

UNIVERSITETI I PRISHTINËS “HASAN PRISHTINA”
FAKULTETI I INXHINIERISË ELEKTRIKE DHE KOMPJUTERIKE
AUTOMATIKË E KOMPJUTERIZUAR DHE ROBOTIKË



PUNIM DIPLOME (Bsc.)

**Realizimi i protokollit USS për komunikim me njësité ngasëse të
Siemens**

Mentori:

Prof. Ass. Dr. Lavdim Kurtaj

Kandidati:

Edison Qerimi

Prishtinë, Korrik 2021

Deklarata e origjinalitetit / Autorësisë

Ky Punim Diplome i nivelit bachelor është puna ime origjinale duke respektuar autorësinë e çdo burimi të informacioneve dhe rregullave për një punim të mirëfilltë shkencor, dhe nuk është dorëzuar, në tërësi apo pjesërisht, për ndonjë gradë në këtë apo ndonjë universitet tjetër. Sipas njohurisë time, punimi nuk përmban asnjë material të botuar ose shkruar nga ndonjë person tjetër, përveç siç deklarohet në brendi të tekstit. Gjithashtu, deklaroj se në shkrimin e punimit kam respektuar rregullat etike të punës shkencore dhe akademike të Universitetit të Prishtinës “Hasan Prishtina”.

Statement of originality / Authority

This Bachelor Thesis is my original work respecting the authorship of any source of information and rules for a genuine scientific thesis, and has not been submitted, in whole or in part, for any degree at this or any other university. To the best of my knowledge, the paper does not contain any material published or written by any other person, except as stated in the text. Also, I declare that in writing the paper I have respected the ethical rules of scientific and academic work of the University of Prishtina “Hasan Prishtina”.

Edison Qerimi

(nënshkrimi – signature)

Përmbajtja

1. Protokollit USS	6
1.1. Hyrje.....	6
1.2. Transferimi i telegramit.....	6
1.3. Struktura e telegramit	7
1.4. Struktura e net bllokut të të dhënave	8
1.4.1. Zona PKW	8
1.4.2. Zona PZD.....	9
1.5. Teknika e transferimit të të dhënave.....	10
2. Pjesët përbërëse për realizimin e protokollit për komunikim	12
2.1. Njësia kontrolluese	12
2.2. Njësia për komunikim.....	13
2.3. Njësia ngasëse	15
2.4. Ekran LCD 2x16 dhe butonat.....	16
3. Softueri i projektit.....	18
3.1. Transmetimi i të dhënave	18
3.2. Pranimi dhe analizimi i të dhënave	20
3.3. Kontrollimi dhe monitorimi i inverterit (PZD)	21
3.4. Leximi dhe shënimi i vlerave të parametrave (PKW)	26
4. Konkluzioni	36
5. Literatura.....	37

Falënderime

Falënderoj familjen time për përkrahjen që më ka dhënë gjatë gjithë kohës së studimeve.

Një falënderim i veçantë i takon mentorit tim Prof. Ass. Dr. Lavdim Kurtaj për mbështetjen dhe ndihmën e tij të pakursyer, për korrektësinë dhe këshillat e dhëna në realizimin e këtij punimi por edhe për mbështetjen e vazhdueshme gjatë periudhës së studimeve.

Falënderoj miqtë e mi për mbështetjen që më kanë dhënë në çdo çast.

Faleminderit të gjithëve!

Abstrakt

Motorët elektrikë janë konsumuesit më të mëdhenj të energjisë elektrike në botë. Sistemet mund të bëhen më eficientë duke përdorur ngasësit me frekuencë te ndryshueshme (VFD). VFD-të gjejnë aplikim në industri nga pajisjet e vogla deri te ato më të mëdha, si në sistemet me pompa dhe të ventilimit. VFD-të përmes rrjetës mund të kontrollohen duke përdorur protokolle të ndryshme si USS, PROFIBUS DP, Modbus, CANopen, EtherNet/IP.

