Pan-tilt postolje za web kameru

Matijašević, Marin

Master's thesis / Diplomski rad

2021

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:200:372813

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-05-13**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij

Pan-tilt postolje za web kameru

Diplomski rad

Marin Matijašević

Osijek, 2021.

SADRŽAJ

1. UVOD *1*

2. PREGLED PODRUČJA *2*

3. IZRADA PAN-TILT POSTOLJA *3*

3.1. Sklopovlje **3**

3.1.1. Shema sustava 9

3.1.2. Nosač web kamere 12

3.2. Programska podrška mikroupravljača **13**

3.2.1. Programski alati 13

3.2.2. Algoritam i implementacija 13

3.3. Programska podrška računalne aplikacije **22**

3.3.1. Programski alati 22

3.3.2. Algoritam i implementacija 23

3.4. Testiranje **25**

4. ZAKLJUČAK *27*

Literatura *28*

Sažetak *29*

Abstract *29*

Prilozi *30*

1. UVOD

U ovom radu opisan je postupak izrade *pan-tilt* postolja za web kameru kojim korisnik upravlja koristeći računalnu aplikaciju. Svrha *pan-tilt* postolja je omogućiti korisniku pomicanje web kamere po horizontalnoj i vertikalnoj osi pomoću ugrađenih servo motora. Za upravljanje postoljem zadužen je mikroupravljač, a za interakciju korisnika sa postoljem koristi se računalna aplikacija. Komunikacija između mikroupravljač i računalne aplikacije ostvarena je preko žične veze koristeći UART (engl. *Universal Asynchronous Receiver-Transmitter*) komunikaciju.

Nakon što se definira primjena te se odaberu komponente potrebno je napisati programski kôd koji služi za programiranje mikroupravljača kako bi izvršavao svoje zadatke. Opisani su korišteni programski alati i način njihovog korištenja te je detaljno opisan programski kôd i tok izvršavanja pojedinih radnji kao i način programiranja ugrađenog mikroupravljača. Izrađeno je kućište u koje je smješteno napajanje sustava i pločica s komponentama, a na vanjsku stranu kućišta je fiksiran nosač web kamere.

Ostatak rada organiziran je kako slijedi. U poglavlju „Izrada pan-tilt postolja“ opisana je izrada sklopovlja te je dana shema sustava i princip rada nosača web kamere. Navedeni su korišteni programski alati i detaljno je opisan programski kôd korišten za programiranje mikroupravljača. Zatim je u poglavlju „Programska podrška računalne aplikacije“ opisan programski alat korišten za izradu grafičkog sučelja računalne aplikacije kojom se upravlja postoljem te je opisan programski kôd. U poglavlju „Testiranje“ je opisan način upravljanja postoljem koristeći računalnu aplikaciju.

2. PREGLED PODRUČJA

Na tržištu postoji veliki izbor postolja za razne web, sigurnosne i profesionalne kamere koje zahtijevaju stabilnije i preciznije izvedbe. Također se koriste razni načini za kontrolu *pan-tilt* postolja od daljinskih upravljača do postolja koja koriste bežične komunikacijske tehnologije kao što su Bluetooth i Wi-Fi te postolja koja koriste klasični žični način upravljanja. Postolja koja koriste Wi-Fi komunikacijsku tehnologiju za primanje upravljačkih signala najčešće dolaze s već ugrađenom kamerom. Slika s ugrađene kamere se također prenosi koristeći Wi-Fi tehnologiju za razliku od *pan-tilt* postolja koja služe za montiranje web kamera čija se slika najčešće prenosi koristeći USB kabel montirane kamere. Primjeri *pan-tilt* postolja koji za komunikaciju koristi Wi-Fi tehnologiju [1], bežični daljinski upravljač [2] i žični upravljač [3] te njihove karakteristike se nalaze u tablici 2.1.

Tablica 2.1. Karakteristika nekoliko *pan-tilt* postolja.

	Amcrest 4MP [1]	***YT-260 [2]***	***MP101 [3]***
Montiranje kamere	*ugrađena kamera*	*standardni vijak*	*standardni vijak*
Način upravljanja	*Wi-Fi komunikacija*	*daljinski upravljač*	*žični upravljač*
Nosivost postolja	*-*	*260 g*	*2700 g*
Napajanje	*baterija*	*baterija*	*adapter*
Pan kut / tilt kut	*355 / 90 stupnjeva*	*230 / 60 stupnjeva*	*90 / 15 stupnjeva*

Za razliku od *pan-tilt* postolja na koja se montiraju kamere, postoje i PTZ (engl. *pan-tilt zoom*) kamere koje se nalaze u sustavu koji omogućava njihovo pomicanje po horizontalnoj i vertikalnoj osi te približavanje (engl. *zoom*) objekata na slici. Ovakvi sustavi su najčešće upravljani upravljačima koji su žično povezani sa postoljem. Također postoje i VPTZ (engl. *virtual pan-tilt zoom*) kamere koje ne koriste nikakve pomične mehanizme, nego se slika koja dolazi sa širokokutne statične kamere programski obrađuje te se približavaju pojedini dijelovi slike, a potom se pomiče kroz scenu stvarajući *pan-tilt* efekt [4].

3. IZRADA PAN-TILT POSTOLJA

Izrada *pan-tilt* postolja za web kameru započinje definiranjem primjene i odabirom potrebnih komponenti. U ovom slučaju radi se o uređaju koji pomoću mikroupravljača upravlja servo motorima na temelju naredbi dobivenih od osobnog računala kako bi postavio montiranu web kameru u željeni položaj. Pri izboru mikroupravljača treba uzeti u obzir njegove dimenzije, mogućnosti, radnu frekvenciju, ugrađene periferne uređaje i način napajanja.

Za pomicanje web kamere se koriste servo motori kako bi se nosač kamere mogao pomicati po horizontalnoj i vertikalnoj osi. S obzirom na vrstu kamere koja će se montirati potrebno je odabrati servo motore s dovoljno jakim zakretnim momentom za pomicanje i zadržavanje kamere u željenom položaju. Također se mora voditi računa o dimenzijama gotovog sustava i uvjetima u kojima će se koristiti. Na temelju ovih zahtjeva kućište se dimenzionira i izrađuje tako da izdrži uvjete u kojima se koristi.

3.1. Sklopovlje

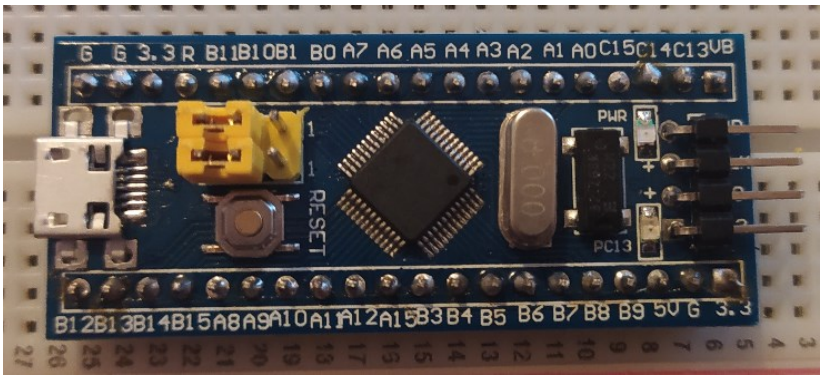
Glavni element sklopovlja je mikroupravljač koji upravlja servo motorima za pomicanje nosača kamere na temelju primljenih naredbi poslanih s osobnog računala koristeći računalnu aplikaciju. Stoga je pri izboru najbitnije da mikroupravljač podržava UART zbog komunikacije s osobnim računalom te da posjeduje mogućnost generiranja pulsno-širinsko moduliranog (PWM – engl. *pulse-width Modulation*) signal koji se koristi za upravljanje servo motorima. Za ovaj sustav odabran je mikroupravljač STM32F103C8T6 proizvođača STMicroelectronics [5] jer posjeduje potrebne karakteristike i malih je dimenzija te je pogodan za ugradnju. Karakteristike odabranog mikroupravljača se nalaze u tablici 3.1.

Tablica 3.1. Karakteristike STM32F103C8T6 mikroupravljača.

Procesor	ARM Cortex - M3
Radna frekvencija	72 MHz
Radna memorija	20 kB
Pohrana	64 kB
Ulazno-izlazne jedinice	37
Komunikacija	UART, SPI, I2C, CAN, USB
Napon napajanja	3.3 - 5 V

Procesor ugrađen u ovaj mikroupravljač pripada skupini ARM (engl. *advanced RISC machines*) procesora sa smanjenim skupom naredbi (RISC – engl. *reduced instruction set computer*). Riječ je o procesorima koji koriste veliki broj registara, paralelno izvršavanje naredbi i korištenje malog broja ciklusa radnog takta za izvršavanje pojedinih naredbi. Ova vrsta procesora zbog svoje jednostavne arhitekture koristi puno manji broj tranzistora u usporedbi s ostalim procesorima opće namjene. Zbog toga na integriranom sklopu ostaje dodatnog prostora koji se koristi za implementaciju perifernih upravljača, uređaja za obradu digitalnih signala i memoriju unutar integriranog sklopa. ARM procesori su zbog malih dimenzija i energetskih zahtjeva često implementirani u raznim ugradbenim sustavima [6].

