Udaljeni sigurnosni sustav s laserima i kamerom te dojavom na mobitel

Umiljanović, Luka

Undergraduate thesis / Završni rad

2016

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:275623

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

zir.nsk.hr

Image not found or type unknown

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij

UDALJENI SIGURNOSNI SUSTAV S LASERIMA I KAMEROM TE DOJAVOM NA MOBITEL

Završni rad

Luka Umiljanović

Osijek, 2016.

Ovdje treba umetnuti: Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom studiju.

Ovdje treba umetnuti: Izjavu o originalnosti završnog rada.

SADRŽAJ

1. UVOD 1

1.1. Zadatak završnog rada 1

2. PRIMIJENJENE TEHNOLOGIJE I ALATI 2

2.1. Arduino integrirano razvojno okruženje 2

2.2. Processing programski jezik i razvojno okruženje 3

2.3. Dropbox 4

2.4. SainSmart MEGA2560 R3 razvojni sustav 5

2.5. ArduCAM Rev.C+ Shield sa OV2640 2MP kamerom 6

2.6. ESP8266-01 WiFi modul 7

2.7. Keyes KY-008 laserski modul 8

2.8. 3DU5C fototranzistor 8

2.9. microSD memorijska kartica 9

2.10. MB-102 eksperimentalna pločica 10

2.11. Otpornici 10

3. REALIZACIJA SUSTAVA 11

3.1. Spajanje fizičkih komponenti 11

3.2. Programsko rješenje sustava 12

3.2.1. Arduino IDE kod 12

3.2.2. Processing IDE kod 13

3.2.3. Mobilne aplikacije 14

4. ZAKLJUČAK 15

LITERATURA 16

SAŽETAK 17

ABSTRACT 18

ŽIVOTOPIS 19

PRILOZI 20

PRILOG A. ArduCAM Rev.C+ pinout 20

PRILOG B. SainSmart MEGA2560 R3 pinout 21

PRILOG C. ESP8266-01 pinout 22

PRILOG D. KY-008 pinout 22

PRILOG E. Shema cijelog sustava 23

PRILOG F. Arduino IDE programski kod 24

PRILOG G. Processing IDE programski kod 28

1. UVOD

Cilj ovog završnog rada je izraditi sustav detekcije i signalizacije pokreta. Kada dođe do prekida laserske zrake dolazi do aktivacije kamere i fotografiranja, gdje se fotografija pohranjuje na SD karticu. Nakon toga je potrebno poslati tekstualnu obavijest i fotografiju korisniku putem WiFi mreže. Tekstualna obavijest se šalje na e-mail račun korisnika, dok se fotografija šalje na Dropbox račun. Da bi korisnik dobio obavijest na mobilnom uređaju, potrebno je imati odabran odgovarajući račun u e-mail aplikaciji i Dropbox aplikaciji. Kroz rad će se razmotriti i odabrati tehnologije kojima je moguće riješiti zadani problem. Nakon odabira tehnologija pojasniti će se realizacija sustava.

1.1. Zadatak završnog rada

U ovom završnom radu potrebno je napraviti mikroupravljački sigurnosni sustav za detekciju pokreta pomoću lasera te dojavu putem interneta. Dojava uključuje slanje slike s udaljenog mjesta na kojemu je postavljen ovaj sigurnosni sustav.

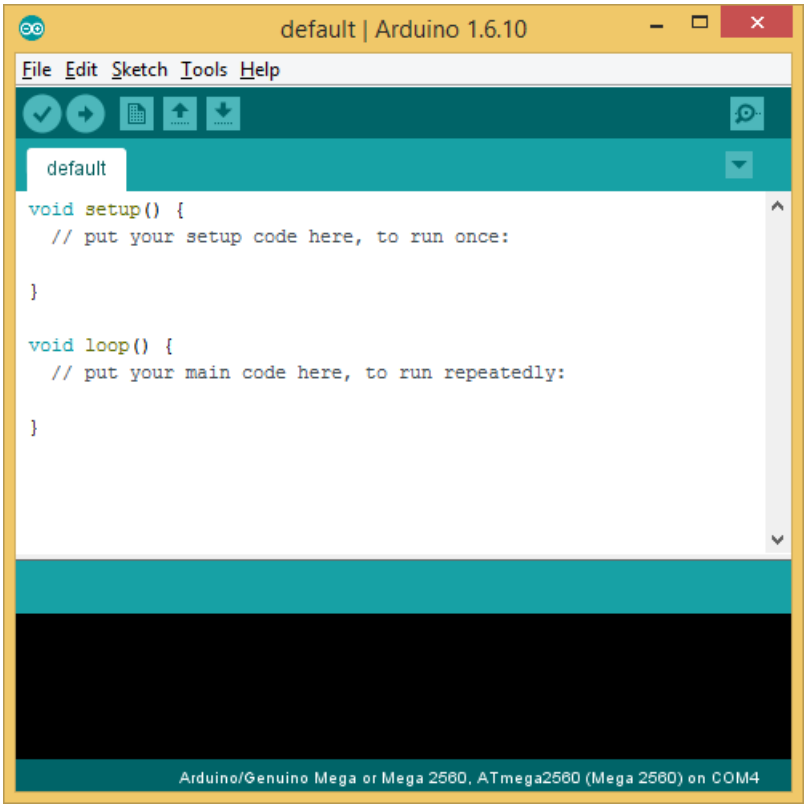
2. PRIMIJENJENE TEHNOLOGIJE I ALATI

2.1. Arduino integrirano razvojno okruženje

Arduino integrirano razvojno okruženje (engl. *Integrated Development Environment (IDE)*) [1][2] je višeplatformska aplikacija napisana u programskom jeziku Java. Potječe iz razvojnih okruženja za programske jezike Processing i Wiring. Dizajnirana je za približavanje programiranja ljudima koji nisu upoznati s razvojem programskih rješenja. Uključuje uređivač koda sa svojstvima isticanja sintakse bojom, samostalnog zatvaranja zagrada, automatskog uvlačenja i nudi mogućnost prevođenja i učitavanja programa jednim klikom na Arduino ploču. Program napisan sa Arduino IDE se naziva skica (engl. *sketch*).

Arduino IDE podržava programske jezike C i C++, pri čemu koristi posebna pravila za organiziranje koda. Nudi programsku biblioteku Wiring iz Wiring projekta koja nudi mnoge česte ulazne i izlazne procedure. Tipična Arduino C/C++ skica sadrži dvije funkcije koje su prevedene i povezane sa programskim stubom [3] *main()* u izvršni ciklički program:

- *setup()*: funkcija koja se izvodi samo jednom na početku programa i koja inicijalizira postavke.
- *loop()*: funkcija koja se poziva sve dok se ne odspoji napajanje ploče.

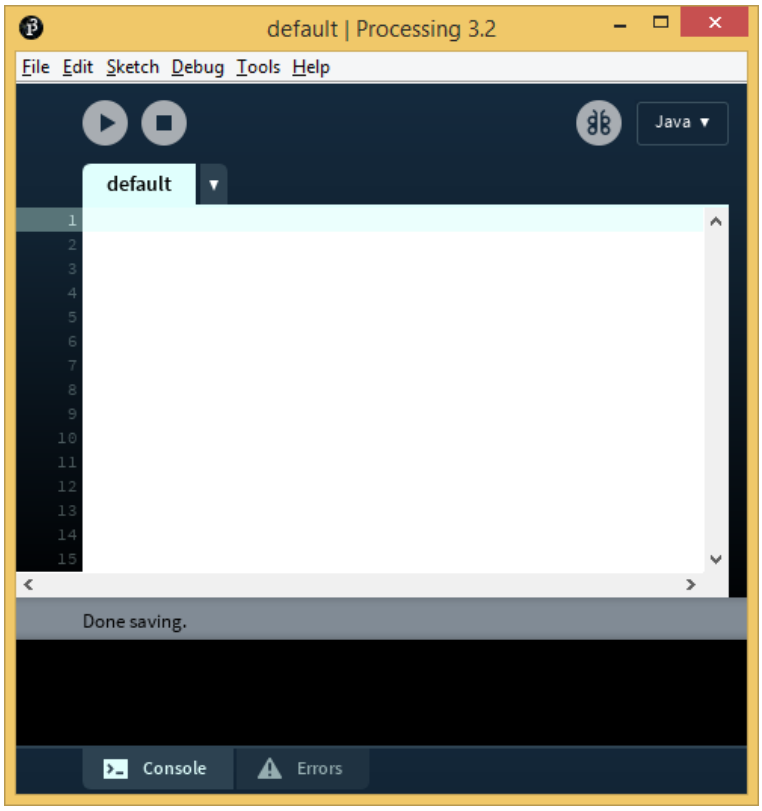


Sl. 2.1. Arduino razvojno okruženje.

