



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

# Upravljanje uređajima govorom



- Δ Svima zainteresiranima
- Δ Osnove programiranja u C-u
- Δ Programiranje STM mikrokontrolera i povezivanje različitih modula
- Δ Shema sustava, upute za povezivanje

## Sažetak

Tehnologija svakim danom sve više i više napreduje. Pa je tako i cjelokupan napredak u području pametnih kuća vidljiv svuda oko nas. No, u današnje vrijeme ima dosta ljudi koji su nepokretni ili su stari i nemoćni. Stoga, bi njima od velike pomoći bilo da određenim stvarima po kući mogu upravljati govorom. Samo područje prepoznavanja govornih naredbi još se dosta razvija. Tako danas postoji veći broj različitih algoritama koji se koriste u tome. U sklopu ovog rada razvijen je jednostavan sustav za upravljanje uređajima pomoću govora. Sustav za upravljanje uređajima govorom razvijen je na *STM32F0 Discovery* porodici mikrokontrolera, a sastoji se od odašiljačkog i prijemnog dijela. Odašiljački dio osim samog mikrokontrolera koristi i *EasyVR 2.0* modul za prepoznavanje govornih naredbi te *nRF24L01+* modul za bežični prijenos podataka. S druge strane, prijemni dio sadrži *nRF24L01+* modul za primanje podataka, *WTV020SD* modul za reprodukciju glazbenih zapisa te jednu LED svjetleću diodu koja simulira kućnu rasvjetu. Također, ukratko su objašnjeni tijekovi izvršavanja programa u odašiljačkom i prijemnom dijelu sustava.

## Sadržaj

1. UVOD.....	3
2. SUSTAV ZA UPRAVLJANJE UREĐAJIMA GOVOROM .....	4
2.1. Odašiljač.....	5
2.1.1. EasyVR 2.0 .....	5
2.1.2. STM32F0 Discovery .....	6
2.1.3. nRF24L01+ .....	7
2.2. Prijemnik .....	7
2.2.1. nRF24L01+ .....	8
2.2.2. STM32F0 Discovery .....	8
2.2.3. WTV020SD .....	8
3. PROGRAMSKA IMPLEMENTACIJA SUSTAVA.....	10
3.1. Odašiljač.....	10
3.2. Prijemnik .....	14
4. ZAKLJUČAK.....	17
5. LITERATURA.....	18
6. POJMOVNIK .....	19

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

Cjelokupan napredak u području pametnih kuća vidljiv je svuda oko nas. Sav razvoj na tom području usko je vezan za trendove koji su zastupljeni u pojedinim društvima. Prvenstveno, razvoj pametnih kuća ponajviše se orijentira na stvaranje kuća koje doprinose zaštiti okoliša te štednji energije. Osim toga, razvoj pametnih kuća ima jako velik potencijal u razvoju pametnih okolina koje bi omogućile kvalitetniji i bolji život starijima, nemoćnima te ljudima sa invaliditetom. Pametna kuća je zapravo skup tehnologija i usluga integriranih u kućnu mrežu, kako bi omogućili ljudima bolju kvalitetu života. Sve u svemu, svakodnevno oko nas izviru nove tehnologije vezane uz pametne kuće, a jedan takav jednostavan sustav razvijen je i opisan u sklopu ovoga seminara. Jedan od zanimljivijih problema u današnje vrijeme je kako možemo na jednostavan u učinkovit način upravljati pojedinim uređajima pomoću govora. Ovo područje razvoja bi uvelike pomoglo starijima i nemoćnima, te invalidima. Njima bi se uvelike olakšalo kada bi postojao nekakav sustav koji bi ih stalno pratio te ostvarivao nekakvu komunikaciju s njima. Primjerice, ako je osoba u bolovima ili se srušila, sustav bi započeo nekakav oblik komunikacije kako bi provjerio je li sve u redu ili je osoba samo slučajno pala. Stoga, kada bi sustav detektirao da se osoba srušila, on bi ju pitao „Jeste li u redu?“. Tada bi recimo očekivao odgovor „Da“ ili „Ne“. Ako bi odgovor bio „Ne“, sustav bi započeo nekakav protokol pružanja pomoći ozlijeđenoj osobi. Ili, kada bi osoba koja je u kolicima htjela jednim potezom ugasiti sva svjetla po kući, jednostavno bi izdala naredbu „Ugasi sva svjetla“ i sustav bi jednostavno pogasio sva svjetla po kući. Poanta svega je da kada bi se napravio ovakav nekakav pametan sustav, osoba bi mogla govorom komunicirati te si uvelike olakšati život.