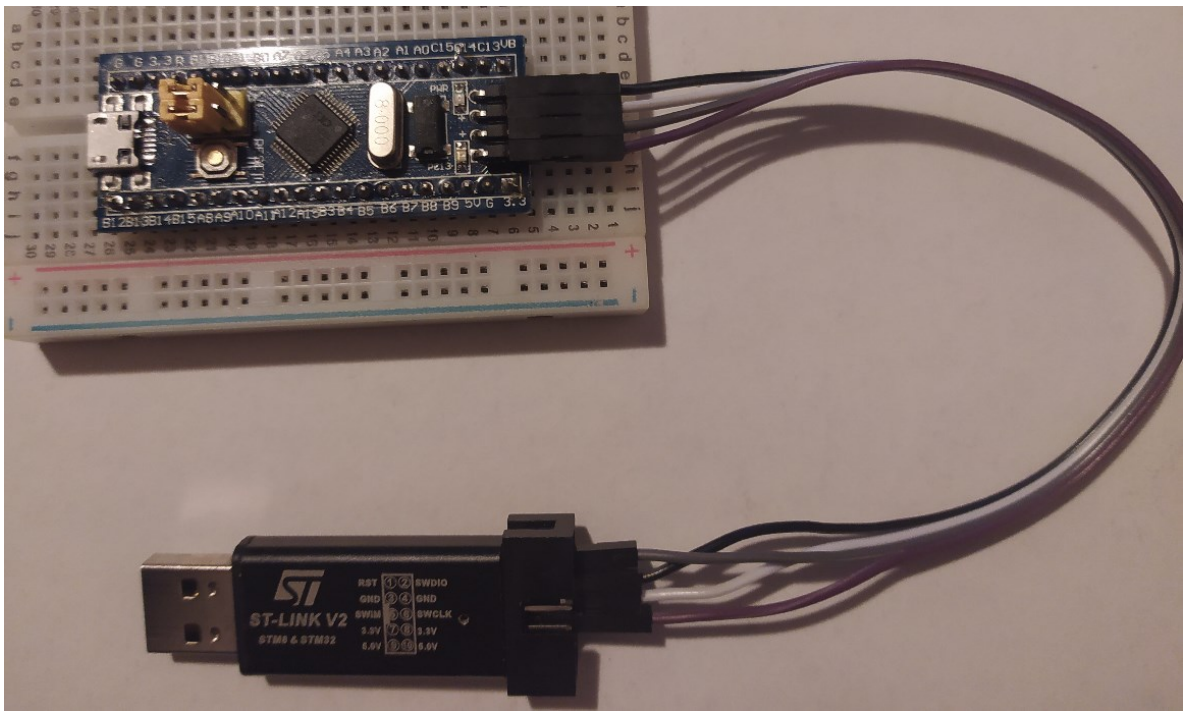
Razvojni sustav u sklopu kojeg je implementiran STM32F103C8T8 mikroupravljač naziva „Blue pill“ [7] posjeduje četrdeset nožica za interakciju s vanjskim uređajima i četiri nožice za povezivanje uređaja za programiranje (slika 3.1). Na pločici se uz mikroupravljač nalaze oscilatori koji pružaju potreban radni takt, tipka za ponovno pokretanje, svjetleće diode koje indiciraju stanje sustava i premosnici čijim prespajanjem se mijenja vrsta memorije unutar integriranog sklopa na koju će programski kôd biti spremljen. Program se može spremati u sistemsku radnu memoriju ili u memoriju za pohranu (engl. *flash*). Napajanje sustava se ostvaruje preko nožica koje su za to predviđene ili preko USB utičnice.



Sl. 3.1. Blue pill razvojni sustav.

Za programiranje STM mikroupravljača koristi se ST-Link adapter koji se povezuje na posebne priključke mikroupravljača za programiranje (slika 3.2) te ga povezuje sa računalom preko USB utora. Koristi se serijska komunikacija preko linija koje služe za dobivanje radnog takt, slanje podataka, napajanje i uzemljenje. Kad se programski kôd pošalje mikroupravljaču ST-Link adapter se može odvojiti, a program ostaje spremljen u unutarnjoj memoriji mikroupravljača.

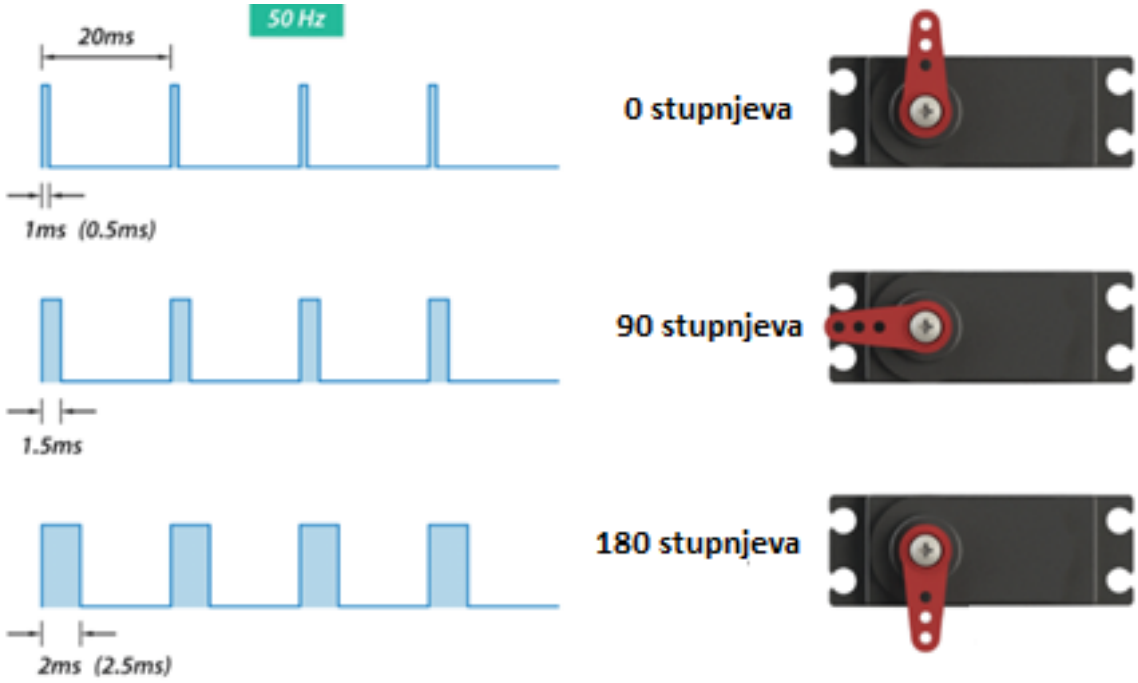
Kako bi se postolje pomicalo po horizontalnoj i vertikalnoj osi potrebna su dva servo motora koja imaju mogućnost pomicanja osi za određeni kut. Za ovaj sustav odabrani su servo motori Tower Pro MG996R [8] (slika 3.3) koji imaju mogućnost zakretanja za 180 stupnjeva. Motor se s mikroupravljačem povezuje koristeći vodič koji služi za primanje upravljačkog signala, te vodiče za povezivanje sa zasebnim napajanjem jer mikroupravljač ne može osigurati dovoljno veliku struju za pomicanje motora. Navedeni servo motor ima mogućnost zadržavanja tereta do 10 kg pri 5 V istosmjernog napona. Za napajanje motora se koristi napon od 5V, a za slanje upravljačkih signala se koristi pulsno-širinska modulacija.



Sl. 3.2. Povezivanje ST-Link adaptera s mikroupravljačem.

Sl. 3.3. Servo motor MG996R

Servo motori su upravljani PWM signalom frekvencije 50 Hz i periodom od 20 ms, na način da kada je dovedeni signal u stanju logičke jedinice u trajanju od 1 ms motor pomiče svoju os na početnu poziciju (0 stupnjeva), a kako se produžava stanje logičke jedinice signala do 2 ms tako se os motora zakreće do maksimalno 180 stupnjeva (slika 3.4.) [9]. Na ovaj način se kontroliraju osi servo motora kako bi pomaknuli nosač kamere u određenom smjeru. Korak pomicanja servo motora može se programski definirati kako bi pomicanje nosača kamere bilo glatko.

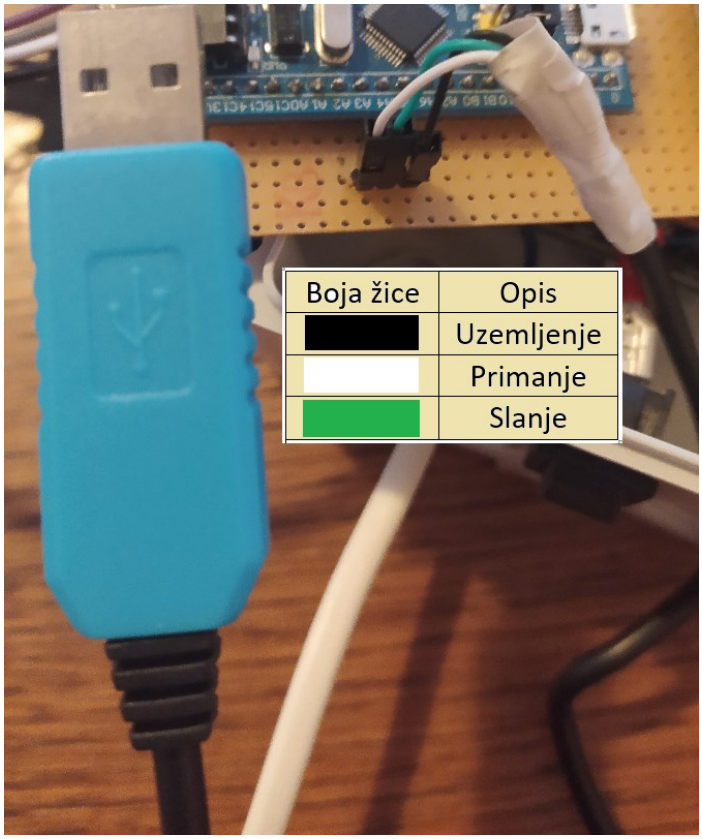


Sl. 3.4. Upravljanje servo motorom.

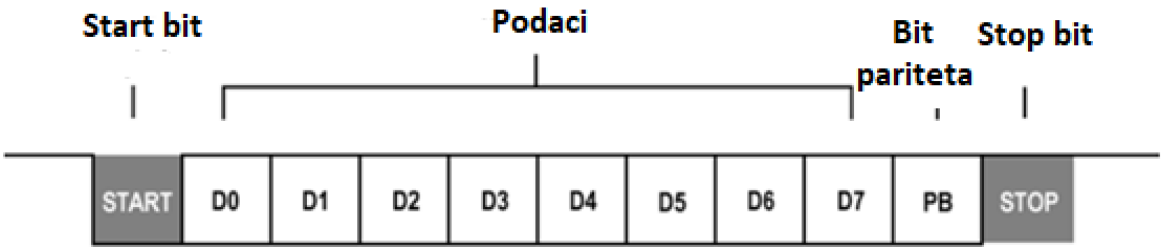
Za komunikaciju s računalnom aplikacijom koristi se pretvornik za serijsku komunikaciju PL2303TA [10] koji s jedne strane ima USB utikač, a s druge strane 4 vodiča (2 za slanje i primanje podataka te uzemljenje i 5 V napajanje). Vodiči se spajaju na priključke mikroupravljača za slanje i primanje podataka te uzemljenje, a napajanje od 5 V nije potrebno jer se pretvornik napaja preko USB utora računala (slika 3.5). Svrha ovog pretvornika je pretvaranje serijske komunikacije koju pruža USART (engl. *universal synchronous asynchronous receiver transmitter*) uređaj mikroupravljača u signale pogodne za primanje preko USB utora osobnog računala.

