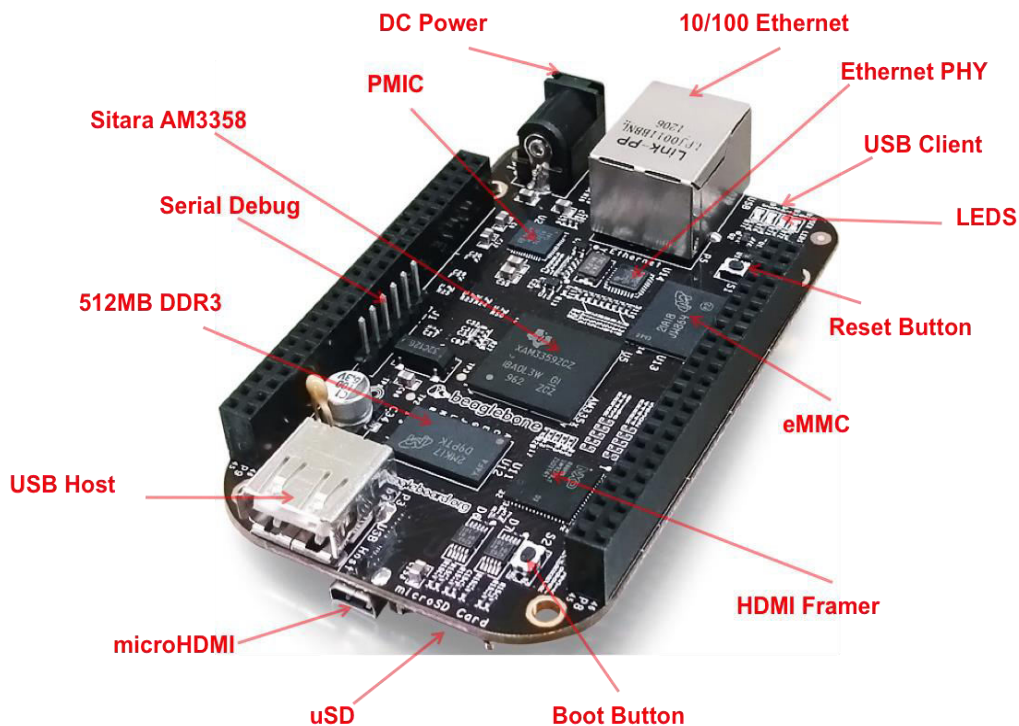
Okosnicu cijelog sustava za videonadzor predstavlja BeagleBone Black Linux platforma kojoj je omogućen pristup Internetu (ili lokalnoj mreži). Na njoj se izvršava sva programska podrška potrebna za funkcioniranje sustava, poput Node.js serverske aplikacije zaduženu za primanje i slanje HTTP paketa, upravljanje web soketima, MongoDB bazom podataka, te komunikaciju sa C++ aplikacijom. C++ aplikacija je zadužena za obradu slike sa web kamere, prosljeđivanje slike preko *IPC pipe*-a Node.js aplikaciji, te upravljanje motorima.

Sklopovlje koje čini cijeli sustav se sastoji od dva DC motora, web kamere i sirene. Aluminijska konstrukcija povezuje ove komponente u jednu funkcionalnu cijelinu za čije upravljanje je zadužena BBB platforma.