2.2. Processing programski jezik i razvojno okruženje

Processing [4] je računalni programski jezik i integrirano razvojno okruženje otvorenog koda stvoreno za elektroničke umjetnosti, novomedijske umjetnosti, i zajednice vizualnog dizajniranja s ciljem učenja osnova računalnog programiranja u vizualnom kontekstu. Jedan od ciljeva Processinga je dozvoliti ne-programerima početak računalnog programiranja potpomognut vizualnom povratnom vezom. Processing jezik gradi na Java programskom jeziku, no koristi pojednostavljenu sintaksu i grafičko korisničko sučelje. Program napisan sa Processing IDE se naziva skica (engl. *sketch*).

Processing uključuje *sketchbook*, minimalnu alternativu integriranom razvojnom okruženju za organiziranje projekata. Svaka skica je zapravo podklasa PApplet Java klase koja implementira većinu značajki Processing jezika. Kod programiranja u Processingu sve dodatno definirane klase biti će tretirane kao unutrašnje klase koje se pretvaraju u čisti Java jezik prije prevođenja. To znači da je korištenje statičkih varijabli i metoda u klasama zabranjeno osim ako se posebno ne navede da se želi kodirati u čistom Java načinu. Processing također dozvoljava korisnicima kreiranje vlastitih klasa unutar PApplet skice. To dozvoljava složenim tipovima podataka korištenje bilo kojeg broja argumenata i izbjegavanje ograničenja jedino korištenjem standardnih tipova podataka kao što su integer, char, float i color.



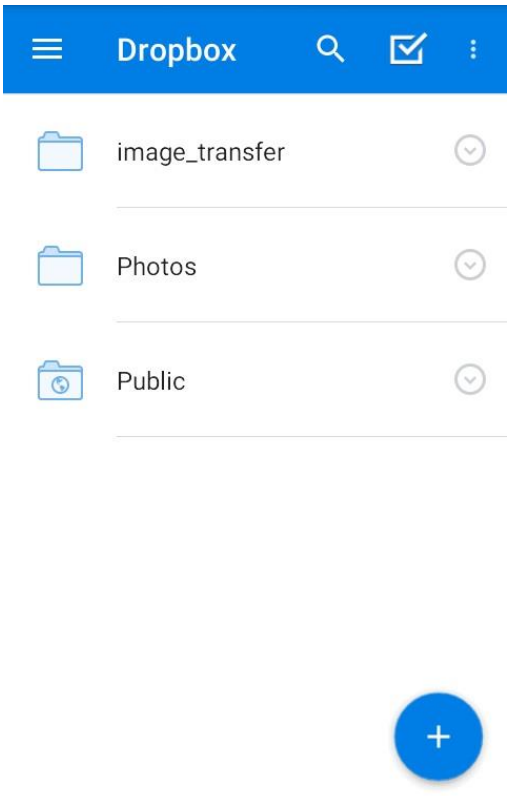
Sl. 2.2. Processing razvojno okruženje.

2.3. Dropbox

Dropbox je *online* servis koji nudi pohranjivanja podataka u oblaku (engl. *cloud*), sinkronizaciju te korisničke programe. Dozvoljava korisnicima stvaranja posebne mape na svom računalu koju Dropbox zatim sinkronizira tako da sadržaj bude dostupan neovisno o tome nalaze li se računala u istoj lokalnoj mreži ili su povezana putem Interneta. Datoteke pohranjene u ovoj mapi su također dostupne putem internetske stranice i mobilnih aplikacija. Također postoji mogućnost dijeljenja mape s drugim osobama, recimo timom koji radi zajedno na istom projektu. Sav sadržaj te mape biti će dostupan svima s kojima se dijeli mapa; svi mogu dodavati, mijenjati ili brisati sadržaj te mape. Datoteke je moguće i dijeliti i s onima koji nisu korisnici servisa tako da se pohrane u mapu Public. Još jedna od prednosti Dropboxa je to što snima povijest svih obrisanih i prethodnih verzija datoteka, tako da postoji mogućnost povratka ako je potrebna datoteka koja je obrisana.

Dropbox [5] koristi *freemium* poslovni model, što znači da je korisnicima ponuđen besplatni račun sa zadanim kapacitetom od 2 GB za pohranu, te mogućnost plaćenih pretplata gdje su dostupni veći kapaciteti i dodatne mogućnosti.

Korisnički program nema ograničenja za veličinu datoteke koja se pohranjuje, dok je za *web* aplikaciju to ograničeno na 10 GB po datoteci.

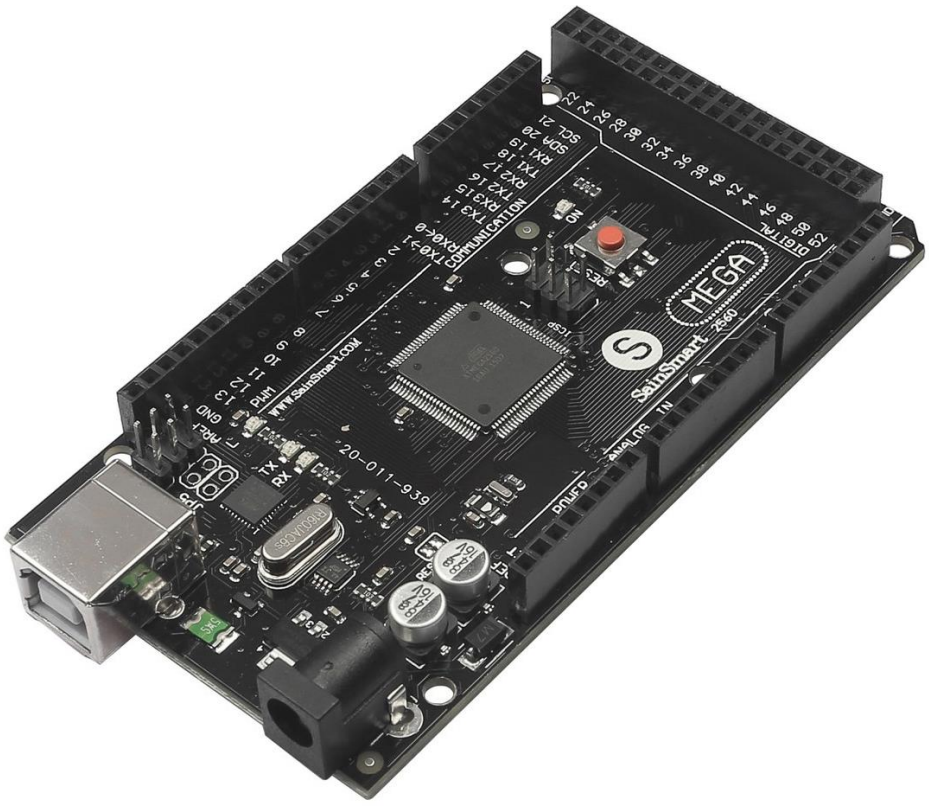


Sl. 2.3. Sučelje Dropbox Android aplikacije.

2.4. SainSmart MEGA2560 R3 razvojni sustav

SainSmart MEGA2560 R3 [6] je razvojni sustav temeljen na ATmega2560 mikrokontroleru . Ima 54 digitalna ulazna/izlazna *pina* (od kojih 15 može biti korišteno za PWM izlaz), 16 analognih ulaza, 4 UART serijska porta, 16 MHz kristalni oscilator, USB priključak, utičnicu za vlastito napajanje, ICSP zaglavlje i *reset* dugme. Sadrži sve potrebno za podržavanje rada mikrokontrolera; jednostavno je potrebno spojiti na računalo putem USB kabela ili ga napajati pomoću AC-na-DC adaptera ili baterije. SainSmart MEGA2560 R3 je nadogradnja razvojnog sustava SainSmart MEGA. Koristi ATMega16U2 umjesto ATMega8U2 čipa, što omogućuje brže brzine prijenosa i više memorije. Nisu potrebni upravljački programi (engl. *driver*) za Linux ili Mac (inf datoteka za Windows operativni sustav je potrebna i uključena je u Arduino IDE). R3 verzija ima dodane SDA i SCL *pinove* koji su pored AREF *pina*, te dva nova *pina* smještena pored RESET *pina*; IOREF omogućuje *shieldovima* [7] prilagođavanje napona dostupnog iz ploče, dok drugi nije spojen i rezerviran je za buduće potrebe. Mega 2560 sustav je kompatibilan s većinom *shieldova* dizajniranih za Uno, ali se i može prilagoditi novim *shieldovima* koji koriste navedene dodatne *pinove*.