UART [11] komunikacija je razvijena u svrhu povezivanja raznih uređaja s računalom i temelji se na serijskoj komunikaciji koja šalje bit po bit informacije. Kako se radi o asinkronoj komunikaciji za rad nije potrebna sinkronizacija takta uređaja koji šalje i koji prima podatke. Povezuje uređaje koristeći dvije linije tako da jedna služi za slanje podataka, a druga za primanje. Početak i kraj

komunikacije definirani su sa posebnim start i stop bitovima koji se nalaze na početku odnosno na kraju niza poslanih bitova (slika 3.6) .



Sl. 3.5. Povezivanje PL2303TA pretvornika.

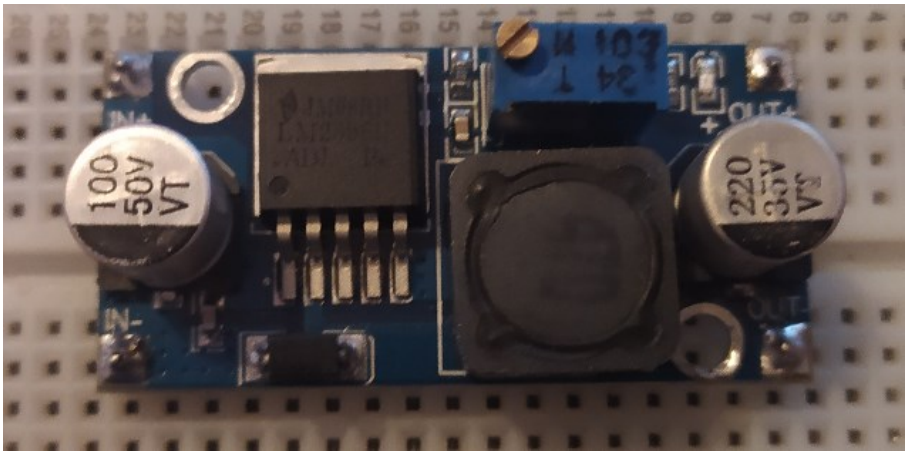


Sl. 3.6. UART komunikacija.

Za napajanje *pan-tilt* postolja potrebne su dvije istosmjerne naponske razine kako bi se napajao mikroupravljač i servo motori. Za ostvarivanje toga se koristi ispravljač (slika 3.7) koji izmjenični napon od 230 V iz mreže pretvara u istosmjerni napon od 5 V koji se koristi za napajanje servo motora. Zatim se istosmjerni napon dodatno snižava na 3.3 V za napajanje mikroupravljača. Odabrani ispravljač napona pruža maksimalnu istosmjernu struju od 5A što je potrebno kako bi se servo motori mogli zadržati u određenom položaju .

Istosmjerni pretvarač napona koji se koristi u ovom sustavu je baziran na LM2596 [12] (slika 3.8) regulatoru istosmjernog napona koji kao ulaznu vrijednost prima istosmjerni napon do maksimalno 40 V, a na izlazu daje vrijednost napona koja je podešena na potenciometru koji se nalazi na pločici naponskog pretvarača.

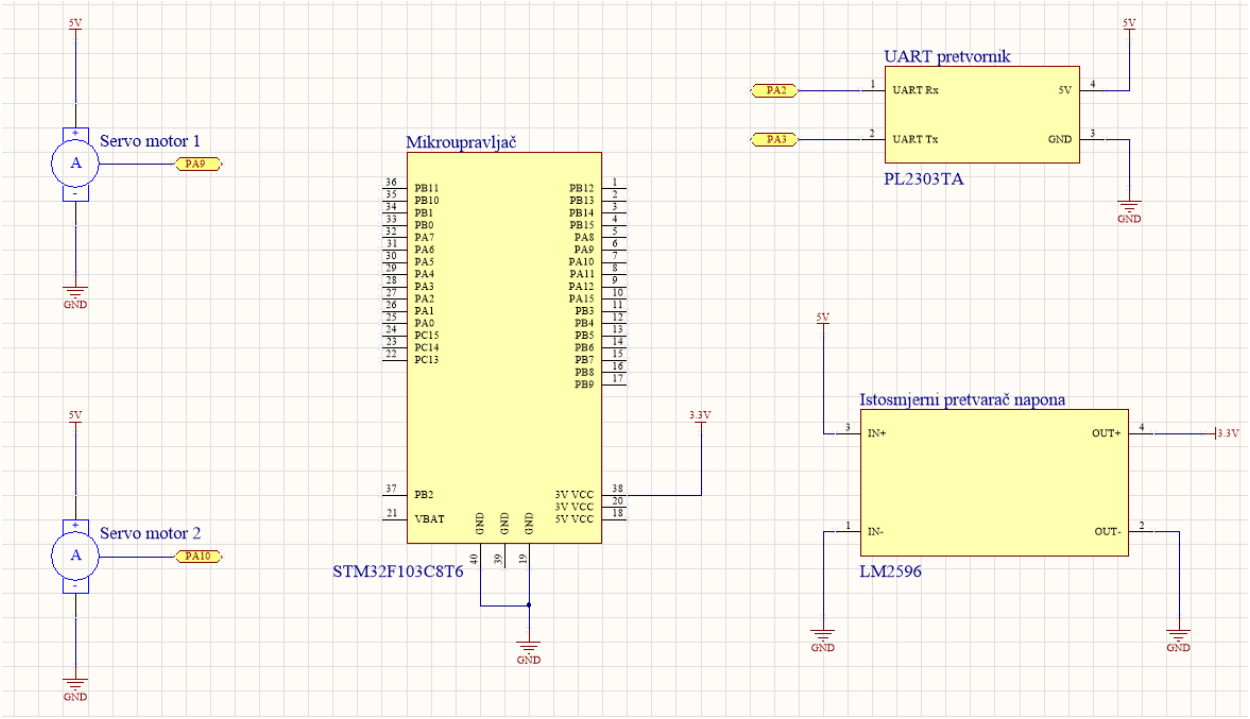
Sl. 3.7. Istosmjerno napajanje.



Sl. 3.8. Istosmjerni pretvarač napona LM2596.

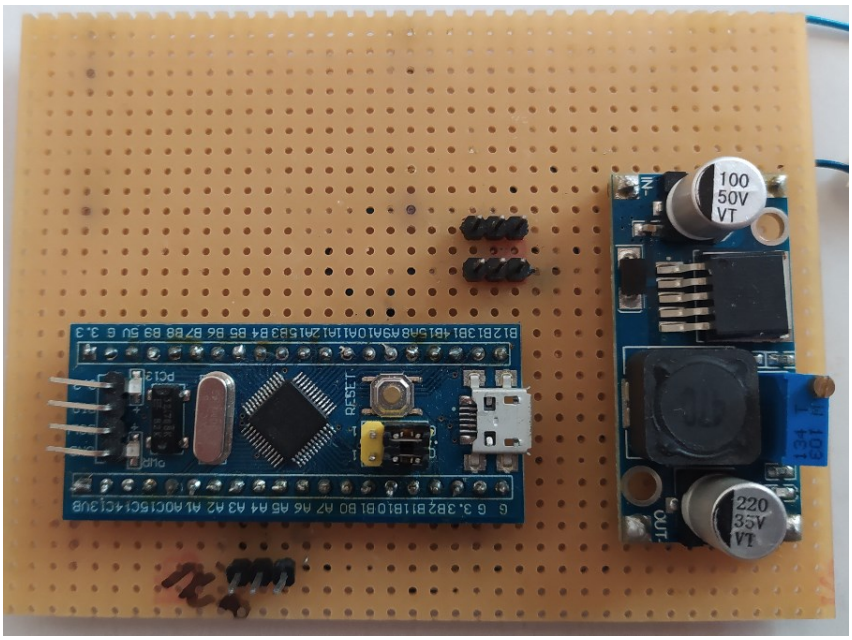
3.1.1. Shema sustava

Na priključke mikroupravljača A9 i A10 koji se koriste za generiranje PWM signala spojeni su servo motori te im se predaje upravljački signal. Pretvornik PL2303TA koji se koristi za UART komunikaciju između računalne aplikacije i mikroupravljača spojen je na priključke A2 i A3 koji se koriste za slanje i primanje podataka. S druge strane je pretvornik spojen s osobnim računalom preko USB utora. Za napajanje sustava koristi se ispravljač izmjeničnog napona koji daje istosmjerni napon od 5 V kojim su napajani servo motori. Ovaj napon se dodatno snižava na 3.3 V koristeći istosmjerni pretvarač napona LM2596 za napajanje mikroupravljača. Sve elektroničke komponente spojene su na zajedničko uzemljenje (GND – engl. *ground*) (slika 3.9).

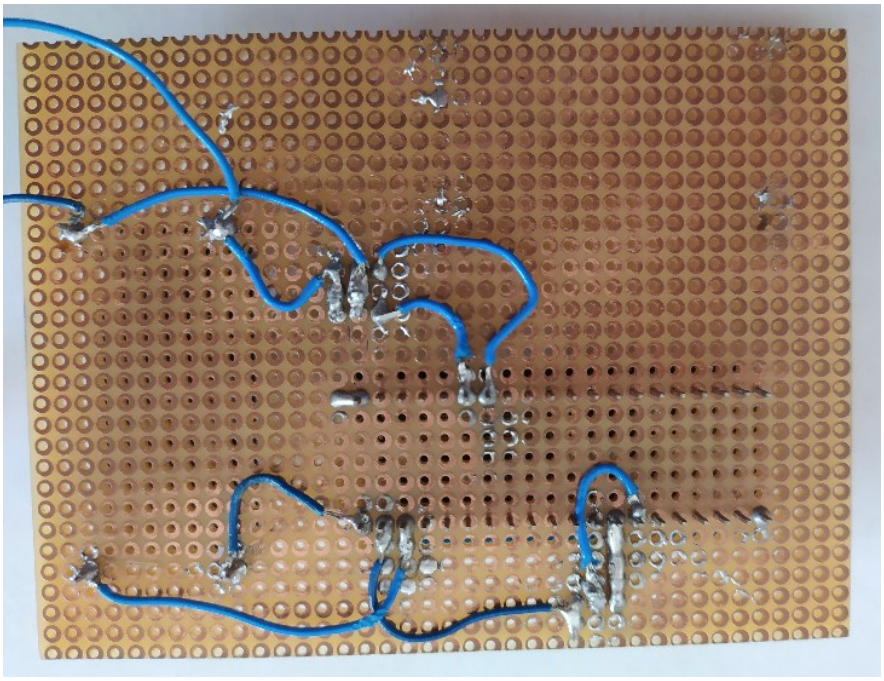


Sl. 3.9. Shema povezivanja komponenti.