U sljedećem poglavlju opisan je jedan jednostavan mogući sustav za upravljanje uređajima po kući govorom. Opisuje se implementacija sustava te njegova primjena na jednostavnim sustavima za upravljanje svjetlom i glazbom.

## 2.Sustav za upravljanje uređajima govorom

Sustav za upravljanje uređajima govorom čini dio sustava za upravljanje svim uređajima koji se nalaze u jednoj pametnoj kući. Sustav za upravljanje uređajima govorom sastoji se od dva glavna dijela koja se mogu vidjeti na slici 1.

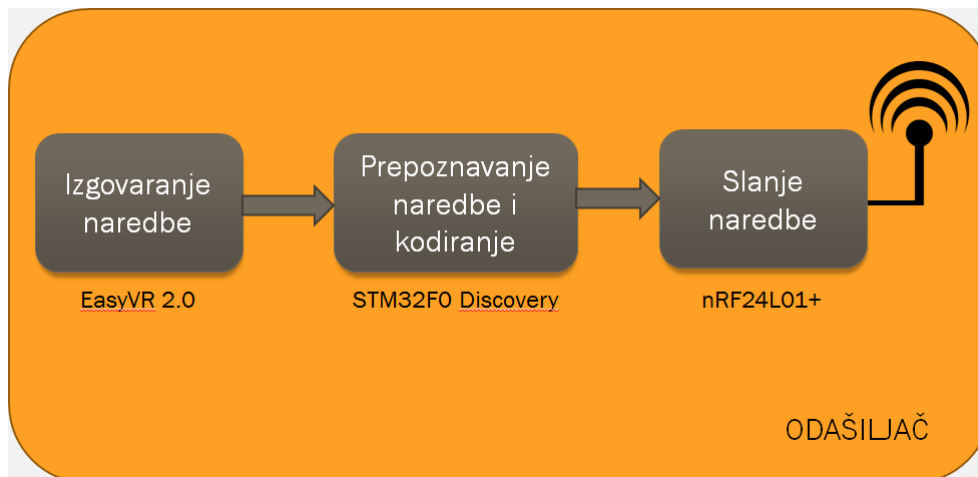


**Slika 1: Prikaz sustava za upravljanje uređajima govorom**

Kao što je na slici prikazano, ovakav sustav sastoji se od dva glavna dijela, a to su odašiljač i prijemnik. Odašiljač je dio koji je mobilan te bi trebao biti što manjih dimenzija, kako bi ga korisnik mogao bez problema nositi sa sobom, a da mu ne predstavlja teret. Također, uređaj bi trebao biti na što manjoj udaljenosti od čovjekovih usta, kako bi što bolje detektirao određene naredbe. S druge strane, prijemnik se tada može nalaziti bio gdje u kući te ako je odašiljač u njegovom dometu on će taj signal primiti bez poteškoća. Time je potrebno implementirati nekoliko čvorova po kući koji će zadanu naredbu uspješno prenijeti do željenog odredišta. Samim time, postićemo bolju pokrivenost prostorija po kući te veći fleksibilnost za sam položaj čovjeka u prostorijama. U sljedećim pododjeljcima ukratko su opisane funkcionalnosti oba dijela ovoga sustava.

## 2.1. Odašiljač

Blokovska shema odašiljača prikazana je na slici 2.