2.2. Hardware

2.2.1. BeagleBone Black

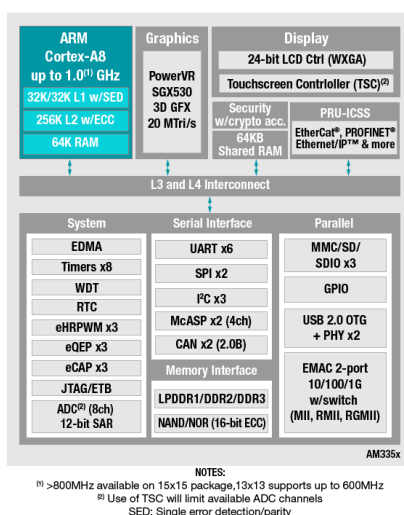
BeagleBone Black je čitav računalni razvojni sustav niske cijene. BBB je veličine kreditne kartice i opremljen je AM335x 1GHz ARM Cortex-A8 aplikativnim procesorom tvrtke Texas Instruments. BBB raspolaže sa 512 MB DDR3 RAM memorije, te 4 GB 8-bitne eMMC *flash* memorije, te koristi 3D grafički akcelerator i NEON *floating-point* akcelerator. Također, BBB razvojni sustav sadrži dva 2 PRU 32-bitna mikrokontrolera. Korisniku sustava su na raspolaganju USB konektor za spajanje USB kompatibilne periferije, mini USB konektor za ostvarivanje veze sa samim razvojnim sustavom i napajanje, HDMI priključak za monitor, Ethernet priključak, utor za SD karticu, te ono što ga razlikuje od standardnog PC-ja 2 46 pinska *header* pomoću kojeg je moguće na BBB razvojnu platformu priključiti bilo kakav hardver i ostvariti sušeljje prema njemu. Također, tu je i DC priključak na koji se može priključiti vanjski 5V adapter za aplikacije koje vuku veću struju.



Slika 2: BeagleBone Black razvojna platforma

BeagleBone Black razvojna platforma namijenjena je korištenju sa Linux operacijskim sustavom, a neke od Linux distribucija za koje postoji izrazita programska podrška namijenjena za implementaciju na BBB platformi su Ubuntu, Angstrom i naravno, Debian.

BBB platforma je odabrana za ovaj projekt zbog svoje niske cijene, malih dimenzija, kompatibilnosti sa već postojećim programskim solucijama i dostupnosti podrške među zajednicom koja već dulje vrijeme koristi ovu platformu. Također Sitara AM335x processor koji koristi BBB nudi velik broj različitih periferija, među kojima je i eHRPWM kontroler koji se koristi za upravljanje motorima PWM načinom rada, te naravno velik broj GPIO priključaka koji su am dostupni.



Slika 3: Blok shema AM335x korištenog u BBB

Sustav predviđa spajanje na postojeću mrežnu infrastrukturu objekta u kojemu se implementira njegova funkcija, te korištenje web kamere. Ovo je pak moguće jer BBB platform nudi Ethernet priključak, te USB priključak na kojeg je moguće spojiti bilo kakvu USB periferiju kao npr. Web kameru ili *WiFi dongle*.

BBB je opremljen TPS65217 PMIC (*Power Management Integrated Circuit*) koji kontrolira potrošnju cijelog sustava. TPS65217 potrebnu struju za rad sustava može vući iz tri različita izvora:

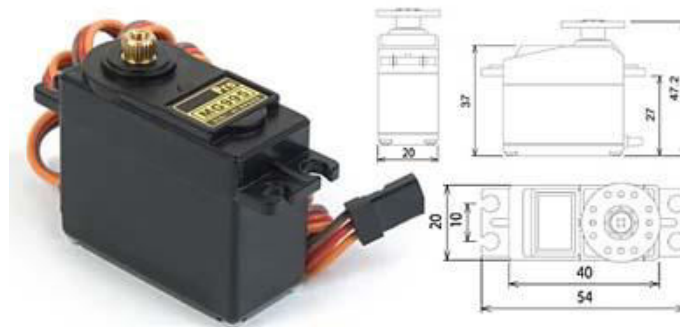
- 5V DC izvor preko DC konektora na pločici sustava ili pinova 5 i 6 na P9 konektoru
- Preko USB *power supply* linija sa mini-B konektora
- Koristeći standardne (3.6/3.7V) litij-ionske punjive baterije

Zbog relativno niske potrošnje, sustav se napaja pomoću 2A/5V *wall* adaptera čiji se napon dovodi na DC priključak pločice BBB sustava.

Distribucija Linuxa za koju je odlučeno da se koristi u projektu je BeagleBone Debian verzije 8.3. sa RT *patchem*, kojeg BeagleBone Black *boot*-a sa vanjske 16 GB mikro SD kartice (cijeli sustav trenutne verzije moguće je realizirati sa SD karticom minimalne veličine 4 GB).