Korištene komponente povezane su u cjelinu s mikroupravljačem koristeći prototipnu pločicu (slika 3.10 i 3.11). Komponente se nalaze sa gornje strane pločice, a vodiči za povezivanje s donje strane. Ovaj proces se sastoji od više koraka od kojih je prvi postavljanje elektroničkih komponenti na odgovarajuće pozicije, lemljenja te na kraju lakiranja kako bi se spriječilo oksidiranje bakrenih spojeva.



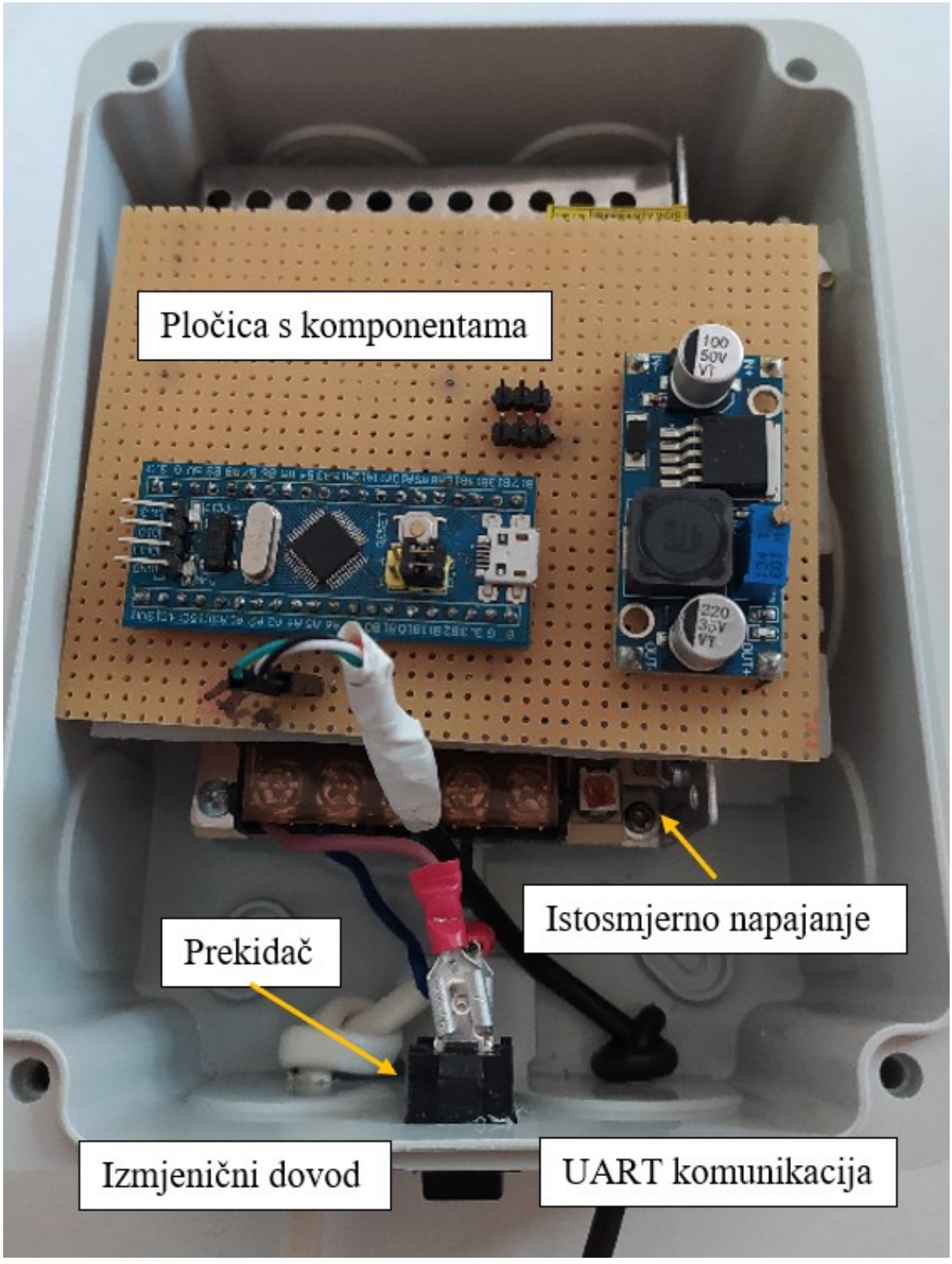
Sl. 3.10. Pločica sa komponentama – gornja strana.



Sl. 3.11. Pločica sa komponentama – donja strana.

Kako bi se sustav povezao u funkcionalnu cjelinu potrebno je kućište čija je funkcija da zaštiti sve komponente od vanjskih utjecaja. Unutar kućišta su smješteni pojedini dijelovi sustava prema slici 3.12. Kućište također služi kao postolje nosača na koji se montira web kamera i tako osigurava njezinu stabilnost. Stoga izvedba mora biti takva da se postolje ne prevrne prilikom pomicanja kamere. S bočne strane plastičnog kućišta se nalazi prekidač za uključivanje i isključivanje

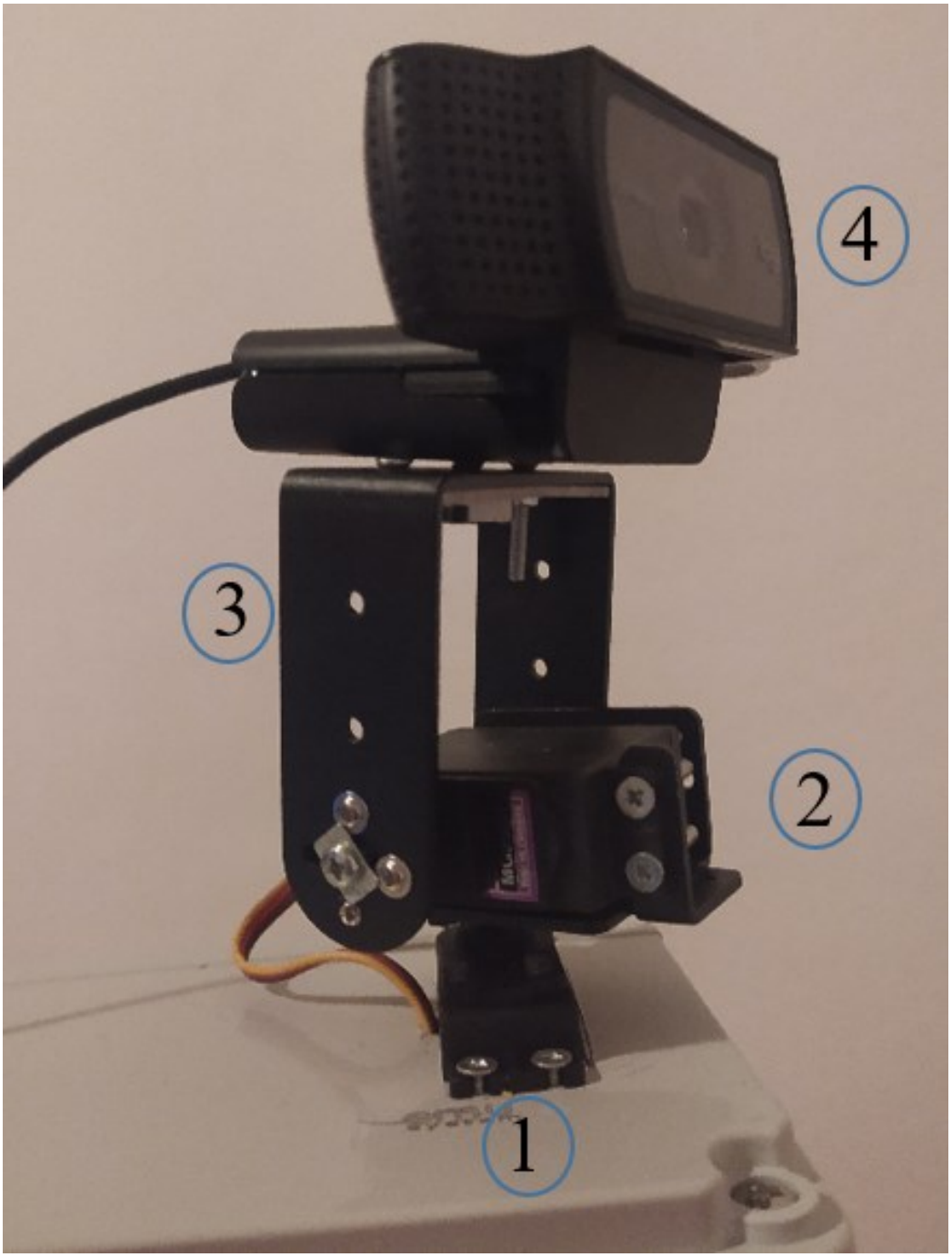
sustava, a iz kućišta izlaze dva kabela od kojih je jedan dovod izmjeničnog napona, a drugi služi za komunikaciju s osobnim računalom.



Sl. 3.12. Komponente unutar kućišta.

3.1.2. Nosač web kamere

Nosač web kamere (slika 3.13) se sastoji od dva servo motora od kojih je prvi (1) fiksiran za poklopac postolja, dok je drugi (2) montiran na osovinu prvog motora i pomiče ga po horizontalnoj osi. Na drugi servo motor (2) pričvršćen je metalni nosač kamere (3) na koji se uz pomoć posebnog vijka standardizirane veličine montira web kamera (4) te se servo motor pomiče po vertikalnoj osi. Na ovaj način je ostvareno *pan-tilt* pomicanje web kamere.



Sl. 3.13. Nosač web kamere sa servo motorima.

3.2. Programska podrška mikroupravljača

Kako bi mikroupravljač dobio svoju funkciju potrebno je napisati programski kôd kojim će biti programiran. Za programiranje ugradbenih sustava najčešće se koristi C programski jezik ili inačica ovog programskog jezika za ugradbene sustave (engl. *embedded C*) koji je posebno prilagođen za ovu primjenu. U ovom slučaju se koristi ugradbeni C programski jezik kako je opisano u ovom poglavlju.