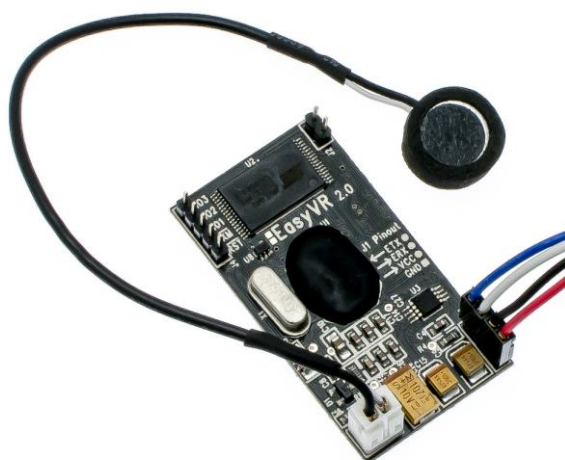


**Slika 2: Prikaz blokovske sheme odašiljača**

Kao što možemo vidjeti, odašiljač se sastoji od *EasyVR 2.0* modula za prepoznavanje određenih naredbi, zatim *STM32F0 Discovery* mikrokontrolera koji predstavlja centralnu jedinicu ovoga podsustava i na kojega se svi ovi moduli spajaju i RF primopredajnika kojime je omogućena bežična komunikacija između odašiljača i prijemnika. Sve ovo je uspješno implementirano, a opisi funkcija koje obavljaju navedeni moduli nalaze se u nastavku.

### 2.1.1. EasyVR 2.0

*EasyVR 2.0* modul je jedna od glavnih komponenti ovoga sustava na kojoj se temelji cijeli ovaj rad, a prikazan je na slici 3.



**Slika 3: Prikaz EasyVR 2.0 modula**

To je višenamjenski modul za prepoznavanje govora napravljen kako bi omogućio dodavanje svestrane sposobnosti prepoznavanja govora na gotovo bilo koje aplikacije. Cjelokupan opis i način rada samog modula opisani su u [1], tako da se čitatelju svakako preporuča proučiti navedenu dokumentaciju, prije nego krene raditi s ovim modulom. Njegova glavna funkcija je prepoznati određenu naredbu, koja je prethodno snimljena na njega, te podatak o naredbi poslati *STM32F0 Discovery* pločici preko serijske veze. Naravno, modul posjeduje mikrofona u kojega se navedene naredbe izgovaraju.

### 2.1.2. STM32F0 Discovery

*STM32F0 Discovery* pločica predstavlja jezgru odašiljača čiji je glavni zadatak da naredbu primljenu sa *EasyVR* modula kodira te šalje na RF primopredajnik, a prikazana je na slici 4.



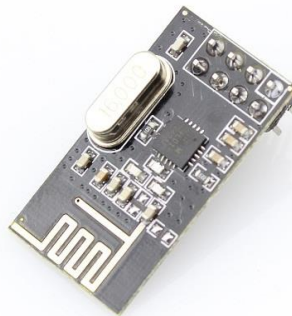
**Slika 4: STM32F0 Discovery razvojna pločica**

Kao što se vidi, na nju se spajaju *EasyVR 2.0* modul te RF primopredajnik. Sama pločica dodatno je spojena na računalo preko

serijskog sučelja kako bi korisnik jednostavnije pratio odvijanje programa unutar podsustava. Navedeno serijsko sučelje će kasnije biti u potpunosti nepotrebno, stoga će se kod konačne izvedbe ono jednostavno maknuti. Dodatne karakteristike i mogućnosti samog mikrokontrolera mogu se vidjeti u [2].

### 2.1.3. nRF24L01+

*nRF24L01+* modul je integrirani primopredajnik koji radi u radio-frekvencijskom području. To je modul niske potrošnje i malog doseg, koji radi u 2.4GHz ISM (*Industrial, Scientific and Medical*) području te podržava brzinu prijenosa do 2Mbps, a prikazan je na slici 5.