Tema kryesore e këtij punimi është realizimi i protokollit USS për komunikim me njësitë ngasëse të Siemens duke përdorur mikrokontrollerin AT89S8253 si njësi të kontrollës përmes së cilit bëhet kontrollimi i frekuencës së inverterit dhe ndryshimi i parametrave të njësive ngasëse të Siemens. Interfejsi fizik për komunikim ndërmjet njësive është RS-485. Gjuha programuese për realizimin e protokollit është C.

Abstract

Electric motors are the largest consumers of electricity in the world. Systems can be made more efficient by using variable frequency drive (VFD). VFDs find application in industry from small appliances to large ones, such as pumps and fan systems. VFDs through the network can be controlled using various protocols such as USS, PROFIBUS DP, Modbus, CANopen, EtherNet/IP.

The main topic of this thesis is the realization of the USS protocol for communication with Siemens driving units using the AT89S8253 microcontroller as a control unit through which the frequency control of inverter is done and the parameters of the Siemens driving units are changed. The physical interface for communication between units is RS-485. The programming language for the implementation of the protocol is C.

1. Protokollit USS

1.1. Hyrje

Protokollit USS përcakton një procedurë për qasje sipas parimit Master/Slave për komunikim përmes bus-it serik, si në figurën 1.1. Në bus mund të lidhën një master dhe maksimumi 31 slave. Zgjedhja e slave-it në telegram bëhet individualisht nga masteri përmes një karakteri. Slave-i nuk mund të kthej përgjigje nëse nuk bëhet kërkesa në master. Komunikimi realizohet në modin half-duplex. Transferimi i të dhënave kryhet nga karaktere individuale në formatin: 1 start bit, 8 bita për të dhëna, 1 bit pariteti çift dhe 1 stop bit.

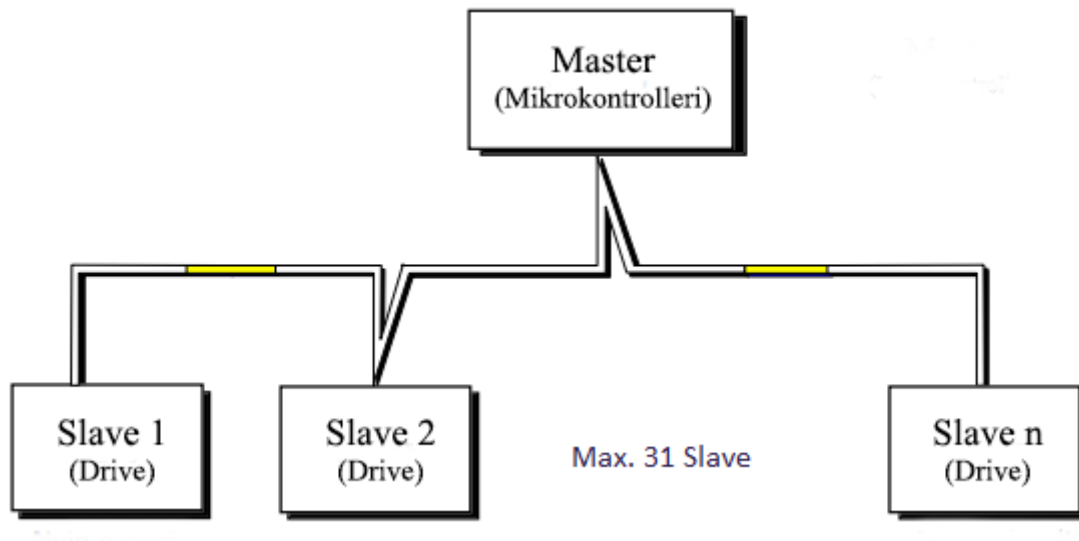


Figura 1.1. Lidhja serike master-slave

1.2. Transferimi i telegramit

Transferimi i telegramit mund të bëhet në formën ciklike dhe aciklike. Në teknologjinë e ngasjes përdoret vetëm forma ciklike. Te forma ciklike masteri vazhdimisht transmeton telegrame në slave dhe pret për telegramin e përgjigjes nga slave i adresuar.

Slave duhet të dërgoj telegramin e përgjigjes në dy raste, nëse:

- pranon telegram pa gabime dhe
- është i adresuar në këtë telegram.