2.2.2. Motori i upravljanje motorima

Sustav videonadzora predviđa korištenje motora za pomicanje kamere. Odlučeno je da se koriste TowerPro MG995 servo motori.



Slika 4:MG995 servo motor

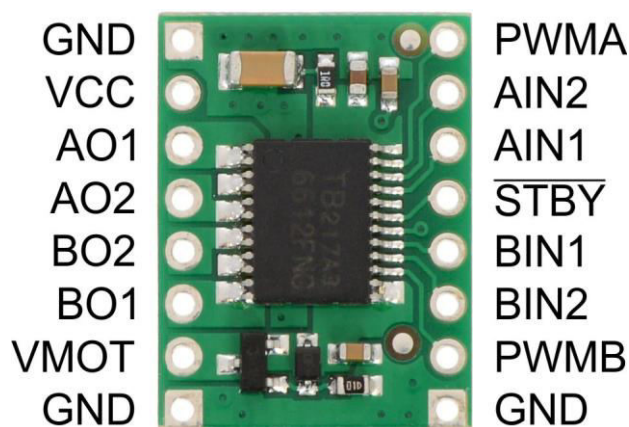
MG 995 su prilično jeftini servo motori koji su razvijeni prvenstveno za korištenje u RC modelarstvu. Malih su dimenzija i teže svega 55 grama. Rade na naponu od 4.8 V i 6.0 V (napon od 5 V ne predstavlja problem i neće utjecati na rad motora), te su zbog redukcije pomoću ugrađenih metalnih zupčanika u stanju dati od 9.5 do 11 kg-cm momenta (što je prilično impresivno za njihovu cijenu).

Servo motori se upravljaju pomoću PWM-a (pulsno širinska modulacija). *Duty cycle* PWM-a određuje položaj osovine motora u rasponu od 0° do 180°. Glavna osovina servo motora je spojena na potencijometar, preko kojeg se osigurava *feedback* do kontrolera unutar servo motora.

Zbog potrebe da motori rade u kontinuiranom načinu, te da im se brzina rotacije može smanjiti ili povećati pomoću PWM-a, motori korišteni u projektu su konvertirani. Kontroler je zajedno sa potencijetrom odstranjen, a *stopper* pin na zupčaniku glavne osovine je također odstranjen. Ovako smo dobili običan DC motor sa velikim stupnjem redukcije koji je u stanju dati veliki moment.

Spajanje DC motora direktno na GPIO priključke logike koja upravlja njima može se pokazati kao jako loša ideja. Naime pojava šiljaka zasigurno će uništiti GPIO priključak. U tu svrhu potrebno je koristiti *driver-e* za motore. Ovo su sklopovi koji služe kao sučelje između signalne elektronike male snage (kao npr. mikrokontroler) i uređaja koji su u stanju povući velike struje (kao npr. motori).

Za pokretanje motora u sustavu koriste se Sparkfun *driver* pločice temeljene na Toshiba TB6612FNG integriranom sklopu.



Slika 5:6612FNG *driver*

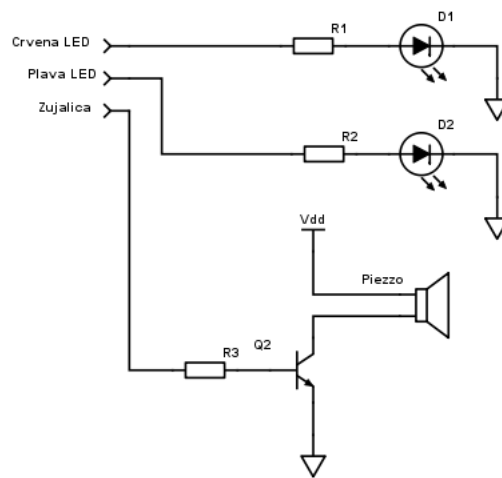
Pinovi AIN1 i AIN2 određuju smjer vrtnje, dok PWMA određuje brzinu vrtnje motora koji je priključen na AO1 i AO2. ANalogno vrijedi i za PWMB, BIN1, BIN2. Na GND i VCC dovodimo napajanje samog *driver-a*, dok između VMOT i GND dovodimo napajanje za motore.