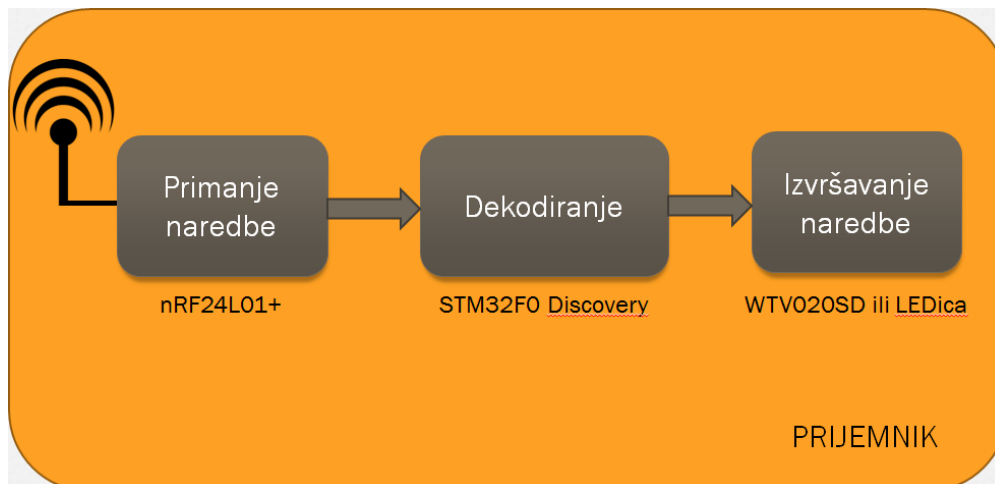


**Slika 5: nRF24L01+ RF primopredajnik**

Modul se spaja na mikrokontroler te s njim komunicira preko SPI sučelja. Njegova prvenstvena uloga je da odašilje kodiranu naredbu koju mu preda mikrokontroler. Dodatne karakteristike ovoga modula mogu se vidjeti u [3].

## 2.2. Prijemnik

Blokovska shema prijemnika prikazana je na slici 6.



**Slika 6: Prikaz blokovske sheme prijemnika**

Kao što se može vidjeti, prijemnik se sastoji od RF primopredajnika koji služi za prijem naredbe poslana od strane odašiljača, zatim imamo *STM32F0 Discovery* mikrokontroler koji čini centar ovoga podsustava i na kojega se spajaju svi ostali moduli i dodaci, te modul za reprodukciju zvučnih zapisa čija je svrha da simulira nekakav glazbeni uređaj koji se inače nalazi u kućama. Osim toga, na mikrokontroler spojena je i jedna LED svjetleća dioda koja simulira kućnu rasvjetu. Navedene komponente i njihova uloga opisani su i nastavku. *STM32F0 Discovery* mikrokontroler te RF primopredajnik opisani su i poglavljima prije, stoga će u nastavku biti opisana samo njihova uloga u ovom podsustavu.

### 2.2.1. nRF24L01+

Ovaj modul [3] opisan je u poglavljima prije gdje je također priložena i njegova slika. Uloga ovom RF primopredajnika u prijemniku je primanje koda naredbe, koji je poslan od strane odašiljača, te njegovo slanje na mikrokontroler. Sa mikrokontrolerom spojen je preko SPI sučelja, pa se sva komunikacija odvija tim putem.

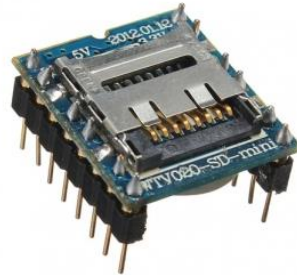
### 2.2.2. STM32F0 Discovery

Na *STM32F0 Discovery* mikrokontroler [2] u ovom podsustavu spojeni su RF primopredajnik, LED svjetleća dioda te modul za reproduciranje zvučnih zapisa. Uloga mikrokontrolera je da naredbu koja mu je pristigla dekodira te da pokrene sklopovlje, ovisno o samoj naredbi.

### 2.2.3. WTV020SD

*WTV020SD* modul je jednostavan integrirani krug za reprodukciju zvučnih zapisa, a prikazan je na slici 7.