Tabela 1. Llojet e telegramit.

Biti 5	Biti 6	Biti 7	Domethënia
0	0	0	Transferimi standard i të dhënave.
0	1	0	Telegram i pasqyruar: telegrami rikthehet nga slave në master i pandryshuar.
0	0	1	Broadcast: i njëjti telegram dërgohet në të gjitha nyjet.
1	X	X	Telegram special: telegrami refuzohet nga të gjithë slave, kur nuk është definuar telegrami special.

1.4. Struktura e net bllokut të të dhënave

Net blloku i të dhënave ndahet në dy zona, në:

- **Zonën PKW (vlera e ID-së së parametrit)**
- **Zonën PZD (proces i të dhënave)**

Net blloku i të dhënave me gjatësi variabile është paraqitur më poshtë në figurën 1.3.

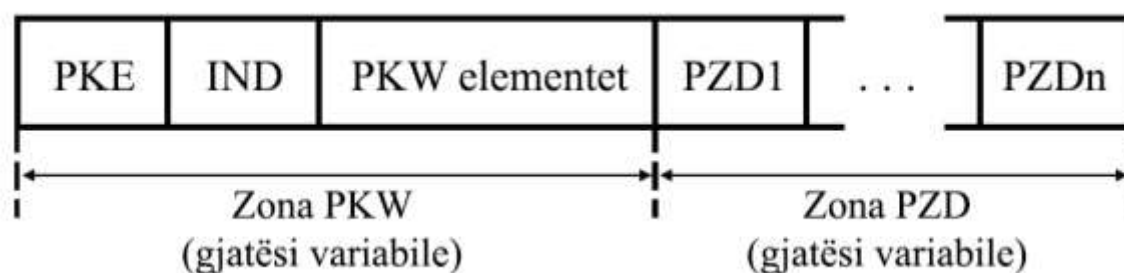


Figura 1.3. Net blloku i të dhënave me gjatësi variabile

1.4.1. Zona PKW

Zona PKW definon një mekanizëm i cili trajton transferimin e të dhënave ndërmjet dy pajisjeve komunikuese. Kjo nënkupton, leximin dhe shënimin e vlerave të parametrave. Në këtë zonë

përmes “detyrave” dhe “përgjigjeve”, mund të bëhet qasja në parametrat e njësive ngasëse. PKW ndahet në tri pjesë:

- **PKE** – Parameter ID, përdoret për të identifikuar dhe iniciuar detyrat dhe përgjigjet për përpunimin e parametrave dhe gjithmonë ka gjatësinë e një fjale (16 bit). Po ashtu në PKE vendoset edhe numri i parametrave.
- **IND** – Indeksi ka gjatësi një fjale.
- **PWE element** – Përmban informatën, të shoqëruar me një detyrë ose përgjigje si vlera, tekste ose përshkrime të parametrave.

Gjatësia e PKW mund të jetë fikse (3 ose 4 fjale) dhe variabile. Në figurën 1.4. është paraqitur struktura e PKW fikse me gjatësi 4 fjale i cili ka vlerën e parametrave 16 bit.