3.2.1. Programski alati

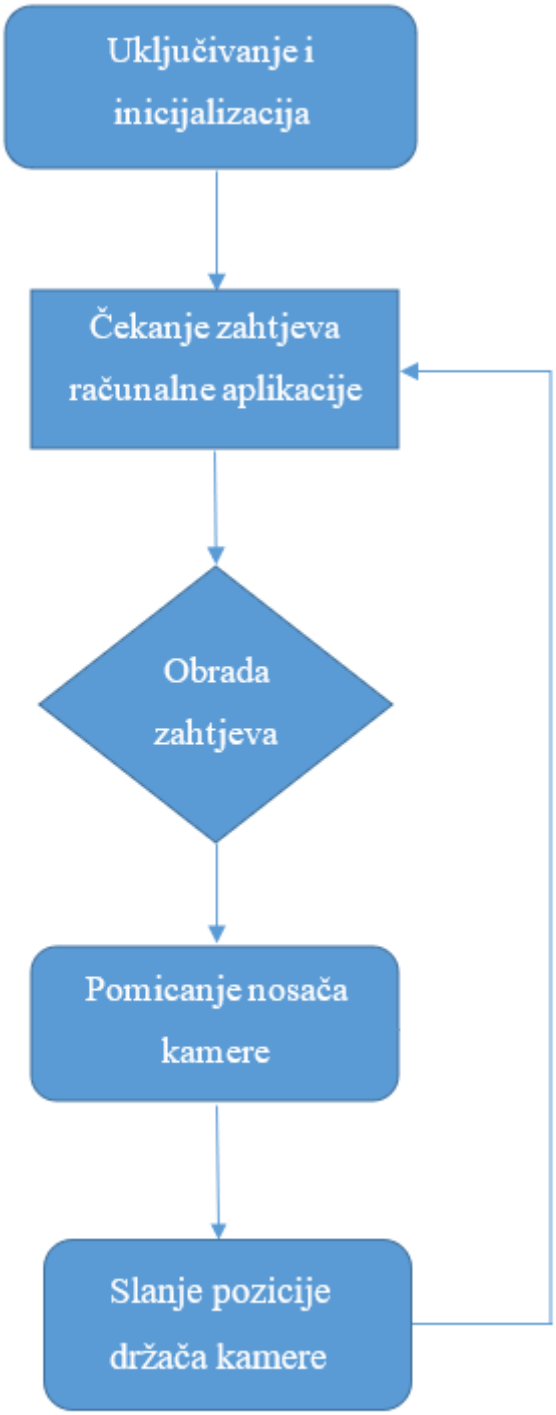
Računalni program korišten za pisanje ugradbenog C programskog jezika kojim se programira mikroupravljač je Atolic TrueSTUDIO [13] koji omogućava pisanje kôda u C ugradbenom programskom jeziku i upotrebu standardnih biblioteka potrebnih za rad mikroupravljača. Program je posebno prilagođen programiranju mikroupravljača tvrtke STMicroelectronics.

U izborniku Atolic TrueSTUDIO programa se odabire mikroupravljač koji je potrebno programirati te program generira biblioteke u kojima se nalaze osnovne funkcije potrebne za rad odabranog mikroupravljača. Funkcije koje se nalaze u bibliotekama se koriste u kombinaciji s vlastitim programskim kôdom za programiranje ponašanja mikroupravljača. Ovaj program također posjeduje ugrađeno sučelje za otkrivanje grešaka (engl. *debbuger*) koje omogućava zaustavljanje programa u određenom koraku i praćenje stanja unutarnjih registara mikroupravljača.

3.2.2. Algoritam i implementacija

U ovom poglavlju je detaljno opisan algoritam programa mikroupravljača i dijagram toka izvršavanja programa. Opisane su pojedine linije programskog kôda i njihova funkcija. Na slici 3.14 prikazan je dijagram toka izvođenja programa mikroupravljača.

Nakon pokretanja sustava mikroupravljač inicijalizira parametre i postavi nosač web kamere u nulti položaj. Zatim prima naredbe koje će računalna aplikacija na zahtjev korisnika poslati preko UART sučelja. Naredbe koje se primaju su smjerovi u kojima korisnik želi pomaknuti nosač web kamere. Nakon što mikroupravljač primi naredbu obradi je te generira signal kojim se pomiču servo motori u nosaču web kamere. Zatim preko UART komunikacije šalje kutove u odnosu na vertikalnu i horizontalnu os na kojem se nalaze servo motori kako bi korisnik računalne aplikacije u svakom trenutku imao informaciju o položaju držača web kamere.



Sl. 3.14. Tok programa mikroupravljača.

STM32 mikroupravljač za upravljanje servo motorima generira pulsno-širinsko modulirani signal na ulazno-izlaznim priključcima A9 i A10 koji podržavaju generiranje PWM signala. Za kreiranje PWM signala na željenim priključcima mikroupravljača potrebno je omogućiti radni takt za

korištene priključke i za ugrađeni brojač (engl. *timer*) (slika 3.15) pomoću kojeg se postavlja vrijeme u kojem je PWM signal u stanju logičke jedinice odnosno nule.

Linija ***Kôd***

```
154:    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA |
        RCC_APB2Periph_AFIO | RCC_APB2Periph_TIM1, ENABLE);
```

Sl. 3.15. Postavljanje radnog takta za priključke i vremenski brojač.

Priključci mikroupravljača postavljaju se u način rada koji omogućava generiranje promjenjivih signala (engl. *alternate function*), definiraju se priključci koji podržavaju generiranje PWM signala (A9 i A10) i postavlja se radna brzina priključaka. Konfiguracijski podaci se spremaju u strukturu te se struktura predaje ugrađenoj funkciji koja ih zapisuje u odgovarajuće registre mikroupravljača (slika 3.16).

Linija ***Kôd***

```
158:    GPIO_InitTypeDef GPIO_InitStructurePWM;
159:    GPIO_InitStructurePWM.GPIO_Mode = GPIO_Mode_AF_PP;
160:    GPIO_InitStructurePWM.GPIO_Pin = PWM_GPIO_PAN | PWM_GPIO_TILT;
161:    GPIO_InitStructurePWM.GPIO_Speed = GPIO_Speed_50MHz;
162:    GPIO_Init(GPIOA, &GPIO_InitStructurePWM);
```

Sl. 3.16. Konfiguriranje priključaka za upravljanje generiranjem PWM signala.

Za generiranje PWM signala koriste se ugrađeni napredni brojači vremena unutar mikroupravljača koji su 16-bitni te mogu brojati do maksimalne vrijednosti koja iznosi 65536. Brojaču se dovodi glavni signal takta iznosa 72 Mhz te se konfigurira da broji od nula do vrijednosti postavljene u registru. Za upravljanje servo motorima potrebno je generirati PWM signal frekvencije 50 Hz čiji je period 20 ms. Kako bi se ostvario PWM signal željene frekvencije ulazni takt od 72 Mhz koji je doveden brojaču se dijeli s odabranim brojem kako bi brojač generirao PWM signal perioda 20 ms. Kao djelitelj frekvencije (engl. *prescaler*) je odabiran broj 22 i uvršten u formulu 3.1. :

$$TIM_{Period} = T_{PWM} * \frac{f_{ul}}{TIM_{Prescaler}} \tag{3.1.}$$

gdje je: *TIM_period* – vrijednost do koje brojač broji, *T_PWM* – period PWM signala, *f_ul* – frekvencija koja se dovodi vremenskom brojaču, *TIM_Prescaler* – djelitelj frekvencije brojača.

Dobivena je vrijednost 65454 što je najbliže maksimalnom broju do kojeg 16-bitni brojač može brojati. Ova vrijednost i vrijednost dijelitelja frekvencije se zapisuje u registre vremenskog brojača (slika 3.17).

Linija ***Kôd***

```
167:    TIM_BaseStruct.TIM_Period = 65454;
168:    TIM_BaseStruct.TIM_Prescaler = 22;
```

Sl. 3.17. Postavke vremenskog brojača.

Zatim se brojač postavlja u PWM način rada tako da je početna vrijednost radnog ciklusa (engl. *duty cycle*) signala nula (slika 3.18) te će se kasnije u glavnom dijelu programa mijenjati sukladno smjeru u kojem je potrebno zakrenuti servo motore.

Linija ***Kôd***

```
172:    TIM_OCInitTypeDef TIM_OCStruct;
173:    TIM_OCStruct.TIM_OCMode = TIM_OCMode_PWM1;
174:    TIM_OCStruct.TIM_OutputState = TIM_OutputState_Enable;
175:    TIM_OCStruct.TIM_OCPolarity = TIM_OCPolarity_High;
176:    TIM_OCStruct.TIM_Pulse = 0;
```

Sl. 3.18. Postavke brojača za generiraje PWM signala.

Potrebno je omogućiti kanale 2 i 3 (Slike 3.19 i 3.20) brojača koji su unutar korištenog mikroupravljača povezani s priključcima A9 i A10. Na ovaj način se generirani PWM signal dovodi na potrebne izlazne priključke.

Linija ***Kôd***

```
179:    TIM_OC2Init(TIM1, &TIM_OCStruct);
```

Sl. 3.19. Omogućavanje kanala 2 brojača.

```
183:    TIM_OC3Init(TIM1, &TIM_OCStruct);
```

Sl. 3.20. Omogućavanje kanala 3 brojača.

Za komunikaciju mikroupravljača s računalnom aplikacijom koristi se UART komunikacijski protokol koji je implementiran u ugrađenom USART upravljaču. Radi se o perifernom uređaju koji omogućava potpunu dvosmjernu serijsku sinkronu i asinkronu komunikaciju koristeći priključke mikroupravljača. Šalju se podaci duljine 8 bitova, a uz njih se šalju start i stop bitovi koji indiciraju početak i kraj serijske komunikacije. USART upravljač generira zastavice koje indiciraju obavljanje slanja i primanja podataka.

Za omogućavanje primanja i slanja podataka koristeći serijsku komunikaciju potrebno je omogućiti radni takt za USART upravljač i priključke koji se koriste za primanje i slanje podataka (slika 3.21). Ove vrijednosti se zatim zapisuju u registre mikroupravljača koristeći ugrađene funkcije.

Linija ***Kôd***

```
81:     RCC_APB1PeriphClockCmd(RCC_APB1Periph_USART2, ENABLE);
82:     RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA, ENABLE);
```

Sl. 3.21. Postavljanje radnog takta za USART i priključke.