**Slika 7: WTV020SD modul za reprodukciju zvučnih zapisa**

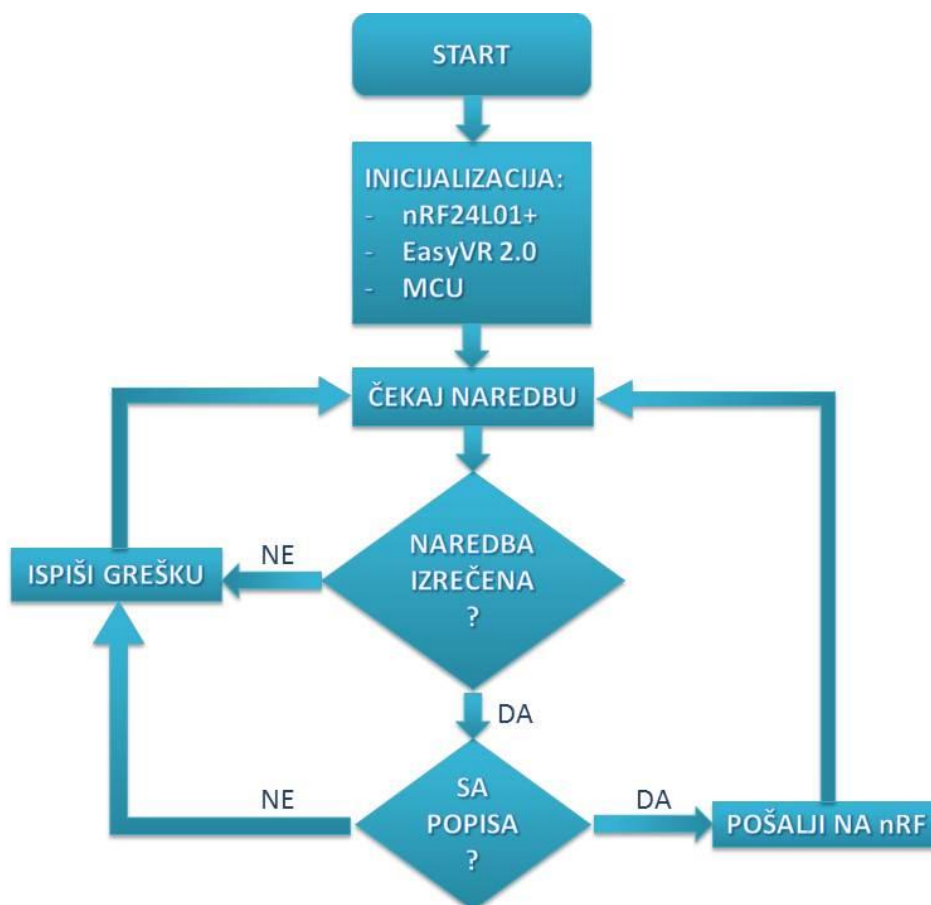
Modulu se na izvode mogu spojiti tipke, te tako može raditi samostalnom načinu rada, neovisno o mikrokontroleru. Tipke koje se mogu spojiti su *play*, *pause*, *stop*, *volume up*, *volume down* te *reset*. U našem slučaju, modul je spojen u serijski način rada gdje se jedan od priključaka koristi kao *clock*, dok se drugi koristi kao *data input* za prijenos podataka. WTV020SD očitava zvučne zapise sa *microSD* kartice, koji su na nju snimljeni u *ADPCM* formatu, i reproducira ih na zvučnik koji je spojen na izvode modula. Detaljnije o samome modulu može se pronaći u [4].

### 3. Programska implementacija sustava

U ovom poglavlju bit će riječi o programskoj implementaciji cijelog sustava za upravljanje uređajima govorom. Ovaj sustav čine komponente navedene u prethodnom poglavlju. U narednim poglavljima bit će govora o samom algoritmu implementiranom sa ovim komponentama.

#### 3.1. Odašiljač

Na slici 8 prikazan je dijagram tijeka programa, odašiljačkog dijela ovoga sustava.



Slika 8: Prikaz dijagrama tijeka odašiljačkog dijela sustava

START započinje uključivanjem mikrokontrolera i dovođenjem napajanja. Zatim slijedi inicijalizacija sučelja mikrokontrolera za komunikaciju s *EasyVR* i *nRF24L01+* modulima, nakon čega se