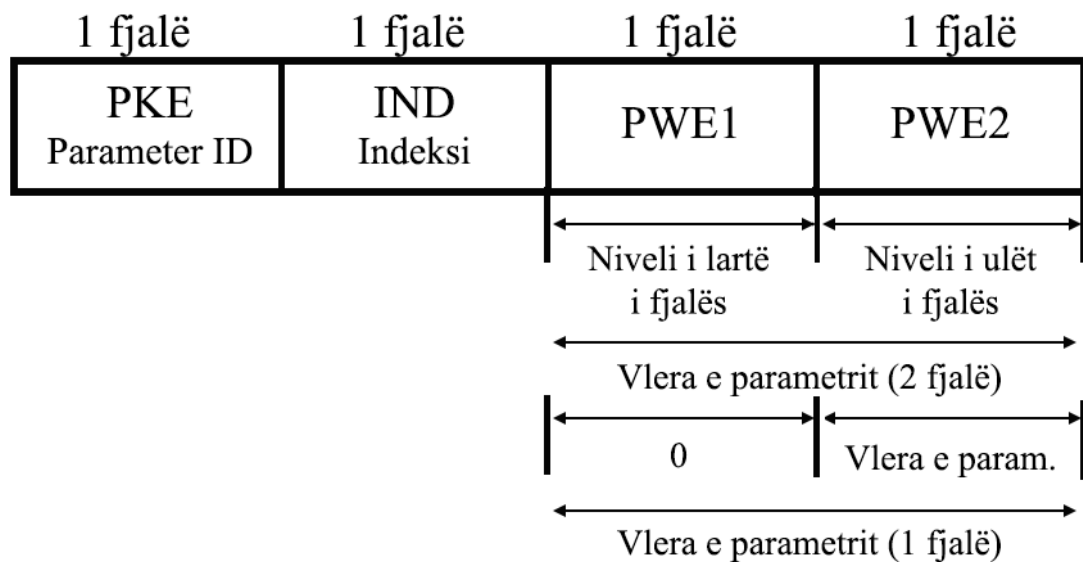


Figura 1.4. PKW fikse me gjatësi 4 fjale.

PKW variabile mund të përmbajë prej 1 në 108 fjale, në qoftë se PZD 16 fjalësh (maksimumi) është e pranishme në bllokun e të dhënave, ose prej 1 në 124 fjale në qoftë se nuk ka PZD.

1.4.2. Zona PZD

Zona PZD e telegramit përdoret për kontrollimin dhe monitorimin e inverterit. PZD-ja e pranuar procesohet gjithmonë me përparësi të lartë në master dhe në slave. Përpunimi i PZD-së ka përparësi ndaj përpunimit të PKW-së dhe gjithmonë transferon të dhënat më aktuale në dispozicion në interfejs.

Zona PZD përmban sinjalet e kërkuara për automatizim:

- Fjalët e kontrollit dhe vlerat e caktuara (setpoint) prej masteri në slave
- Fjalët e statusit dhe vlerën aktuale prej slave në master.

Procesi i të dhënave në këtë zonë vazhdimisht transferohet ndërmjet master-it dhe slave-it.

Zona PZD është strukturuar si më poshtë në tabelën 2.

Tabela 2. Zona PZD.

	PZD1	PZD2	PZD3	PZD4
Telegrami detyrë (master-slave)	Fjala e kontrollit STW	Setpointi kryesor HSW	Setpointi dytësor HSW2	Fjala dytësore e kontrollit STW2
Telegrami përgjigje (slave-master)	Fjala e statusit ZSW	Vlera aktuale kryesore HIW	Fjala e statusit dytësore ZSW2	Vlera aktuale dytësore HIW2

1.5. Teknika e transferimit të të dhënave

Karakteret transferohen me gjatësi 11 bit. Bitin për fillim, 8 bit për të dhëna, bitin e paritetit (çift) dhe bitin për ndalim.

- **Vonesa kohore e fillimit** – Karakteri i fillimit STX ka vlerën gjithmonë 02 hex, gjatë dërgimit të të dhënave, në frame mund të këtë edhe 02 hex tjetër. Prandaj para STX është e nevojshme një vonesë kohore prej se paku 2 kohë të ekzekutimit të karakterit si në figurën 1.6.
- **Vonesa kohore e përgjigjes** – Koha ndërmjet karakterit të fundit të transmetimit të telegramit (BCC) dhe karakterit të fillimit të pranimit të telegramit (STX) njihet si vonesa kohore e përgjigjes si në figurën 1.6. Vlera maksimale e vonesës kohore të përgjigjes duhet të jetë më e vogël se 20ms dhe nuk mund të jetë më e vogël se vonesa kohore e fillimit.