SainSmart MEGA2560 R3 razvojni sustav je jeftinija verzija Arduino Mega 2560 sustava [8], no sadrži sve jednake funkcionalnosti.

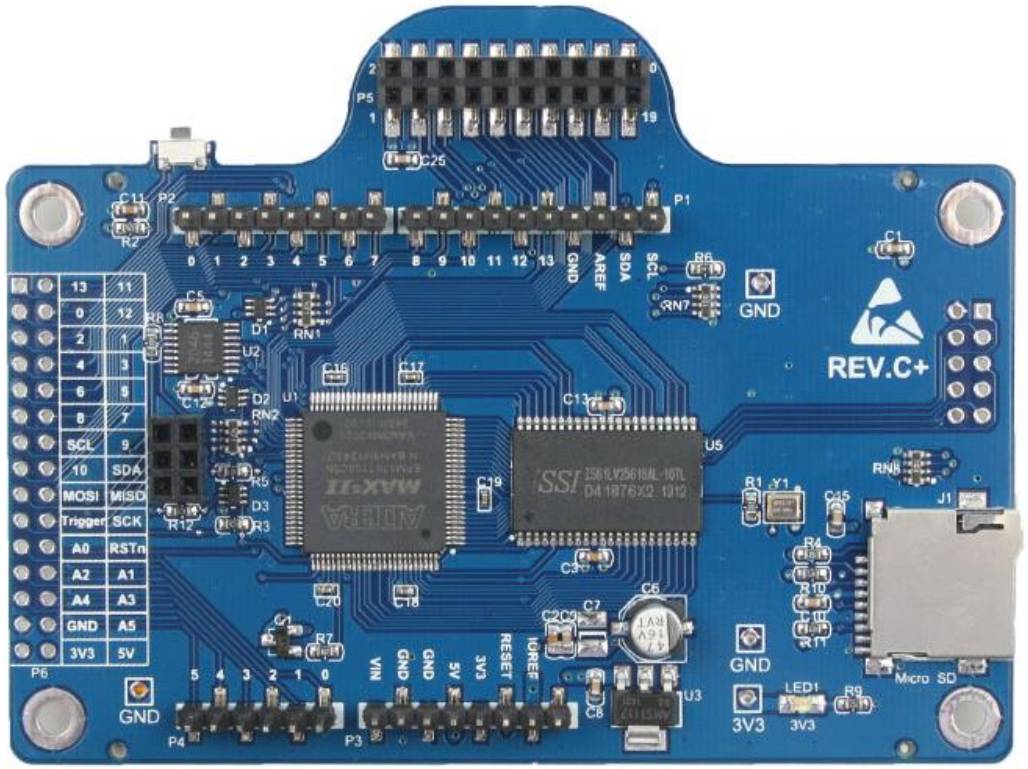


Sl. 2.4. SainSmart MEGA2560 R3 razvojni sustav.

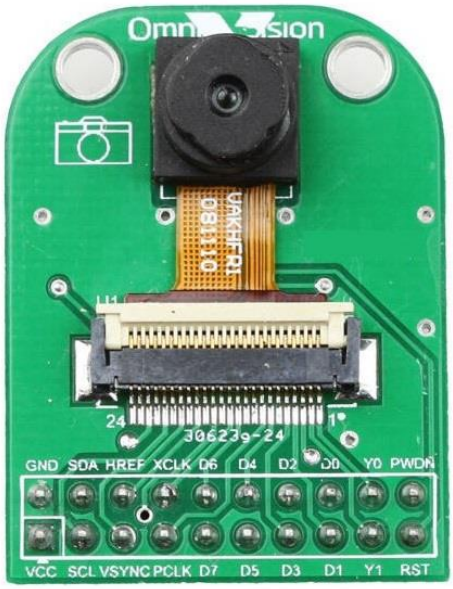
2.5. ArduCAM Rev.C+ Shield sa OV2640 2MP kamerom

ArduCAM shield s kamerom je univerzalna upravljačka ploča za Arduino. Prikriva složenost funkcioniranja kamere i pruža jednostavan način korištenja upravljačkog sučelja kamere, kao i gotove biblioteke izvornog koda i primjere koda. ArduCAM podržava različite vrste kompaktibilnih modula kamera od 0.3MP do 5MP s različitim formatima kao što su RAW, RGB, YUV, JPEG. Lako je upariva s Arduino razvojnim sustavima i sklopovima. ArduCAM Rev.C+ [9][10] je najnovija verzija ArduCAM *shielda*. Pruža poboljšane performanse i povećanu funkcionalnost nego prijašnje verzije (ArduCAM Rev.C). Može snimati fotografije do maksimalno 5MP, što činu ovu upravljačku ploču idealnim rješenjem za IoT aplikacije. Također sadrži ugrađen micro SD/TF priključak. ArduCAM je dizajnirana za Arduino, no nije ograničena na Arduino: može biti korištena na različitim platformama kao što su Raspberry Pi, Maple, Chipkit, Beaglebone black, sve dok te platforme imaju SPI i I2C sučelja.

OV2640 modul [11] nudi funkcionalnost 2MP kamere. Jedna od glavnih svojstava je to da ima paralelno sučelje i mogućnost JPEG izlaza.



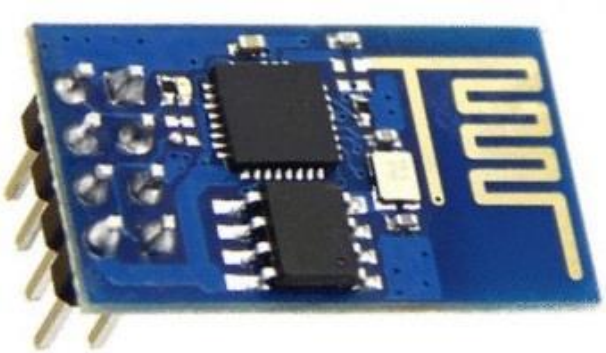
Sl. 2.5. ArduCAM Rev.C+ Shield.



Sl. 2.6. OV2640 modul.

2.6. ESP8266-01 WiFi modul

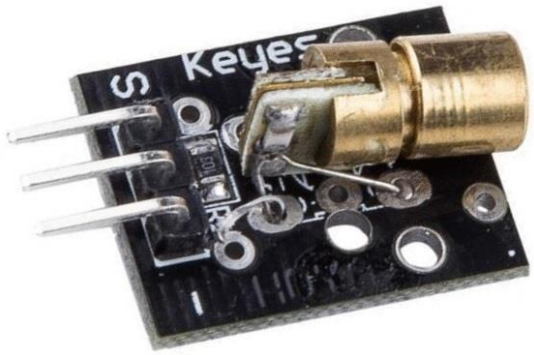
ESP8266 [12] je sustav na čipu sa TCP/IP stogom koji omogućava bilo kojem mikrokontroleru pristup WiFi mreži. EP8266 ima mogućnost rada kao WiFi bazna stanica (engl. *access point*) ili WiFi klijent (engl. *WiFi station*). Postoji više različitih modela i proizvođača modula (Espressif, AI-Thinker…). U ovom slučaju korišten je modul proizvođača AI-Thinker. Svaki ESP8266 modul dolazi programiran sa AT *firmwareom*, što znači da je moguće dobiti WiFi povezanost gotovo jednako kao i sa WiFi *shieldom*. ESP8266 je izrazito efikasan i popularan sustav s velikom zajednicom korisnika. Posebno je potrebno naglasiti da čip radi na 3.3 V te komunicira serijskim putem na 3.3 V i nema 5 V tolerantne ulaze, stoga je potrebna logička konverzija ili djelitelj napona kako bi mogao komunicirati sa SainSmart MEGA2560 R3 razvojnim sustavom koji funkcionira na 5 V (kao i većina mikrokontrolera).



Sl. 2.7. ESP8266-01 WiFi modul.