inicijaliziraju sami moduli. Tada sustav prelazi u stanje čekanja gdje čeka da se naredba izgovori, te je taj kodni odsječak prikazan u nastavku.

```
initBlueLED();
spi_nrfInit();

uartEVROpen(USART1, 9600); //PA9 i PA10
uartPCOpen(USART2, 9600); //PA2 i PA3

if(!EVRdetect()) {

    uartPutString("EasyVR not detected! ", USART2);
    while(1); //Error loop
}

EVRsetPinOutput(IO1, RESET);
uartPutString("EasyVR detected! ", USART2);
EVRsetTimeout(5);
EVRsetLanguage(ENGLISH);

group = TRIGGER;

while(1) {

    EVRsetPinOutput(IO1, SET); //LED on (listening)
    uartPutString("Say command in Group ", USART2);
    uartSendInt(&group, USART2);
    EVRrecognizeCommand(group);

    do {
        //tu se moze nesto raditi dok se ceka naredba
        delay_ms(3000);
    } while (!EVRhasFinished());

    EVRsetPinOutput(IO1, RESET); //LED off

    idx = EVRgetWord();
    if (idx >=0) {

        group = GROUP_1;
    }
    idx = EVRgetCommand();
    if (idx >= 0) {
        uint8_t train = 0;
        char name[32];
        uartPutString("Command: ", USART2);
        uartSendInt(&idx, USART2);
        if(EVRdumpCommand(group, idx, &name[0], &train)) {
```

```
        uartPutString(" = ", USART2);
        uartPutString(name, USART2);
    }
    else
        uartPutString("nesto ", USART2);

    EVRplaySound(BEEP, VOL_FULL);
    //some action
    EVRaction();
}
else { //errors or timeout
    if(EVRisTimeout()){
        uartPutString("Timed out, try again...", USART2);
    }
    err = EVRgetError();
    if (err >= 0) {
        uartPutString("Error ", USART2);
        //uartPutString(err, USART2);
    }
}
}
}

void EVRaction (void) {
    switch (group) {

        case GROUP_0:
            switch (idx)
            {
                case G0_POKRENI:
                    group = GROUP_1;
                    break;
            }
            break;
        case GROUP_1:
            switch (idx)
            {
                case G1_GLAZBA:
                    group = GROUP_2;
                    sendAction = 49;
                    break;
                case G1_SVJETLO:
                    group = GROUP_2;
                    sendAction = 51;
                    break;
            }
            break;
    }
}
```

```

    case GROUP_2:
        switch(idx) {
            case G2_UPALI:
                group = GROUP_1;
                //Inicijaliziraj nRF
                nRF24L01_initializeDebug(0, 32, 0);
// (Tx mode, payload = 32 bytes, autoAck = disable)
                nRF24L01_clearFlush();
                //sprintf(txBuff, "%c", &idx);
                sendAction += 4;

                //posalji paket
                nRF24L01_writeTxPayload(&sendAction, 32, 1);

                //posalji na UART da je poslano
                uartPutString("Poruka poslana ", USART2);
                GPIO_SetBits(GPIOC, GPIO_Pin_8);
                //pricekaj sekundu
                delay_ms(1000);

            break;

            case G2_UGASI:
                group = GROUP_1;
                //Inicijaliziraj nRF
                nRF24L01_initializeDebug(0, 32, 0);
// (Tx mode, payload = 32 bytes, autoAck = disable)
                nRF24L01_clearFlush();
                //sprintf(txBuff, "%c", &idx);
                sendAction += 5;

                //posalji paket
                nRF24L01_writeTxPayload(&sendAction, 32, 1);

                //posalji na UART da je poslano
                uartPutString("Poruka poslana ", USART2);
                //pricekaj sekundu
                delay_ms(1000);
                GPIO_ResetBits(GPIOC, GPIO_Pin_8);

            break;
        }
    break;
}
}

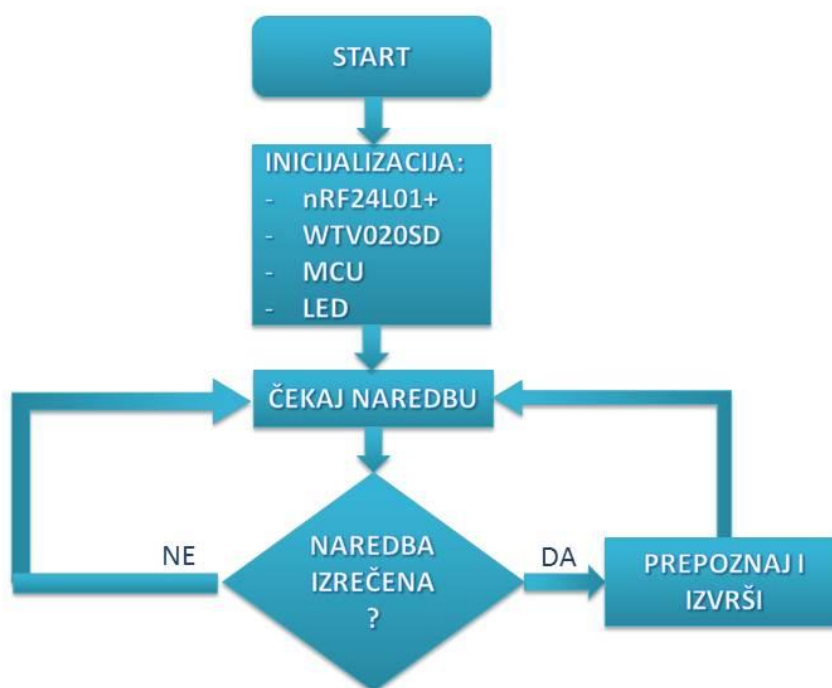
```