Pinovi PWMA i PWMB se spajaju na P9.16 i P8.13 na BBB razvojnom sustavu respektivno. Naime, alternativna funkcija GPIO priključaka spojenih na P9.16 i P8.13, na koju su GPIO priključci konfigurirani prilikom učitavanja *default-ne .dtb* datoteke prilikom *boot*-anja Linux kernela, su izlazi iz kanala sklopova eHRPWMA i eHRPWMB čija je funkcija, začudo generiranje PWM signala.

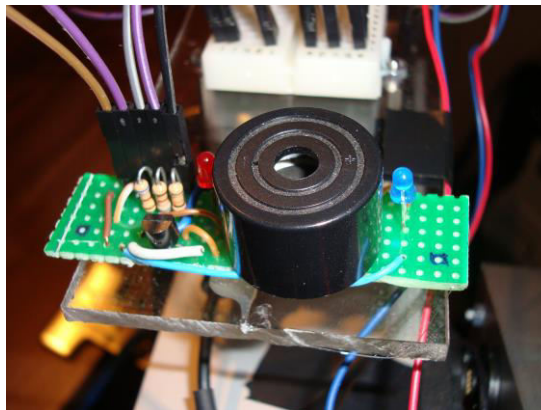
Pinovi AIN1, AIN2, BIN1 i BIN2 spojeni su na priključke BBB-a P8.7-10. Po *default*-u oni su konfigurirani kao GPIO priključci.

2.2.3. Sirena

Ukoliko dođe do detekcije pokreta sustav predviđa aktiviranje određenih mjera koje bi trebale odvratiti uljeza od njegovih namjera. Ovo se postiže aktivacijom sirene. Sirena je jednostavan sklop koji se sastoji od *piezzo* zujalice i pripadajućeg pokretačkog sklopovlja, te LED dioda. Pokretačko sklopovlje *piezzo* zujalice je zapravo jedan BC547 tranzistor. Baza tranzistora je preko otpornika od 10 k Ω spojena (u našem slučaju) na P8.15 izvod BeagleBone Black pločice. LED diode su spojene na P8.17 i P8.16 izvode.



Slika 6: Shema sirene



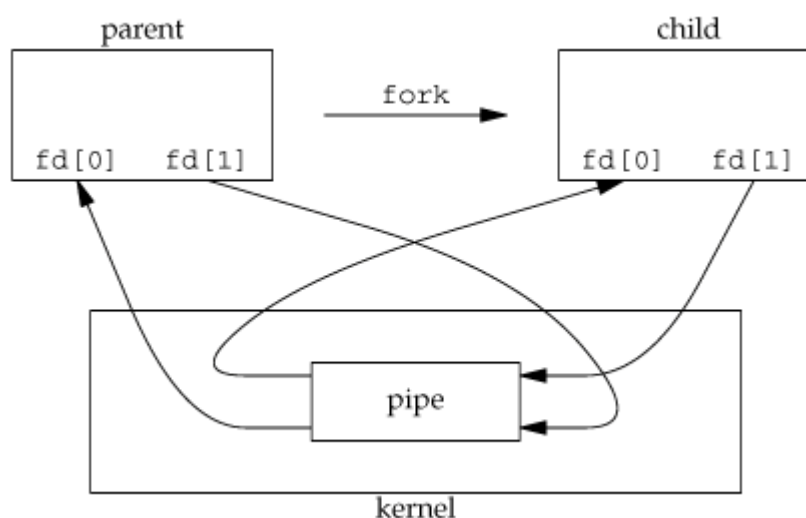
Slika 7: Realizacija sirene na perfboardu

2.3. Software

Ovdje će u kratkim crtama biti opisana softverska podrška koja se koristi u sustavu za videonadzor. Neće se ulaziti u dubinu kako radi cijeli sustav, jer bi to izlazilo iz okvira ovog rada. Također, cijeli kod je moguće naći *online*.