Potrebno je konfigurirati USART periferni uređaj tako da se postavi brzina prijenosa jednog simbola po sekundi (engl. *baud rate*), definirati veličinu podataka dovoljnu za slanje podatka veličine 8 bitova te postavi način rada koji omogućava primanje i slanje podataka (slika 3.22). Zatim se podaci zapisuju u odgovarajuće registre mikroupravljača i pokreće se UART komunikacija.

Linija ***Kôd***

```
84:     USART2_InitStructure.USART_BaudRate = 115200;
85:     USART2_InitStructure.USART_WordLength = USART_WordLength_8b;
86:     USART2_InitStructure.USART_StopBits = USART_StopBits_1;
87:     USART2_InitStructure.USART_Parity = USART_Parity_No;
88:     USART2_InitStructure.USART_HardwareFlowControl =
        USART_HardwareFlowControl_None;
89:     USART2_InitStructure.USART_Mode = USART_Mode_Rx |
        USART_Mode_Tx;
90:     USART_Init(USART2, &USART2_InitStructure);
91:     USART_Cmd(USART2, ENABLE);
```

Sl. 3.22. Postavke UART komunikacije u mikrokontroleru.

Postavljeni su priključci korišteni za UART komunikaciju. Priključak koji mikroupravljaču služi za slanje podataka (A2) se konfigurira kao ulaz tako da podržava generiranje promjenjivog signala (engl. *alternate function*) maksimalne frekvencije 10 Mhz (slika 3.23). Na isti način se konfigurira i priključak za slanje podataka (A3) s time da je postavljen kao izlazni.

Linija ***Kôd***

```
94:       PA2_InitStruct.GPIO_Mode = GPIO_Mode_AF_PP;
95:       PA2_InitStruct.GPIO_Pin = GPIO_Pin_2;
96:       PA2_InitStruct.GPIO_Speed = GPIO_Speed_10MHz;
97:       GPIO_Init(GPIOA, &PA2_InitStruct);
```

Sl. 3.23. Konfiguracija priključka za slanje podataka.

Kako bi se primljeni podaci preko UART komunikacije spremali u međuspremnik (engl. *buffer*) koristi se DMA (engl. *Direct Memory Access*) upravljač koji je ugrađen u mikroupravljaču. Ovaj upravljač omogućava direktan pristup memoriji mikroupravljača gdje se nalazi definirani međuspremnik. DMA upravljač povezuje odabrani periferni uređaj s memorijom mikroupravljača i omogućava dvosmjerni prijenos podataka.

Konfiguriranje DMA upravljača obavlja se direktnim zapisivanjem vrijednosti u registre mikroupravljača kako bi se prikazao i ovaj način programiranja mikroupravljača bez korištenja funkcija iz dostupnih biblioteka. Kako se radi o još jednom perifernom uređaju također mu se dovodi radni takt tako da se u registar koji određuje korišteni takt zapiše konfiguracijski bit koristeći ILI operator (slika 3.24).

Linija ***Kôd***

```
112:      RCC->AHBENR |= RCC_AHBENR_DMA1EN;
```

Sl. 3.24. Radni takt za DMA upravljač.

DMA upravljač se sastoji od dva bloka, a svaki ima određen broj kanala koji se mogu koristiti za određene periferne uređaje kao što je opisano u priručniku korištenog mikroupravljača. U ovom slučaju je potrebna komunikacija između USART periferije i memorije te se koristi prvi blok (DMA1) i kanal 6 (slika 3.25) preko kojeg se podaci primljeni preko USART periferije spremaju direktno u memoriju, odnosno kanal 7 (slika 3.26) preko kojeg se podaci iz memorije dostavljaju USART periferiji za slanje. DMA periferija mora znati u kojem smjeru se šalju podaci, dali je to iz memorije do perifernog uređaja ili obrnuto pa se postavlja vrijednost *DMA_CCR7_DIR*.

DMA_CCR_MINC vrijednost postavlja da se memorijska adresa svakim spremanjem povećava kako bi se izbjeglo pisanje preko prethodno unesenog podatka u memoriju dok se ne dođe do kraja zadanog polja u memoriji. Posljednji bit omogućava DMA periferija da generira zastavicu kada je prijenos podataka gotov.

Linija ***Kôd***

```
115:    DMA1_Channel6->CCR |= DMA_CCR6_MINC | DMA_CCR6_TCIE;
```

Sl. 3.25. Konfiguracija DMA kanala za primanje podataka.

Linija ***Kôd***

```
118:    DMA1_Channel7->CCR |= DMA_CCR7_MINC | DMA_CCR7_DIR |
        DMA_CCR7_TCIE;
```

Sl. 3.26. Konfiguracija DMA kanala za slanje podataka.

Kako bi se DMA kontroler koristio za primanje i slanje podataka preko USART periferije napisane su dvije funkcije koje se koriste u glavnom dijelu programa. Prva funkcija služi za primanje podataka i nazvana je *Usart2_DMA_Receive()* te kao argumente prima adresu polja podataka u memoriji kojim je definiran međuspremnik i broj podataka koji se spremaju u memoriju. Omogućava se interakcija USART periferije i DMA kontrolera koristeći ugrađenu *USART_DMACmd()* funkciju. Zatim se pokreće prijenos primljenog podatka u memoriju i kada se prijenos obavi prijenos se zaustavi dok idući podataka nije primljen preko UART-a (slika 3.27).

Linija ***Kôd***

```
123:    DMA1_Channel6->CNDTR = nbr_of_char;
125:    DMA1_Channel6->CPAR = (uint32_t)&USART2->DR;
126:    DMA1_Channel6->CMAR = (uint32_t)buffer;
128:    USART_DMACmd(USART2, USART_DMAReq_Rx, ENABLE);
130:    DMA_Cmd(DMA1_Channel6, ENABLE);
131:    while(!DMA_GetFlagStatus(DMA1_FLAG_TC6));
132:    DMA_ClearFlag(DMA1_FLAG_TC6);
133:    DMA_Cmd(DMA1_Channel6, DISABLE);
```

Sl. 3.27. Funkcija za slanje podataka koristeći DMA i USART.

Funkcija za slanje podatka preko USART upravljača koristeći DMA upravljač je konfigurirana na isti načina kao i funkcija za primanje podataka s time da je promijenjen samo smjer prijenosa tako da se podaci za prijenos iz memorije šalju USART perifernom uređaju te ih on zatim koristeći UART komunikaciju šalje računalnoj aplikaciji.

Funkcija *pomicanjeMotora*() se koristi za obradu primljenih naredbi od računalne aplikacije i izračun vrijednosti koja se koristi za generiranje PWM signala koji pomiču servo motore u zadani smjer. Vrijednost koja je zapisana u CCR (engl. capture compare register) registar koji je odgovoran za kreiranje radnog ciklusa kojim se upravljaju servo motori je u rasponu od 2200 do 6800. Vrijednost 2200 stvara radni ciklus PWM signala u trajanju 1 ms što servo motor postavlja u početnu poziciju, a vrijednost 7000 stvara radni ciklus trajanja 2 ms te se servo motor zakreće za 180 stupnjeva od početne pozicije (vidi sliku 3.4.). Ukoliko se primi naredba za pomicanje nosača kamere u desnu stranu za jedan stupanj čemu odgovara primljeni znak „R“ vrijednosti koja generira PWM signal se dodaje vrijednost koja odgovara pomaku za 1 stupanj. Kad se dosegne maksimalni pomak servo motora u desnu stranu prestaje se dodavati vrijednost i pomicati motor (slika 3.28).

Linija ***Kôd***

```
55:     if(buffer[0] == 'R' && TIM1->CCR2 <= 6800)
56:     {
57:     TIM1->CCR2 += korak_Motora;
58:     }
```

Sl. 3.28. Primjer pomicanja servo motora u desnu stranu.

Kako bi računalna aplikacija mogla poslati pozicija držača kamere u stupnjevima napisana je funkcija *kutMotora()* koja računa kut na temelju vrijednosti korištene za generiranje PWM signala. Funkcija od vrijednosti postavljene u registru brojača oduzima vrijednost 2199 kako bi se dobila promjena radnog ciklusa PWM signala koja je u rasponu od 2200 – 7000. Zatim se dobivena vrijednost dijeli s vrijednosti 26 koja odgovara pomaku od jednog stupnja te se dobiva pomak horizontalnog servo motora (slika 3.29). Ista je procedura i za vertikalni servo motor. Zatim se dobivena vrijednost u stupnjevima šalje preko UART-a računalnoj aplikaciji.

Linija ***Kôd***

```
193:    kut_X = (TIM1->CCR2 - 2199)/26;
```

Sl. 3.29. Primjer određivanja kuta .

Glavni dio programa se sastoji od poziva inicijalizacijskih funkcija za konfiguriranje potrebnih sučelja USART, DMA i PWM. Nakon što se inicijaliziraju potrebni periferni uređaji mikroupravljač postavlja servo motore nosača za kameru u početnu poziciju koja je definirana u funkciji *inicijalna_pozicija()*. (slika 3.30).

Linija ***Kôd***

```
218:    USART2_init();
219:    MyDMA_Init();
220:    PWM_Init();
224:    inicijalna_pozicija();
```

Sl. 3.30. Inicijalizacija potrebnih funkcija.

Operacija mikroupravljača definirana je unutar beskonačne petlje koja je realizirana koristeći *while* petlju čiji je argument 1 te se izvršava dok god je mikroupravljač aktivan. U beskonačnoj petlji se nalazi *Usart2_DMA_Receive()* funkcija kojoj je predana adresa na kojoj se nalazi međuspremnik i broj znakova koji se prima. Nakon toga slijedi funkcija *pomicanjeMotora()* koja pomiče servo motore te funkcija *pozicija()* koja šalje računalnoj aplikaciji kutove na kojima se nalazi držač kamere u odnosu na vertikalnu i horizontalu os (slika 3.31).

Linija ***Kôd***

```
229:    while(1)
230:    {
231:    Usart2_DMA_Receive(buffer, 1);
232:    pomicanjeMotora();
233:    pozicija();
224:    }
```

Sl. 3.31. Funkcija za primanje podataka preko UART komunikacije.