*EasyVR* modul se postavi da čeka nekoliko sekundi gdje čeka izgovor naredbe. Kada to vrijeme istekne, *EasyVR* šalje poruku preko serijskog sučelja da je vrijeme isteklo. Kada se naredba izgovori, *EasyVR* nastoji prepoznati naredbu. Svaka naredba dobije svoj jedinstveni kod, koji se

pošalje preko *nRF24L01+* modula, ako je prepoznavanje uspješno. Prepoznavanje se odvija u beskonačnoj petlji.

### 3.2. Prijemnik

Na slici 9 prikazan je dijagram tijeka programa prijemnog dijela ovoga sustava.



**Slika 9: Prikaz dijagrama tijeka programa prijemnog dijela sustava**

START započinjem uključivanjem mikrokontrolera i dovođenjem napajanja. Zatim slijedi inicijalizacija sučelja mikrokontrolera za komunikaciju sa *WTV020SD* modulom, *nRF24L01+* modulom te za upravljanje LED svjetlećom diodom. Tada sustav prelazi u stanje čekanja gdje čeka naredbu sa *nRF24L01+* modula te je kodni odsječak prikazan u nastavku.

```
//Crvena ledica  
RCC_AHBPeriphClockCmd(RCC_AHBPeriph_GPIOC, ENABLE);  
GPIO_StructInit(&zeleni);  
zeleni.GPIO_Pin = GPIO_Pin_7;  
zeleni.GPIO_Mode = GPIO_Mode_OUT;
```

```
zelena.GPIO_Speed = GPIO_Speed_50MHz;
zelena.GPIO_OType = GPIO_OType_PP;
GPIO_Init(GPIOC, &zelena);

//Inicijalizacija USART1
uart_open(USART1, 9600);
uart_put_string("Inicijalizacija je uspjela\n", USART1);

//Inicijalizacija SPI1 za nRF24L01
spi_nrfInit();

WTV20SD_init();
WTV20SD_reset();
WTV20SD_unmute();
WTV20SD_volume(4);
delay_10us(20000);

testUnidirectionalMessagingRx();

while(1) {
    }
}
```

Kada naredba stigne, prepozna se kod naredbe te se ona izvrši. Nakon toga, sustav ponovno prelazi u stanje čekanja. Programski kod izvršavanja pristigle naredbe prikazan je u nastavku.

```
void testUnidirectionalMessagingRx(void) {

    unsigned char rxBuff[buffLen];
    //char temp[buffLen];
    char temp=0;

    //inicijaliziraj nRF
nRF24L01_initializeDebug(1, 32, 0); // (Rx mode, payload = 32, autoAck = disable)
nRF24L01_clearFlush();

    //napisi na UART da je u stanju cekanja
    uart_put_string("Cekanje poruke... ", USART1);

    while (1)
    {
        //pogledaj jel ima sto u Rx bufferu
        if(nRF24L01_irqRxDrActive()) {

            //ako ima izvuci paket
            nRF24L01_readRxPayload(rxBuff, 32);
            //ocisti interrupt u nRFu
        }
    }
}
```

```
nRF24L01_irqClearAll();

//napisi paket na seriju
//strncpy(&temp, rxBuff[0], 32);
temp = rxBuff[0];
uart_put_string(&temp, USART1);

if(rxBuff[0] == 53) {
    WTV20SD_playVoice(0);
    uart_put_string("Glazba upali! ", USART1);
}
else if(rxBuff[0] == 54) {
    WTV20SD_stopVoice();
    uart_put_string("Glazba ugasi !", USART1);
}
else if(rxBuff[0] == 55) {
    GPIO_SetBits(GPIOC, GPIO_Pin_7);
    uart_put_string("Svjetlo upali! ", USART1);
}
else if(rxBuff[0] == 56) {
    GPIO_ResetBits(GPIOC, GPIO_Pin_7);
    uart_put_string("Svjetlo ugasi! ", USART1);
}
}
}
```