2.7. Keyes KY-008 laserski modul

KY-008 [13] je jedan od najčešće korištenih uređaja kojeg je moguće pronaći u Arduino kompletima senzora. Uparen s različitim komponentama može biti korišten za kreiranje zamki i optičkih efekata. Modul je zanimljiv jer može biti napajan izravno pomoću Arduino i sličnih razvojnih sustava jer je za rad potrebno 30 mA struje. Radni napon je 5 V, a odašiljana zraka je valne duljine 650 nm. Sadrži tri *pina*: Vdc, Ref i GND. Središnji *pin* (Ref) daje izlazan napon jednak ulaznom i najčešće se ne koristi. Modul sadrži i rupice koje se mogu koristiti za lakšu instalaciju i učvršćivanje.



Sl. 2.8. KY-008 laserski modul.

2.8. 3DU5C fototranzistor

Fototranzistor je bipolarna elektronička komponenta koja pretvara svjetlosnu energiju u električnu. Načinjen je od silicija ili nekog drugog poluvodičkog materijala i obično je obložen neprozirnim ili čistim spremnikom kako bi se poboljšala izloženost svjetlosti. Nema spoj baze, nego je baza odspojena i izložena svjetlosti, što omogućava tok struje kroz fototranzistor. Iako postoji mala količina struje (tamna struja) koja teče fototranzistorom kada nema svjetlosti, što je zbog malog broja nosioca koji su ubačeni u emiter, fototranzistor je neaktivan sve dok ne dođe u kontakt sa svjetlošću. Djelovanje fototranzistora ovisi o izloženosti fotonima (svjetlosnim česticama). Veći intenzitet svjetlosti jednak je većem naponu. Fotostruja se na bazi povećava i širenjem od baze do emitera ona se koncentrira i pretvara u napon. Neke od prednosti fototranzistora su to što proizvode veću struju od fotodioda, proizvode struju i napon za razliku od fotootpornika koji proizvode samo struju, iznimno su brzi i njihov izlaz je gotovo trenutan, jednostavni su i nisu skupi, te funkcioniraju s većinom izvora vidljive svjetlosti.

3DU5C NPN silicijski fototranzistor [14] je široko korišten u raznim vrstama električnih krugova. Za rad je potrebno 30 mA struje a maksimalni radni napon je 10 V.



Sl. 2.9. 3DU5C fototranzistor.

2.9. microSD memorijska kartica

Memorijska kartica (ponekad zvana još i *flash* kartica) je mali medij za pohranu podataka za uporabu manjim prijenosnim ili udaljenim računalnim uređajima. Postoji više različitih memorijskih kartica, npr. SD kartica (engl. *secure digital card*), CF kartica (engl. *CompactFlash card*), SmartMedia kartica, itd. Kartice su različitih dimenzija i memorijskih kapaciteta. Većina raspoloživih kartica ima nepromjenjivu memoriju, što znači da su podaci stabilni i nisu ugroženi gubitkom napajanja te ne moraju biti periodično osvježeni. Memorijske kartice su elektronički krute (engl. *solid state*) memorije, što znači da nemaju pokretnih dijelova i otporniji su na mehaničke kvarove. Također su manjih fizičkih dimenzija, lakše, izrazito prijenosne, potpuno tihe i dozvoljavaju brži pristup podacima od tvrdih diskova. Brzina zapisivanja i čitanja kartice određuje kojom brzinom se oslobađa privremeni međuspremnik (engl. *buffer*) za prijem novih podataka. Kartice većeg kapaciteta mogu biti sporije od kartica manjeg kapaciteta.

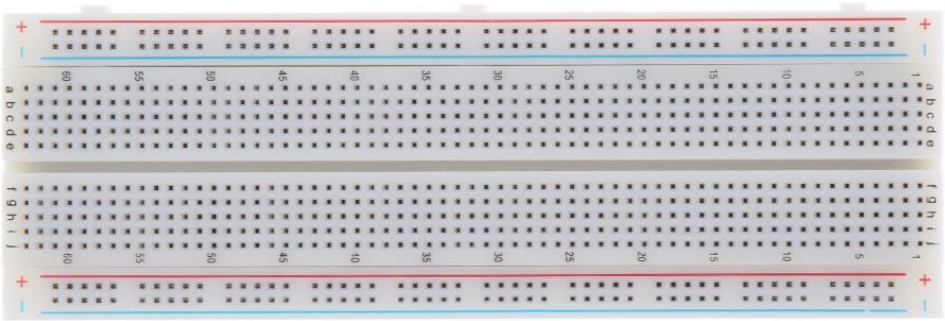
MicroSD [15] kartice su najmanje komercijalno dostupne memorijske kartice s dimenzijama 15x11x1 mm ali danas mogu pohraniti više od dvjesto gigabajta podataka. U ovom radu korištena je kartica kapaciteta 32 GB.



Sl. 2.10. MicroSD kartica i adapter.

2.10. MB-102 eksperimentalna pločica

Eksperimentalna pločica (engl. *solderless breadboard*) [16] je široko korišteno pomagalo za dizajniranje i testiranje električnih krugova. Nije potrebno spajanje komponenti lemljenjem nego se povezivanje ostvaruje umetanjem nožica i žica u rupice na pločici, čime se omogućuje jednostavno testiranje i prototipiranje. Pločica se sastoji od metalnih traka sa spojnicama koje su međusobno povezane prema određenom pravilu i zaštićene plastičnim kućištem. Obično je podijeljena u četiri dijela: dva vanjska i dva unutarnja. U sredini pločice se nalazi rascjep standardizirane širine. Rupice unutar svakog unutrašnjeg područja su spojene vodoravno čineći u istom redu dva vodoravna niza od po pet rupica. Redovi su označeni brojevima (1-60), a stupci slovima (a-j). Rupice u vanjskim dijelovima su povezane okomito i obično se koriste za napajanje. Ti dijelovi se nazivaju „tračnice“ i najčešće se uz svaku nalaze crvena i plava (crna) linija sa simbolima (+) i (-). Crvena tračnica se obično koristi za dovođenje napona, a plava za uzemljenje. Na većini pločica rupice u tračnicama nisu spojene po cijeloj dužini, nego su prekinute na sredini i potrebno ih je spojiti žicom ako ih se želi koristiti zajedno. Razmaci između dvije rupice su standardizirani i iznose 2.5 mm. Eksperimentalne pločice nemaju vlastito napajanje, iako neke dolaze s priključnim točkama.



Sl. 2.11. MB-102 eksperimentalna pločica.

2.11. Otpornici

Otpornici su elektroničke komponente koje imaju nepromjenjiv električni otpor. Otpornikov otpor, koji je jednak naponu na otporniku podijeljenom sa strujom koja protječe kroz otpornik, ograničava tok elektrona kroz električni krug. To su pasivni elementi, što znači da samo troše energiju. Obično se dodaju u električne krugove s aktivnim komponentama kao što su operacijska pojačala, mikrokontroleri i drugi integrirani krugovi. Većinom se koriste za ograničavanje struje, podjelu napona i *pull-up* ili *pull-down* I/O linija. Pri realizaciji sklopa korišteni su 10 kΩ i 20 kΩ otpornici. Određivanje vrijednosti otpora vrši se pomoću boja na otporniku [17].

3. REALIZACIJA SUSTAVA

Sustav se može podijeliti u tri cjeline: detekcija pokreta, snimanje i spremanje fotografije, slanje putem WiFi veze. Na početku se vrši povezivanje ESP8266 modula na Internet mrežu. Kada se laserska zraka emitirana KY-008 modulom i primana fototranzistorom prekine, dolazi do aktivacije OV2640 kamere spojene na ArduCAM Rev.C+ *shield*, fotografiranja te spremanja na microSD karticu. Na kraju se vrši povezivanje s korisničkim e-mail računom i slanje tekstualne poruke obavijesti te slanje slike serijskom vezom u Dropbox direktorij.

3.1. Spajanje fizičkih komponenti

U prilogu su dani prikazi *pinova* pojedinih komponenti, dok je u ovom potpoglavlju opisan način njihovog spajanja. Komponente su spojene Arduino spojnim žicama i u nekim slučajevima pomoću eksperimentalne pločice.

Način spajanja pinova SainSmart MEGA2560 R3 razvojnog sustava i ArduCAM Rev.C+ s*hielda* dan je tablicom 3.1.

Tab. 3.1. Spajanje MEGA2560 i Rev.C+.

SainSmart MEGA2560	**ArduCAM Rev.C+**
SCL	SCL
SDA	SDA
D10	D10
D9	D9
A5	A5
A4	A4
ICSP (6 pinova)	SPI (6 pinova)
5V	5V
GND	GND

5V *pinovi* MEGA2560 i Rev.C+ su međusobno povezani pomoću + (crvene) tračnice eksperimentalne pločice. GND *pinovi* MEGA2560 i Rev.C+ su međusobno povezani pomoću - (plave) tračnice eksperimentalne pločice.