Kako je već spomenuto, sustav koristi BeagleBone Debian, Linux distribuciju namijenjenu za rad na nekoj od BeagleBone platform. Prilikom paljenja sustava, BBB *bootloader* započinje sa *boot*-anjem Linux jezgre i *load*-anjem željene *.dto* datoteke u kojoj je opisana hardverska konfiguracija cijele razvojne pločice (opisi pinova i njihova funkcija). Nakon što su pokrenuti svi potrebni servisi i aplikacije koje su potrebne za normalno funkcioniranje operacijskog sustava potrebno je pokrenuti aplikaciju za videonadzor.

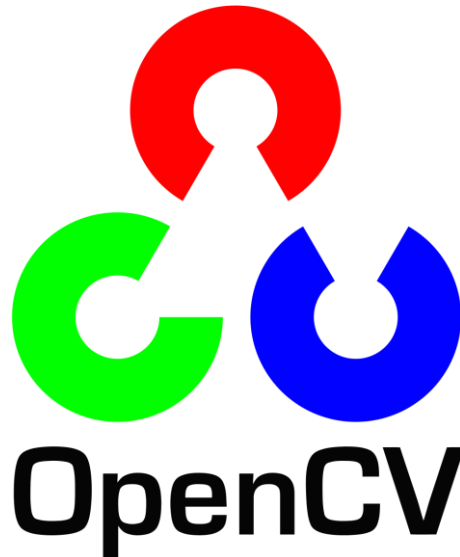
Softver sustava za videonadzor se u principu sastoji od dvije aplikacije, C++ aplikacije zadužene za obradu slike i upravljanje motorima i Node JS serverske aplikacije zadužene za registraciju korisnika, slanje slike i pružanje interaktivnog sučelja prema *client-side* aplikaciji. Komunikacija između dvije aplikacije odvija se preko IPC UNIX *pipe*-ova. Na slici 6 je prikazana ideja iza IPC mehanizma u UNIX sustavima koji korištenjem *pipe*-ova, implementiranih u kernelu, omogućuje komunikaciju između dva ili više procesa (npr. aplikacija u *user-space-u*).



Slika 8: Princip rada UNIX *pipe*-a

2.3.1. C++ aplikacija

Glavna funkcija C++ aplikacije je uzimanje slike sa video ulaza (u našem slučaju to je web kamera), ovisno o tome da li je korisnik uključio opciju *motion-detection* aplikacija u stvarnom vremenu (implementirana RT funkcionalnost u stanju je zadovoljiti *soft real time zahtjeve*) obrađuje niz od dvije slike između kojih pokušava naći određenu razliku koja je veća od zadanog praga (*threshold*), te ukoliko je pronađena razlika između dvije slike koja prelazi prag, aplikacija će zaključiti da je došlo do pomaka nekog od objekata na slici te se ovo signalizira ostatku sustava. Ako je pak opcija *motion-detection* isključena cijeli gore naveden postupak se preskače.



Slika 9:OpenCV

C++ aplikacija je također zadužena za upravljanjem motorima i sirenom. Princip upravljanja vanjskom periferijom, pa tako i GPIO sučeljem, iz bilo kojeg *user-space* programa na Linux-ima sastoji se od čitanja i pisanja u određene datoteke koje se kreiraju prilikom *load*-anja *device driver*-a.

Aplikacija također mora poslati zahtjev za stvaranjem dva IPC *pipe*-a, *img_node*-a koji služi za slanje slike Node JS aplikaciji i *ctrl_node*-a preko kojeg se razmjenjuju kontrolne komande. Bilo koja slika (bez obzira da li je uključena opcija *motion-detection*) se pomoću IPC mehanizma preko *img_node pipe*-a šalje Node JS aplikaciji. Node JS aplikacija pak preko *ctrl_node*-a C++ aplikaciji šalje komande ovisno o kojima C++ aplikacija pali i gasi motore i sirenu.