## 4. Zaključak

Ovakav sustav je jedna vrlo praktična, jednostavna i jeftina stvar za izvesti. Ovakvi sustavi bi posebice mogli koristiti kod ljudi koji se ne mogu kretati te kod starijih i nemoćnih. Naravno, primjena ovog sustava nije ograničena te se može koristiti u bilo kojem kućanstvu te za bilo koju aplikaciju. U budućnosti, ovaj sustav bi se mogao bolje razviti tako da se implementiraju određeni algoritmi za prepoznavanje govora na nekakvim DSP procesorima. Osim toga, pokazalo se da se uređaju smanjuje učinkovitost, kada u pozadini svira određena glazba ili se čuje razgovor ljudi. Stoga je u budućnosti potrebno implementirati povratnu vezu koja će kompenzirati buku okoline te na taj način uvelike povećati učinkovitost prepoznavanja govornih naredbi. Uz to, moguće je implementirati ovakav sustav na sve kućanske aparate, radi jednostavnijeg korištenja.

## 5. Literatura

- [1] EasyVR User Manual, 2014.  
URL: [http://download.tigal.com/veear/EasyVR\\_2/EasyVR\\_User\\_Manual\\_3.6.6.pdf](http://download.tigal.com/veear/EasyVR_2/EasyVR_User_Manual_3.6.6.pdf) (01.04.2014.)
- [2] STM32F0 Discovery User Manual, 2012.  
URL: [http://www.st.com/st-web-ui/static/active/en/resource/technical/document/user\\_manual/DM00050135.pdf](http://www.st.com/st-web-ui/static/active/en/resource/technical/document/user_manual/DM00050135.pdf) (17. 05. 2012.)
- [3] nRF24L01+ Product Specification, 2008.  
URL: [https://www.sparkfun.com/datasheets/Components/SMD/nRF24L01Plus\\_Preliminary\\_Product\\_Specification\\_v1\\_0.pdf](https://www.sparkfun.com/datasheets/Components/SMD/nRF24L01Plus_Preliminary_Product_Specification_v1_0.pdf) (13. 03. 2008.)
- [4] WTV020-SD Module User Manual, 2008.  
URL: [http://letsmakerobots.com/files/WTV020\\_manual\\_V1.3.pdf](http://letsmakerobots.com/files/WTV020_manual_V1.3.pdf) (19.11.2008.)

## 6. Pojmovnik

Pojam	Kratko objašnjenje	Više informacija potražite na
EasyVR 2.0	Modul za prepoznavanje govornih naredbi	<a href="http://www.veear.eu/products/easyvr/">http://www.veear.eu/products/easyvr/</a>
STM32F0 Discovery	STM razvojna pločica	<a href="http://www.st.com/web/en/home.html">http://www.st.com/web/en/home.html</a>
nRF24L01+	RF primopredajnik	<a href="https://www.nordicsemi.com/eng/Products/2.4GHz-RF/nRF24L01P">https://www.nordicsemi.com/eng/Products/2.4GHz-RF/nRF24L01P</a>
WTV020SD	Modul za reprodukciju zvučnih zapisa	<a href="https://www.sparkfun.com/products/11125">https://www.sparkfun.com/products/11125</a>
RF prijenos	Prijenos radio frekvencijskom području	<a href="http://en.wikipedia.org/wiki/Radio_frequency">http://en.wikipedia.org/wiki/Radio_frequency</a>