Potrebno je umetnuti microSD karticu u odgovarajuće mjesto na Rev.C+ *shieldu*.

Način povezivanja *pinova* SainSmart MEGA2560 R3 razvojnog sustava i ESP8266-01 modula dan je tablicom 3.2.

Tab. 3.2. spajanje MEGA2560 i ESP8266-01

SainSmart MEGA2560	ESP8266-01
RX2	TX
3.3V	VCC
3.3V	CH_PD
GND	GND

Kako TX2 *pin* MEGA2560 radi na 5 V, a RX *pin* ESP8266 na 3.3 V, potrebno je koristiti djelitelj napona [18], gdje koristimo 10kΩ i 20kΩ otpornike. Na ulaz djelitelja napona dovodimo TX2 *pin* MEGA2560, dok izlaz djelitelja spajamo na RX *pin* ESP8266 modula. R_1 iznosi 10kΩ, a R_2 20kΩ. Djelitelj napona je izveden na eksperimentalnoj pločici. Također, 3.3V *pin* MEGA2560 i VCC, te CH_PD ESP8266 su međusobno povezani pomoću + (crvene) tračnice eksperimentalne pločice (suprotna strana od one gdje je dovedeno 5V). GND *pinovi* MEGA2560 i ESP8266 su međusobno povezani pomoću - (plave) tračnice eksperimentalne pločice.

KY-008 laserski modul funkcionira tako da se - *pin* spoji na GND, a S *pin* na 5V koristeći eksperimentalnu pločicu. Srednji *pin* se ne spaja.

Fototranzistor 3DU5C funkcionira tako da se minus pol spoji na A15 *pin* MEGA2560, dok se plus pol spaja na 5V. Također se koristi otpornik od 10 kΩ (*pull-down* otpornik) [19], tako da se jedna nožica spoji na A0 *pin* a druga na GND koristeći eksperimentalnu pločicu.

Shema cijelog sustava nalazi se u prilogu E.

3.2. Programsko rješenje sustava

3.2.1. Arduino IDE kod

Prije pisanja programskog koda, potrebno je uključiti službene biblioteke potrebne za rad ArduCAM Rev.C+ *shielda* i OV2640 kamere. Upute su opisane na poveznici [20], dok se biblioteke mogu preuzeti sa poveznice [21].

Cjelokupan programski kod nalazi se u prilogu F.

Prije korištenja je potrebno izmijeniti sljedeće linije koda sa korisničkim podacima:

```
Serial2.println("AT+CWJAP=\"SSID\",\"PASSWORD\""); // unijeti SSID i lozinku mreže
Serial2.println("AT+CIPSEND=NUMBER OF E-MAIL BITS + 2"); // unijeti odgovarajući broj bitova
Serial2.println("MAIL FROM:<E-MAIL>"); // unijeti e-mail adresu korisnika
```

```
Serial2.println("AT+CIPSEND=NUMBER OF E-MAIL BITS + 2"); // unijeti odgovarajući broj bitova
Serial2.println("RCPT To:<E-MAIL>"); // unijeti e-mail adresu korisnika
```

Kako većina e-mail servisa filtrira poruke s nepoznatih izvora, potrebno je koristiti alternativni odlazni server. U ovom slučaju korišten je smpt2go [22]. Potrebno se je registrirati, te spojiti e-mail račun koji želimo koristiti za obavijesti (*Settings – Users – Add SMPT User*).

Nakon toga je potrebno pretvoriti e-mail adresu i dobivenu lozinku (za uvid kliknuti na dodani e-mail račun pod *Manage SMTP Users*) u Base64 format [23]. To je moguće na poveznici [24].

Zatim je potrebno izmijeniti sljedeće linije koda sa korisničkim podacima:

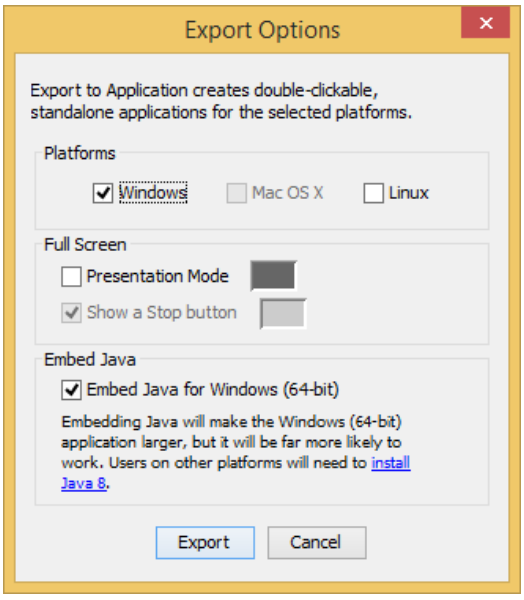
```
Serial2.println("AT+CIPSEND=NUMBER OF USERNAME BASE64 BITS + 2"); // unijeti broj bitova
Serial2.println("USERNAME BASE64"); // e-mail adresa u Base64 formatu
Serial2.println("AT+CIPSEND=NUMBER OF PASSWORD BASE64 BITS + 2"); // unijeti broj bitova
Serial2.println("PASSWORD BASE64"); // lozinka u Base64 formatu
```

3.2.2. Processing IDE kod

Kod napisan u Processingu služiti će za primanje fotografije na računalu.

Cjelokupan programski kod nalazi se u prilogu G.

Nakon pisanja koda potrebno je izvesti kod u aplikaciju (*File – Export Application*) postavkama kao na slici 3.1.



Sl. 3.1. Postavke za izvoz Processing aplikacije.

Nakon toga će se stvoriti dvije mape različitih verzija aplikacije (32-bitna i 64-bitna). Ovisno o vrsti Windows operacijskog sustava (32-bit ili 64-bit) potrebno premjestiti odgovarajuću mapu u Dropbox direktorij.

Prije korištenja sustava prvo je potrebno pokrenuti tu aplikaciju na računalu.

3.2.3. Mobilne aplikacije

Da bi korisnik mogao primati e-mail obavijesti i fotografije potrebno je imati instalirane e-mail i Dropbox aplikacije na mobilnom uređaju. U ovom slučaju korišten je Gmail e-mail račun stoga je korištena i Gmail mobilna aplikacija koju je moguće preuzeti na poveznici [25]. Dropbox mobilnu aplikaciju je moguće preuzeti na poveznici [26]. Također se je potrebno prijaviti sa istim korisničkim računom kao i na Dropbox aplikaciji na računalu.

4. ZAKLJUČAK

U ovom završnom radu izrađen je mikroupravljački sustav detekcije i signalizacije pokreta. Sustav je uspješno realiziran i obavlja zamišljeni zadatak: vrši detekciju pokreta, snimanje i spremanje fotografije, te slanje fotografije i tekstualne obavijesti putem Interneta. Jedina mana je što s korištenim WiFi modulom nije bilo moguće realizirati slanje fotografije, nego je potrebno korištenje računala. Do problema dolazi zbog loše komunikacije korištenog modula i razvojnog sustava, te ograničenosti modula da može slati maksimalno 2048 bajta po paketu. Rješenje bi moglo biti korištenje pretvarača logičke razine (engl. *logic level converter*), novije verzije Wifi modula ili čak korištenje Ethernet *shielda*. Unatoč tome, i ovakva realizacija pokazuje mogućnost samostalnog sličnog Internet stvari (engl. *Internet of Things*) projekta. Sustav ima potencijalno korisnu primjenu u stvarnom životu, čak i uz korištenje navedenih dijelova.

LITERATURA

[1] Arduino uvod, https://goo.gl/qTqfJK, rujan 2016.

[2] Arduino razvojno okruženje (IDE), https://goo.gl/YSvlZc, rujan 2016.

[3] Stub, http://goo.gl/llSLVN, rujan 2016.

[4] Processing, http://goo.gl/NiNfCx, rujan 2016.

[5] Dropbox, http://goo.gl/HVZYDn, rujan 2016.

[6] SainSmart MEGA2560 R3 razvojni sustav, http://goo.gl/b6j7xS, rujan 2016.

[7] Arduino shieldovi, http://goo.gl/1F3hBu, rujan 2016.

[8] Arduino MEGA 2560, https://goo.gl/VDHf3T, rujan 2016.

[9] ArduCAM Rev.C+ Shield, http://goo.gl/LUZnm6, rujan 2016.