3.3. Programska podrška računalne aplikacije

Za upravljanje postoljem izrađena je računalna aplikacija pomoću koje korisnik upravlja pomicanjem nosača kamere slanjem naredbi koristeći UART komunikaciju. Za izradu računalne aplikacije odabran je C# programski jezik koji se koristi u sklopu Windows .NET platforme koja omogućava izradu korisničkih aplikacija za Windows operacijski sustav koji je u današnje vrijeme najraširenij. C# programski jezik korišten za definiranje ponašanja aplikacije objektno je orijentirani programski jezik razvijen u tvrtki Microsoft te se bazira na programskim jezicima C, C++ i Java, a u sklopu .NET platforme se koristi za razvijanje vizualnih aplikacija.

Komunikacija računalne aplikacije i mikroupravljača odvija se preko UART komunikacije tako da računalna aplikacija šalje naredbe koje odgovaraju odabranom smjer u kojem se želi pomaknuti držač kamere te prikazuje sliku uživo koja dolazi sa priključene kamere. Prilikom pomicanja držača kamere računalna aplikacija prima informacije o kutovima na kojima se držač nalazi u odnosu na horizontalnu i vertikalnu os te ih ispisuje u grafičkom sučelju.

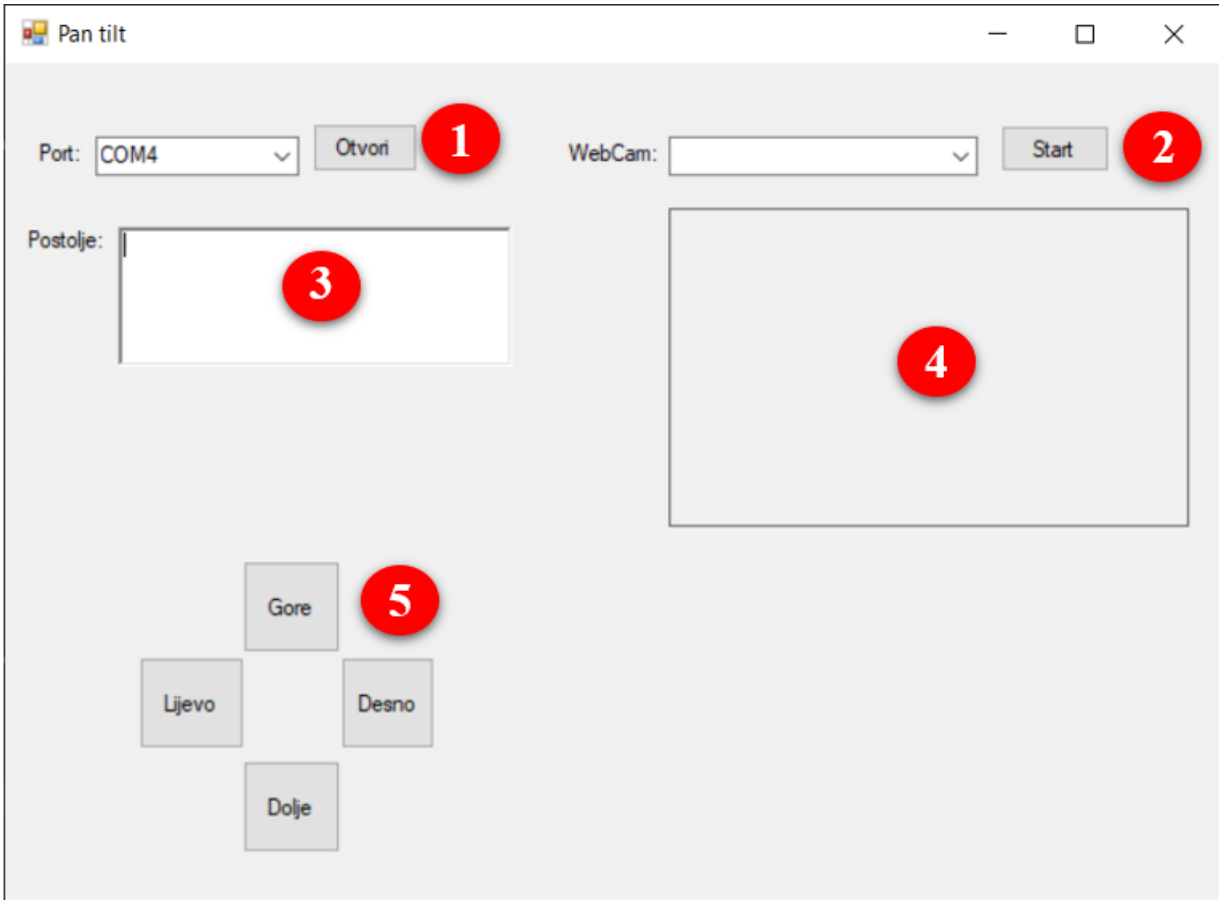
3.3.1. Programski alati

Programski alat korišten za kreiranje aplikacije u C# programskom jeziku je Microsoft Visual Studio 2019 [14] koji omogućava pisanje programskog kôda i stvaranje vizualnog izgleda računalne aplikacije. Riječ je o integriranom razvojnom okruženju koje omogućava razvoj računalnih programa, izradu web stranica i mobilnih aplikacija koristeći razne programske jezike kao što su C, C++, C#, Java Script, HTML i druge.

Unutar Visual Studio razvojnog okruženja koristi se grafičko sučelje koje razvojnim programerima omogućava vizualni prikaz i stvaranje aplikacija za Windows operacijski sustav. Moguće je stvaranje alatnih traka, tipki, padajućih izbornika, tekstualnih okvira i svi ostalih komponenti koje posjeduju klasične Microsoft Windows aplikacije. Zatim se ponašanje pojedinih komponenti definira pisanjem C# programskog kôda.

Grafičko sučelje aplikacije (slika 3.32) se sastoji od padajućeg izbornika (1) u kojem se odabire ulaz računala na koji je postolje priključeno koristeći USB kabel, izbornika za odabir kamere koja je montirana na postolje (2), tekstnog okvira za prikaz poruka koje pristižu od mikroupravljača (3), okvira u kojem je prikazana slika koja dolazi sa web kamere (4) i tipki za odabir smjera pomicanja (5). Nakon što je odabran odgovarajući ulaz aplikacija je spremna za komunikaciju sa mikroupravljačem postolja. Zatim se koristeći tipke (gore, dolje, lijevo, desno) šalju naredbe za

pomicanje nosača web kamere u odabranom smjeru za kut od 1 stupanja. Istovremeno se u aplikaciji može vidjeti slika koja uživo dolazi sa kamere.



Sl. 3.32. Grafičko sučelje računalne aplikacije.

3.3.2. Algoritam i implementacija

Nakon što je stvoreno grafičko sučelje napisan je C# programski kôd koji definira funkcionalnost aplikacije. Prva metoda koja se poziva prilikom pokretanja aplikacije dohvaća popis dostupnih USB ulaza računala. Ovo je potrebno kako bi korisnik iz padajućeg izbornika odabrao ulaz računala preko kojeg se odvija UART komunikacija sa postoljem i ulaz računala na koji je priključena web kamera. Postavljena je brzina prijenosa znakova kod serijske komunikacije sukladno brzini definiranoj u mikroupravljaču (slika 3.33). Zatim se dohvaća popis priključenih kamera te se omogućava korisniku da iz izbornika odabere kameru koja je montirana na držač kamere postolja.

Linija ***Kôd***

```
30:       //uart
31:       string[] ports = SerialPort.GetPortNames();
32:       cboSetComPort.Items.AddRange(ports);
33:       serialPort1.BaudRate = 115200;
34        //web kamera
35:       filterInfoCollection = new
          FilterInfoCollection(FilterCategory.VideoInputDevice);
36:       foreach (FilterInfo filterInfo in filterInfoCollection)
37:       cboSelectCam.Items.Add(filterInfo.Name);
38:       videoCaptureDevice = new VideoCaptureDevice()
```

Sl. 3.33. Inicijalne postavke računalne aplikacije.

Za primanje podataka koji pristižu od mikroupravljača koristi se funkcija *ReadExisting*() koja dohvaća primljene znakove preko UART-a i sprema ih u međuspremnik naziva *dataIn* (slika 3.34) te se njegov sadržaj ispisuje u tekstni okvir grafičkog sučelja računalne aplikacije (slika 3.35).

Linija ***Kôd***

```
48:       dataIn = serialPort1.ReadExisting();
```

Sl. 3.34. Primanje podataka preko UART-a.

```
54:       rtbUartReceive.Text = dataIn;
```

Sl. 3.35. Ispis primljenih podataka u tekstnom okviru računalne aplikacije.

Definirana je funkcija tipki koje se koriste u računalnoj aplikaciji. Tipka „Otvori“ se koristi za postavljanje ulaza računala koji će se koristiti za UART komunikaciju sa postoljem. Metoda otvara ulaz za komunikaciju koji je korisnik odabrao u padajućem izborniku, ukoliko ulaz nije dostupan ili se dogodila greška korisniku se u aplikaciji pojavljuje obavijest sa tekstom greške.

Linija ***Kôd***

```
30:       serialPort1.PortName = cboSetComPort.Text;
31:       serialPort1.Open();
```

Sl. 3.36. Postavke tipke „Otvori“.

Tipka „Start“ pokreće prikazivanje slike uživo koja dolazi sa web kamere koja je odabrana u padajućem izborniku. Za prikaz slike se koristi AForge.NET biblioteka koja je razvijena za potrebe računalnog vida i umjetne inteligencije te je dostupna u sklopu .NET platforme. Slika koja pristiže se prikazuje u okviru za sliku koji se nalazi u grafičkom sučelju aplikacije kako bi korisnik mogao uživo pratiti promjenu kadra kamere kako se pomiče držač kamere na postolju. Definira se da aplikacija uzima sliku s web kamere koja je odabrana u padajućem izborniku. Slika koja dolazi s kamere se prikazuje u okviru za slike pa je potrebno postaviti da se okvir osvježava sa svakom novom sličicom videa koji dolazi s kamere. Zatim se pokreće prikaz slike u okviru računalne aplikacije (slika 3.37).

Linija ***Kôd***

```
115:    videoCaptureDevice = new
        VideoCaptureDevice(filterInfoCollection[cboSelectCam.SelectedIndex].MonikerStrin
116:    videoCaptureDevice.NewFrame += VideoCaptureDevice_NewFrame;
117:    videoCaptureDevice.Start();
```

Sl. 3.37. Postavke tipke „Start“.

Sljedeće se definira funkcija tipki dolje, gore, lijevo, desno na čiji se pritisak mikroupravljaču šalje smjer u kojem da pomakne držač web kamere. Za primjer kad se odabere pomicanje u desnu stranu, preko UART-a se šalje znak „R“ i znak za kraj prijenosa definiran u protokolu (slika 3.38).