2.3.2. Node JS aplikacija

Za početak malo o Node.js-u. Node.js je JavaScript razvojno okruženje zasnovano na Chromovom V8 JavaScript *engine*-u. Node.js implementira *event-driven*, neblokirajući I/O model, no istovremeno je efikasan i ima mali memorijski utisak, što ga čini pogodnim za korištenje u ugradbenim računalima kao što je BeagleBone Black.

Glavna uloga Node JS aplikacije je pokretanje http servera. Aplikacija koristi Express Node.js web *framework* koji pruža set robusnih funkcionalnosti u cilju olakšavanja razvoja web aplikacija:

- Omogućuje definiranje middleware-a za odgovore HTTP zahtjevima
- Definiranje *routing* tablice koja se koristi da bi se izvele različite radnje ovisno o HTTP metodi i URL-u
- Dinamičko renderiranje HTML stranica ovisno o različitim argumentima

Node.js aplikacija tako pomoću korištenjem Express Node.js biblioteka krajnjem korisniku sustava servira stranice koje su potrebne za interakciju sa sustavom. Između ostalog, Node.js aplikacija je zadužena i za posluživanje i autentikaciju *login* i *register* zahtjeva te kreiranja *user* sesija. Za ovo je uz Express zadužen Passport *middleware* za autentikaciju. Također, sustav koristi i lokalnu MongoDB bazu korisnika sa kojom Node.js aplikacija komunicira preko *mongoose* biblioteke.



Slika 10: Neka od gotovih softverskih rješenja korištena u projektu

Komunikacija sa krajnjim korisnikom, odnosno sa *client-side* aplikacijom se vrši koristeći Socket.IO biblioteka. Socket.IO je JavaScript biblioteka koja nam omogućava bidirekcionalnu komunikaciju u stvarnom vremenu između klijenata i server. Sastoji se od dva dijela, *client-side* biblioteke koju pokreće preglednik, te *server-side* biblioteke za Node.js. Korištenje Socket.IO nam omogućuje realiziranje *real-time* web aplikacija, što je idealno ukoliko je potrebno ostvariti interakciju sa korisnikom, kao u našoj aplikaciji.



Slika 11:Socket.IO biblioteka

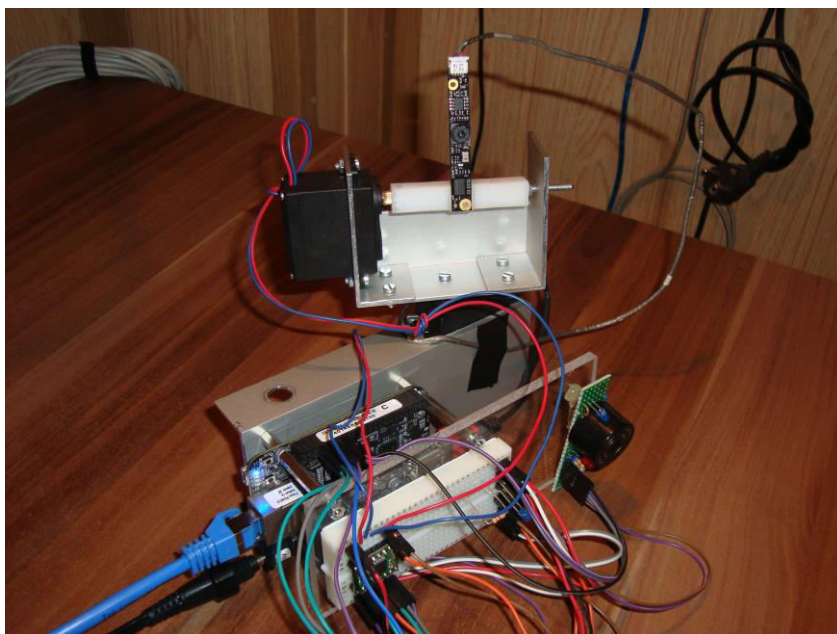
Također, kao što je već prije spomenuto Node.js aplikacija treba ostvariti komunikaciju sa C++ aplikacijom preko IPC *pipe*-ova. Ovo se u Node.js-u postiže upotrebom Node-ipc biblioteke. Pomoću ove biblioteke otvaraju se dvije konekcije prema *img_node* i *ctrl_node pipe*-ovima. Node.js aplikacija sa *img_node pipe*-a prima sliku kodiranu kao base-64 niz, te ju preko web socket-a (korištenjem Socket.IO biblioteke) prosljeđuje *client-side* aplikaciji. Također, komande se preko web socket-a prosljeđuju Node.js aplikaciji koja ih pak preko *ctrl_node pipe*-a prosljeđuje C++ aplikaciji, koja je zadužena za upravljanje periferijom.