[10] ArduCAM Rev.C+ Shield Data Sheet, http://goo.gl/5f5ZXb, rujan 2016.

[11] OV2640 modul, http://goo.gl/3xBnuZ, rujan 2016.

[12] ESP8266 modul, https://goo.gl/QTzskQ, rujan 2016.

[13] Keyes KY-008 laserski modul, https://goo.gl/PtyADt, rujan 2016.

[14] 3DU5C fototranzistor Data Sheet, http://goo.gl/wm2OEA, rujan 2016.

[15] Micro SD kartice, http://goo.gl/Fg0380, rujan 2016.

[16] Eksperimentalna pločica, http://goo.gl/7NVtR8, rujan 2016.

[17] Određivanje vrijednosti otpora otpornika, http://goo.gl/uFDW75, rujan 2016.

[18] Djelitelj napona, http://goo.gl/FVSygJ, rujan 2016.

[19] Pull-down otpornik, http://goo.gl/5fwJp4, rujan 2016.

[20] Dodavanje nove biblioteke, https://goo.gl/AUzfPl, rujan 2016.

[21] ArduCAM biblioteke, https://goo.gl/fohXjm, rujan 2016.

[22] Smpt2go, http://goo.gl/U8enUh, rujan 2016.

[23] Base64, https://goo.gl/8MMwK, rujan 2016.

[24] Pretvorba u Base64, http://goo.gl/s5VTME, rujan 2016.

[25] Gmail mobilna aplikacija, http://goo.gl/wc9r31, rujan 2016.

[26] Dropbox mobilna aplikacija, http://goo.gl/V653kz, rujan 2016.

SAŽETAK

Naslov: Udaljeni sigurnosni sustav s laserima i kamerom te dojavom na mobitel

U ovom radu izrađen je sustav detekcije i signalizacije pokreta na udaljenoj lokaciji. Sustav je podijeljen na tri dijela: detekcija pokreta, snimanje fotografije i spremanje, te slanje. U radu su razmotrene tehnologije i alati kojima je moguće realizirati sustav, te dijelovi koji su korišteni. Objašnjena je realizacija sustava spajanjem fizičkih komponenti, kao i programskim putem.

Ključne riječi: Arduino, laser, sigurnosni, sustav, kamera, Internet

ABSTRACT

Title: Remote security system with lasers, camera and cell phone alert

In this paper movement detection and signaling system on remote location was developed. System is divided into three parts: movement detection, photo recording and saving, and sending. Technologies and tools with which it is possible to implement this system, as well as parts used were considered in this paper. Implementation of system by wiring physical components, as well as programming was explained.

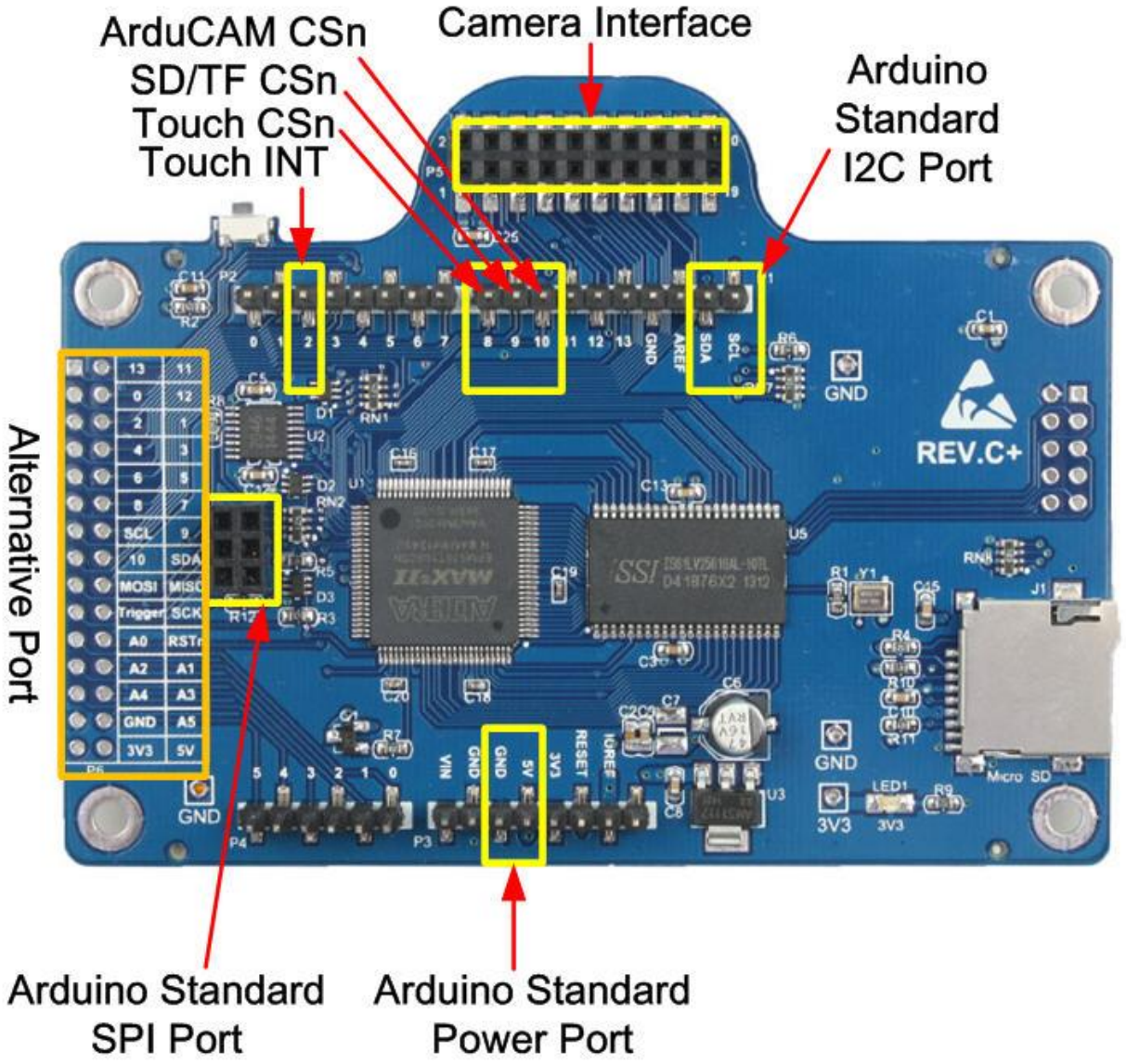
Keywords: Arduino, laser, security, system, camera, Internet