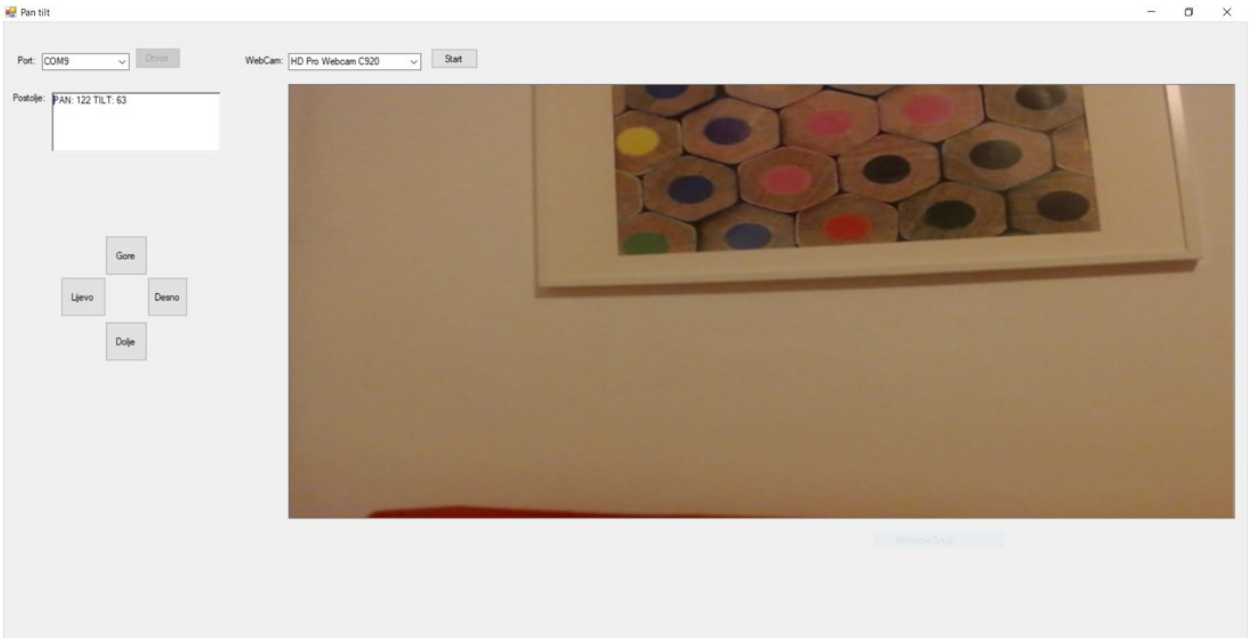
Linija ***Kôd***

```
74:         serialPort1.WriteLine("R");
```

Sl. 3.38. Slanje naredbe za pomicanje u desnu stranu.

3.4. Testiranje

Postolje je povezano s računalom na kojem se nalazi aplikacija za upravljanje te je priključeno na izvor napajanja. Zatim je pokrenuta računalna aplikacija i odabran je USB ulaz računala na koji je postolje spojeno te web kamera koja se koristi. Nakon što je odabrana web kamera u aplikaciji se može vidjeti slika koja pristiže (slika 3.39). Za pomicanje nosača kamere u željeni smjer potrebno je pritisnuti neku od tipki za smjer što rezultira pomicanjem web kamere. Nosač kamere se može pomicati za 180 stupnjeva po horizontalnoj osi (engl. *pan*) i 140 stupnjeva po vertikalnoj osi (engl. *tilt*). Pri pomicanju se mijenjaju vrijednosti kutova na kojima se nalazi držač kamere te ih mikroupravljač šalje računalnoj aplikaciji koja ih ispisuje u grafičkom sučelju.



Sl. 3.39. Računalna aplikacija za upravljanje postoljem.

4. ZAKLJUČAK

Zadatak ovog rada je izrada postolja sa nosačem za web kameru koje se zakreće po horizontalnoj i vertikalnoj osi na zahtjev korisnika. Potrebno je bilo izraditi računalnu aplikaciju za upravljanje i osmisliti način montiranja web kamere. Zatim povezati sve komponente u funkcionalnu cjelinu i testirati gotov sustav.

Odabrane su komponente potrebne za realizaciju sustava te je osmišljen način montiranja web kamere. Odabran je mikroupravljač koji zbog svojih malih dimenzija i zahtjeva za napajanjem te upravljačkih mogućnosti uspješno obavlja potrebne zadatke, a servo motori koji su ugrađeni u nosač kamere zbog svojih karakteristika omogućavaju stabilno pomicanje montirane kamere. Za ispravan rad servo motora odabrano je odgovarajuće napajanje koje svojim dimenzijama može biti ugrađeno u kućište postolja.

Razvijeno rješenje opisano u ovom radu moguće je dodatno unaprijediti dodavanjem mogućnosti bežične komunikacije i ugrađivanjem baterije kako bi uređaj bio prijenosan. Također se nosač kamere može izvesti drugačije tako da se na postolje mogu montirati razne kamere, ali to zahtjeva i upotrebu snažnijih elektromotora u kombinaciji sa zupčastim prijenosom kako bi se povećala stabilnost i zakretni moment nosača kamere. Računalna aplikacija korištena za upravljanje postoljem se može unaprijediti dodavanjem raznih mogućnosti kao što su detekcija objekata na slici, prepoznavanje objekata i praćenje.

Literatura

[1] „Amcrest 4MP ProHD Indoor WiFi Camera“. [online]. Dostupno na: https://www.amazon.com/Amcrest-Security-Two-Way-4-Megapixel-IP4M-1041B/dp/B095XCGKTF [Pristupljeno: 10.9.2021.].

[2] „Mcoplus YT-260 Video Motorized Pan Head“. [online]. Dostupno na: https://www.amazon.com/Mcoplus-Control-Motorized-Extreme-Smartphone/dp/B074WVL4NM [Pristupljeno: 10.9.2021.]

[3] „Bescor Motorized Pan & Tilt Head“. [online]. Dostupno na: https://www.amazon.com/Bescor-Motorized-Power-supply-Extension/dp/B00L88SB5O [Pristupljeno: 12.9.2021.].

[4] S. D., Zimmermann, H. L., Martin, „,Electronic pan/tilt/zoom camera system‘, Proc. SPIE 1763, Airborne Reconnaissance XVI“, str. 1-2, 1993.

[5] „STM32f103c8t6 Datasheet“. [online]. Dostupno na: https://www.st.com/resource/en/datasheet/stm32f103c8.pdf [Pristupljeno: 5.9.2021.].

[6] L., Ryzhyk, „The ARM Architecture“, str. 1-4, 2006.

[7] „Blue Pill“. [online]. Dostupno na: https://www.ebay.com/itm/STM32F103C8T6-ARM-STM32-Minimum-System-Development-Board-Module-For-Arduino/233028113080 [Pristupljeno 10.9.2021.].

[8] „MG996R High Torque Metal Gear Dual Ball Bearing Servo“. [online]. Dostupno na: https://www.electronicoscaldas.com/datasheet/MG996R_Tower-Pro.pdf [Pristupljeno: 10.9.2021.].

[9] D., Nedelkovsk, „How Servo Motor Works & How To Control Servos using Arduino“ [online]. Dostupno na: https://howtomechatronics.com/how-it-works/how-servo-motors-work-how-to-control-servos-using-arduino/ [Pristupljeno 6.9.2021.].

[10] „PL2303TA USB to Serial Bridge Controller“. [online]. Dostupno na: https://cdn-shop.adafruit.com/datasheets/DS_PL2303TA_d20120504.pdf [Pristupljeno: 10.9.2021.].

[11] U., Nanda, S. K., Pattnaik, „Universal Asynchronous Receiver and Transmitter (UART)“, Department of Electronics and Communication Engineering Silicon Institute of Technology Bhubaneswar, India.,str. 1, 2016.

[12] „LM2596 simple switcher“. [online]. Dostupno na: https://www.ti.com/lit/ds/symlink/lm2596.pdf?ts=1631562606765&ref_url=https%253A%252F%252 [Pristupljeno: 2.9.2021.].

[13] „TrueSTUDIO“. [online]. Dostupno na: https://www.st.com/en/development-tools/truestudio.html [Pristupljeno: 10.9.2021.].

[14] „Visual Studio“. [online]. Dostupno na: https://visualstudio.microsoft.com/ [Pristupljeno: 10.9.2021.].

Sažetak

U ovom radu je opisan postupak izrade *pan-tilt* postolja za web kameru. Postupak se sastoji od odabira potrebnih elektroničkih komponenti i pisanja programskog kôda za programiranje mikroupravljača. Detaljno su opisani svi potrebni koraci i razlozi odabira pojedinih komponenti te su navedene njihove karakteristike. Opisani su bitni dijelovi napisanog programskog kôda i način programiranja mikroupravljača kako bi izvršavao potrebne zadatke. Razvijena je računalna aplikacija za upravljanje postoljem koja korisniku omogućava pomicanje nosača kamere te je opisan UART serijski komunikacijski protokol koji se koristi za komunikaciju između aplikacije i postolja. Nosač web kamere pomiču ugrađeni servo motori od kojih se jedan zakreće po horizontalnoj osi, a drugi po vertikalnoj osi. Izrađena je prototipna pločica koja povezuje elektroničke komponente te je smještena u kućište koje također služi i kao postolje za nosač web kamere.

Ključne riječi: mikroupravljač*, Pan-tilt* postolje za web kameru, računalna aplikacija, UART

Abstract

This paper describes the process of making a pan-tilt stand for a webcam. The process consists of selecting the necessary electronic components and writing the program code for programming the microcontroller. All the necessary steps and reasons for selecting individual components are described in detail and their characteristics are listed. The essential parts of the written program code and the way of programming the microcontroller in order to perform the necessary tasks are described. A computer stand management application has been developed that allows the user to move the camera mount, and a UART serial communication protocol is used to communicate between the application and the stand. The webcam bracket is moved by built-in servo motors, one of which rotates on the horizontal axis and the other on the vertical axis. A prototype board was made that connects the electronic components and is housed in a housing that also serves as a stand for the webcam holder.

Key words: desktop aplication ,microcontroller, Pan-tilt camera stand, UART

Prilozi

Na CD-u se nalaze:

- Programski kôd za mikroupravljač
- Programski kôd i grafičko sučelje računalne aplikacije
- Shema sustava
- Priručnici korištenih komponenti