U slučaju da C++ aplikacija detektira pomak, šalje se zastavica preko *ctrl_node pipe*-a koju prima Node.js aplikacija, ovo će pak uzrokovati da Node.js aplikacija pošalje mail korisniku sustava. Za ovo se koristi Nodemailer biblioteka.



Slika 12:Nodemailer biblioteka

2.4. Realizacija sustava i poboljšanja



Slika 13: Realizacija *proof of concept* verzije G.O.O.M.H. sustava

Uz sve gore navedene komponente sustav za dobivanje slike koristi običnu web kameru izvađenu iz starog laptopa. Slika koju dobivamo iz kamere (u trenutnoj verziji) je vrlo niske kvalitete, no veći problem predstavlja loša sklopovska kompenzacija i regulacija kontrasta slike samog modula kamere, te naponski propadi što pak može rezultirati lažnim detekcijama. Ovaj problem će se otkloniti u kasnijim verzijama sustava implementacijom performansama superiornije kamere i uvođenjem složenijih algoritama obrade slike. S druge strane na umu moramo imati da će ovo povećati ukupnu cijenu sustava.

Također, kako se C++ aplikacija poziva nakon prijave korisnika, kada se prvi korisnik prijavi procesna moć sustava će drastično opasti što će dovesti do toga da prijava novih korisnika traje podulje vrijeme. U kasnijim verzijama predviđa se korištenje mehanizama koji će u nekoj mjeri riješiti ovaj problem.



Slika 14: Slika web korisničkog sučelja

3. Zaključak

Kao što smo već vidjeli, bez obzira na stupanj razvijenosti koji moderan način života nosi sa sobom, ne smijemo zaboraviti na opasnosti i prijetnju od čevjekovog najvećeg neprijatelja, njega samog. U ovu svrhu pojedinac bi trebao razmisliti o načinima na koji može osigurati sebe i svoje bližnje, kao i svoja materijalna bogatstva.

Temeljem realizacije projekta, bez obzira na njegove nedostatke, možemo primjetiti da je danas sigurnost u u manjoj ili većoj mjeri dostupna svima, te je dokaz da bi osigurali prilično konfigurabilan i skalabilan sustav ne trebamo raspolagati većim budžetom.

Opisani sustav ostavlja puno mjesta dodatnom unapređenju, kao na primjer povezivanje više ovakvih sustava u mrežu, što bi korisniku osiguralo *state of the art* sustav videonadzora sa potpunom kontrolom nad konfiguracijom.

4. Literatura

5. Pojmovnik

Pojam	Kratko objašnjenje	Više informacija potražite na
White paper	Kratak dokument koji daje uvid u neko područje, tehniku, politiku, proizvod, metodu, standard i sl.	en.wikipedia.org/wiki/White_paper
PWM	Pulsno širinska modulacija	https://en.wikipedia.org/wiki/Pulse-width_modulation
GPIO	General Purpose Input/Output	https://en.wikipedia.org/wiki/General-purpose_input/output
IPC	Inter Process Communication	https://en.wikipedia.org/wiki/Inter-process_communication
Named pipe (FIFO)	Koncept cijevovoda u UNIX kernelu za postizanje IPC-a	https://en.wikipedia.org/wiki/Named_pipe