ŽIVOTOPIS

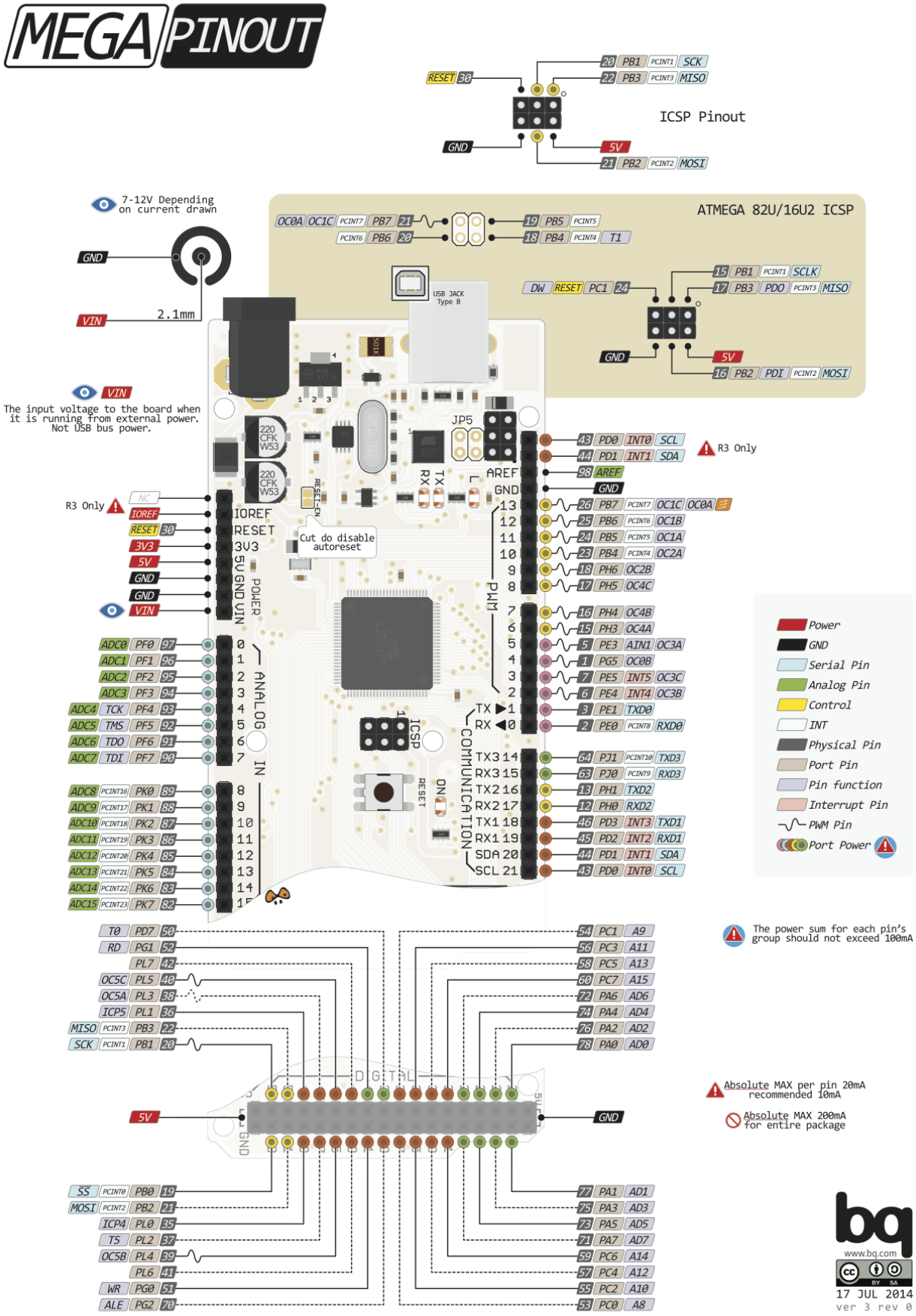
Luka Umiljanović rođen je 21. 11. 1994. u Našicama, gdje živi i danas. Nakon osnovne škole, 2009. upisuje prirodoslovno-matematičku gimnaziju u Srednjoj školi Isidora Kršnjavoga u Našicama, gdje kroz 4 godine uči programiranje u različitim programskim jezicima. Srednju školu završava redovito 2013. godine te u istoj upisuje preddiplomski studij računarstva na Elektrotehničkom fakultetu u Osijeku.

PRILOZI

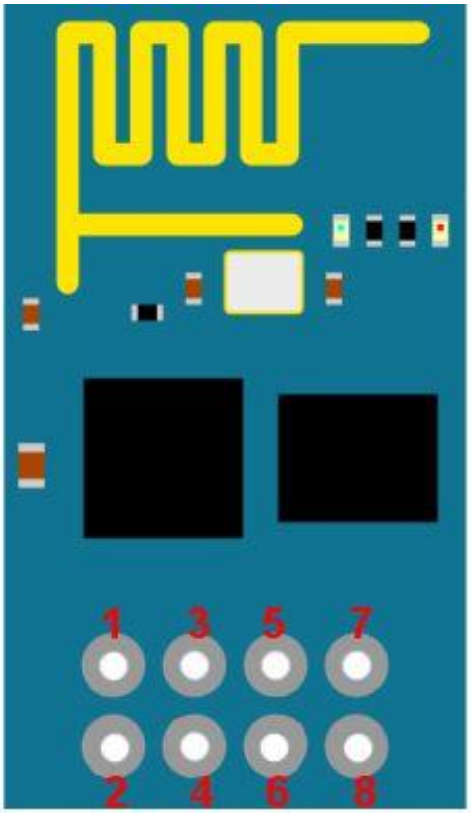
PRILOG A. ArduCAM Rev.C+ pinout



PRILOG B. SainSmart MEGA2560 R3 pinout



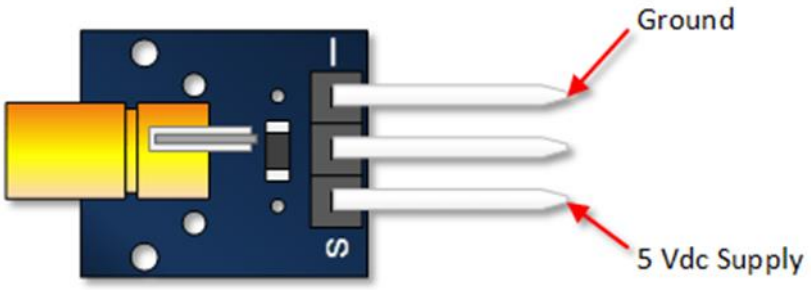
PRILOG C. ESP8266-01 pinout



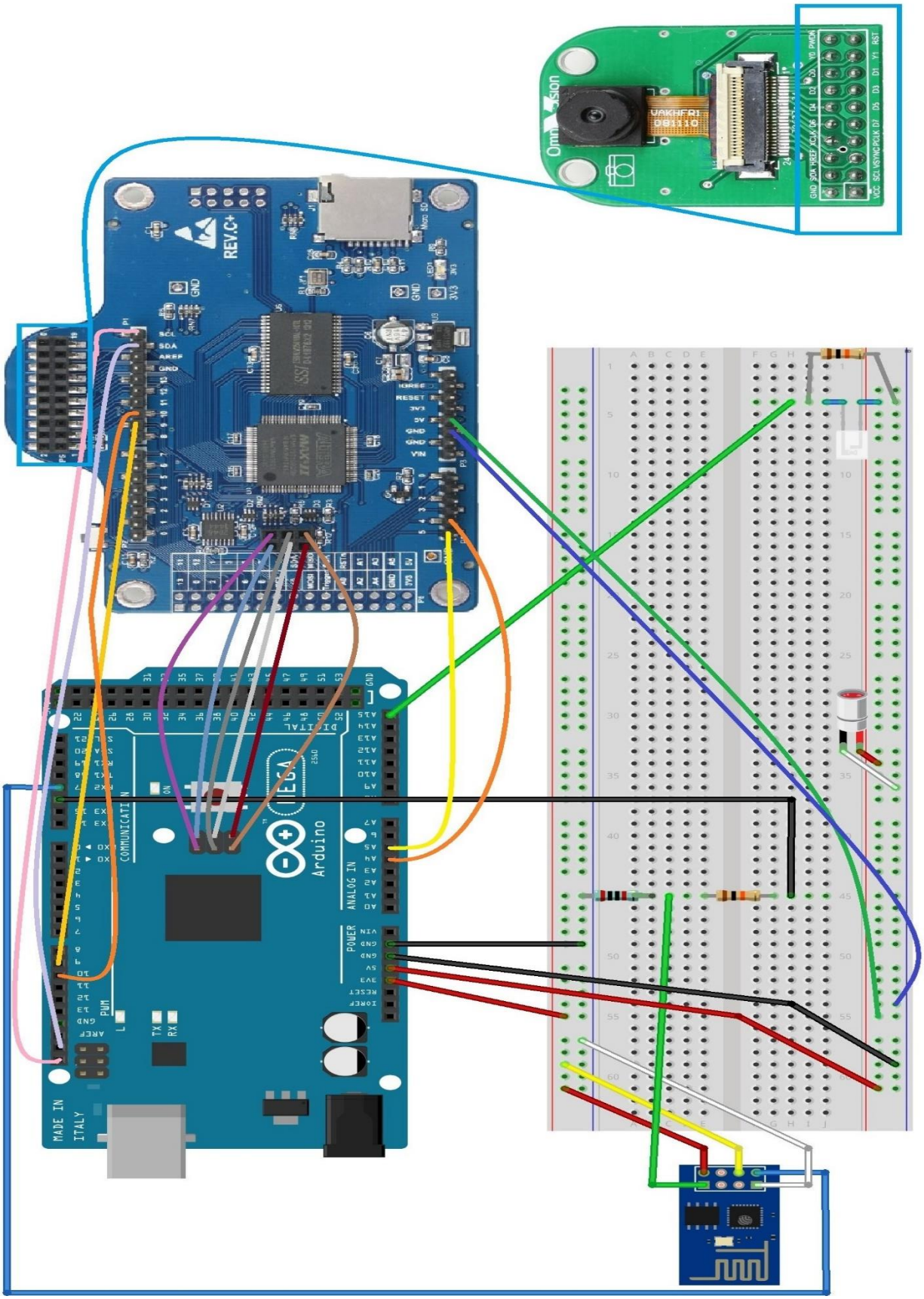
ESP8266 Pins

1. GND - Circuit Ground
2. TX - UART0 Transmit
3. GPIO2 - General Purpose I/O
4. CH_EN - Chip Enable, Active High
5. GPIO0 - General Purpose I/O
6. RESET - Reset, Active Low
7. RX - UART0 Receive
8. VCC - Circuit Power = +3.3V DC

PRILOG D. KY-008 pinout



PRILOG E. Shema cijelog sustava



PRILOG F. Arduino IDE programski kod

```
#include <UTFT_SPI.h>
#include <SD.h>
#include <Wire.h>
#include <ArduCAM.h>
#include <SPI.h>
#include "memorysaver.h"

#if defined(__arm__)
  #include <itoa.h>
#endif

#define SD_CS 9

const int SPI_CS = 10; //postavi pin 10 kao SS signal za SPI

const int sensor = A15;

File photoFile;
int x = 1;
int y = 1;

ArduCAM camDetect(OV2640,SPI_CS);
UTFT glcdDetect(SPI_CS);
boolean noticeFlag = true;

char label[]="AT+CIPSTART=\"TCP\",\"mail.smtp2go.com\",2525";
char *labelPtr;

char labelA[]="AT+CIPSTART=\"TCP\",\"mail.smtp2go.com\",80";
char *labelAPtr;

void setup()
{
  Serial.begin(115200);
  Serial2.begin(115200);

  uint8_t vid,pid;
  uint8_t tmp;
#if defined(__SAM3X8E__)
  Wire1.begin();
#else
  Wire.begin();
#endif

  pinMode(sensor, INPUT);

  pinMode(SPI_CS, OUTPUT);

  Serial2.println("AT");                                    delay(500);
  Serial2.println("AT+CWMODE=1");                           delay(500);
  Serial2.println("AT+RST");                                delay(500);
  Serial2.println("AT");                                    delay(500);
  Serial2.println("AT+CWJAP=\"SSID\",\"PASSWORD\"");        delay(500);
  Serial2.println("AT+CIPMUX=1");                           delay(500);

  labelPtr = label;
  labelAPtr = labelA;
```

```
  SPI.begin();

  //provjera SPI sabirnice:
  camDetect.write_reg(ARDUCHIP_TEST1, 0x55);
  tmp = camDetect.read_reg(ARDUCHIP_TEST1);

  camDetect.set_mode(MCU2LCD_MODE); //promjena MCU načina

  glcdDetect.InitLCD();

  //provjera je li modul kamere OV2640:
  camDetect.rdSensorReg8_8(OV2640_CHIPID_HIGH, &vid);
  camDetect.rdSensorReg8_8(OV2640_CHIPID_LOW, &pid);

  camDetect.set_format(BMP);

  camDetect.InitCAM();

  if (!SD.begin(SD_CS))
  {
    Serial.println("Error with SD Card");
  }
}

void loop()
{
  char str[8];
  byte buff[256];
  static int i = 0, k = 0, n = 0;
  File outputFile;
  uint8_t tmp,tmp_last;
  uint8_t start_capture = 0;

  if(analogRead(sensor)<=500)
  {
    noticeFlag = false;
    camDetect.set_mode(MCU2LCD_MODE);
    camDetect.set_format(JPEG);
    camDetect.InitCAM();

    camDetect.OV2640_set_JPEG_size(OV2640_640x480);

    while(analogRead(sensor)<=500);
    delay(500);
    start_capture = 1;
  }
  else
  {
    if(noticeFlag )
    {
      if(!camDetect.get_bit(ARDUCHIP_TRIG,VSYNC_MASK))               //dolazak  novog
podatkovnog okvira
      {
         camDetect.set_mode(MCU2LCD_MODE);
         glcdDetect.resetXY();
         camDetect.set_mode(CAM2LCD_MODE);
         while(!camDetect.get_bit(ARDUCHIP_TRIG,VSYNC_MASK)); //čeka da VSYNC nestane
      }
```

```
  }
}
if(start_capture)
{
  camDetect.flush_fifo();

  camDetect.clear_fifo_flag();

  camDetect.start_capture();
}
if(camDetect.get_bit(ARDUCHIP_TRIG ,CAP_DONE_MASK))
{
  //stvaranje imena slikovne datoteke:
  k = k + 1;
  itoa(k, str, 10);
  strcat(str,".jpg");

  outputFile = SD.open(str,O_WRITE | O_CREAT | O_TRUNC);

  i = 0;
  tmp = camDetect.read_fifo();

  buff[i++] = tmp;  //zapisuje prve podatke fotografije u međuspremnik

  //čitanje JPEG podataka iz FIFO memorije:
  while( (tmp != 0xD9) | (tmp_last != 0xFF) )
  {
    tmp_last = tmp;
    tmp = camDetect.read_fifo();
    //zapis podataka fotografije u međuspremnik ako nije pun:
    if(i < 256)
      buff[i++] = tmp;
    else
    {
      //zapis 256 bajtova podataka fotografije u datoteku:
      outputFile.write(buff,256);
      i = 0;
      buff[i++] = tmp;
    }
  }
  //zapis ostatka bajtova u međuspremnik:
  if(i > 0)
    outputFile.write(buff,i);

  outputFile.close();

  if (k == 1)
  {
    //Serial.println("1.jpg OK");
    //AT+CIPSEND = number of data bits (including spaces) + 2
    Serial2.println(label);                                    delay(500);
    Serial2.println(labelPtr);                                 delay(500);
    Serial2.println(labelA);                                   delay(500);
    Serial2.println(labelAPtr);                                delay(500);
    Serial2.println("AT+CIPSEND=19");                          delay(500);
    Serial2.println("EHLO 192.168.0.13");                      delay(500);
    Serial2.println("AT+CIPSEND=12");                          delay(500);
    Serial2.println("AUTH LOGIN");                             delay(500);
    Serial2.println("AT+CIPSEND=NUMBER OF USERNAME BASE64 BITS + 2");delay(500);
```

```
      Serial2.println("USERNAME BASE64");                        delay(500);
      Serial2.println("AT+CIPSEND=NUMBER OF PASSWORD BASE64 BITS + 2");delay(500);
      Serial2.println("PASSWORD BASE64");                        delay(500);
      Serial2.println("AT+CIPSEND=NUMBER OF E-MAIL BITS + 2");   delay(500);
      Serial2.println("MAIL FROM:<E-MAIL>");                     delay(500);
      Serial2.println("AT+CIPSEND=NUMBER OF E-MAIL BITS + 2");   delay(500);
      Serial2.println("RCPT To:<E-MAIL>");                       delay(500);
      Serial2.println("AT+CIPSEND=6");                           delay(500);
      Serial2.println("DATA");                                   delay(500);
      Serial2.println("AT+CIPSEND=81");                          delay(500);
      Serial2.println("Motion Detected! First image will be available at Your Dropbox
account shortly.");  delay(500);
      Serial2.println("AT+CIPSEND=3");                           delay(500);
      Serial2.println(".");                                     delay(500);
      Serial2.println("AT+CIPSEND=6 ");                          delay(500);
      Serial2.println("QUIT");                                   delay(500);
    }

    camDetect.clear_fifo_flag();
    start_capture = 0;

    send(); //funkcija koja šalje fotografiju sa SD kartice na računalo

    camDetect.set_format(BMP);
    camDetect.InitCAM();
    noticeFlag = true;
  }
}

void send ()
{
  pinMode(9,OUTPUT);  //SD_CS pin
  while(x==1)
  {
    Serial.flush();
    if(y==1)
    {
      delay(200);
      File photoFile = SD.open("1.jpg");
      if (photoFile)
      {
        while (photoFile.available())
        {
          Serial.write(photoFile.read());
        }
        photoFile.close();
        x=0;
        y=0;
        break;
      }
      else
      {
        x=0;
        y=0;
        break;
      }
    }
  }
}
```

PRILOG G. Processing IDE programski kod

```
import processing.serial.*;

Serial COMport;
OutputStream photograph;

void setup()
{
  size(300, 220);

  COMport = new Serial(this, "COM4", 115200);
  COMport.clear();

  photograph = createOutput("photograph.jpg");
}

void draw()
{
  try
  {
    while (COMport.available()>0)
    {
      photograph.write(COMport.read());
    }
  }
  catch (IOException e)
  {
    e.printStackTrace();
  }
}

void end()   //mouseMoved
{
  try
  {
    photograph.flush();  //zapisuje ostatak podataka u datoteku
    photograph.close();
  }
  catch (IOException e)
  {
    e.printStackTrace();
  }
}
```