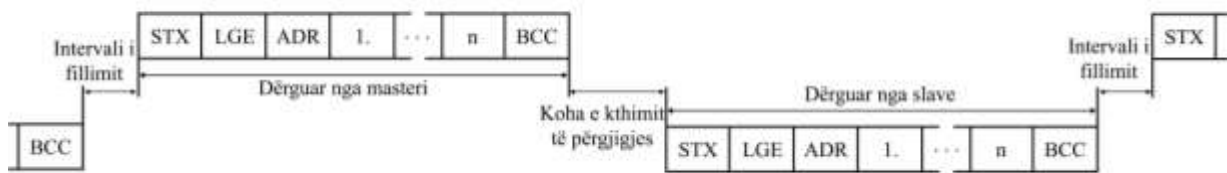


Figura 1.5. Procedura e transferimit të të dhënave

- **Koha e ekzekutimit të karakterit** – Koha e ekzekutimit të karakterit është koha e nevojshme për të dërguar një karakter. Kjo kohë është funksion i baud rate-it dhe llogaritet me formulën:

$$\text{koha e ekzekutimit të karakterit} = \frac{11 * 1000}{\text{baud rate}} [\text{ms}]$$

- **Koha e mbetur e ekzekutimit të telegramit** – Monitorimi i kohës së mbetur të ekzekutimit të telegramit varet nga gjatësia e telegramit.
 - *Telegrami me gjatësi variabile* – Monitorimi i kohës maksimale të mbetur të ekzekutimit të telegramit së pari ngarkohet pasi të pranohet STX, me vlerën e cila merret për një telegram me 252 net karaktere. Pasi të jetë pranuar LGE, koha e monitorimit ngarkohet me vlerën e saktë.
 - *Telegrami me gjatësi fikse* – Monitorimi i kohës maksimale të mbetur të ekzekutimit të telegramit mund të fillojë drejtpërdrejt me vlerën e saktë (nuk merret parasysh vlera e LGE).
- **Koha e mbetur e ekzekutimit të telegramit e kompresuar** – Definohet si koha e ekzekutimit e cila nevojitet për transmetimin e LGE, ADR, net karaktereve në mënyrë të njëpasnjëshme në bllok. Llogaritet me formulën:

$$(\text{numri i net karaktereve} + 3) * \text{koha e ekzekutimit të karakterit}$$

- **Koha maksimale e mbetur e ekzekutimit të telegramit** – Shuma e vonesave kohore të karakterit është e barabartë me 50% të kohës së mbetur të ekzekutimit të telegramit të kompresuar. Koha maksimale e mbetur e ekzekutimit të telegramit llogaritet me formulën:

$$1.5 * (\text{numri i net karaktereve} + 3) * \text{koha e ekzekutimit të karakterit}$$

2. Pjesët përbërëse për realizimin e protokollit për komunikim

Komunikimi me njësitë ngasëse të Siemens përbëhet nga këto pjesë:

- *Njësia kontrolluese për gjenerimin e kërkesës – masteri (mikrokontrolleri)*
- *Njësia për komunikim*
- *Njësia ngasëse – slave*

2.1. Njësia kontrolluese

Për realizimin e njësisë kontrolluese është përdorur mikrokontrolleri AT89S8253 i prodhuar nga Atmel. Është mikrokontroller 8-bitësh me performancë të lartë me 12KB memorie programuese, 256B RAM si memorie e përkohshme. AT89S8253 është mikrokontroller i fuqishëm i cili ofron fleksibilitet të lartë dhe me çmim të arsyeshëm për aplikime të shumta në sistemet e kontrollit.

Si burim të frekuencës se klokut në rastin tonë kemi përdorur kristalin e kuarcit me frekuencë 8MHz. Ky mikrokontroller ka 4 porte hyrëse/dalëse, 32 pina hyrës/dalës, 3 kohorë/numërues, 9 burime të interruptit, një portin serik UART për komunikim. Për realizimin e detyrës do të përdoren të tre kohorët dhe 3 interrupte. Në këtë njësi është realizuar programi i cili përmes portit serik kryen komunikimin me njësinë ngasëse të Siemens. Nga mikrokontrolleri (masteri) gjenerohet kërkesa dhe njësia ngasëse (slave) e kthen përgjigjen. Në fund në mikrokontroller përgjigjja e pranuar analizohet dhe të dhënat e pranuar nga njësia ngasëse paraqiten në ekranin LCD. Në figurën 2.1. është paraqitur pllaka zhvillimore Easy8051A.

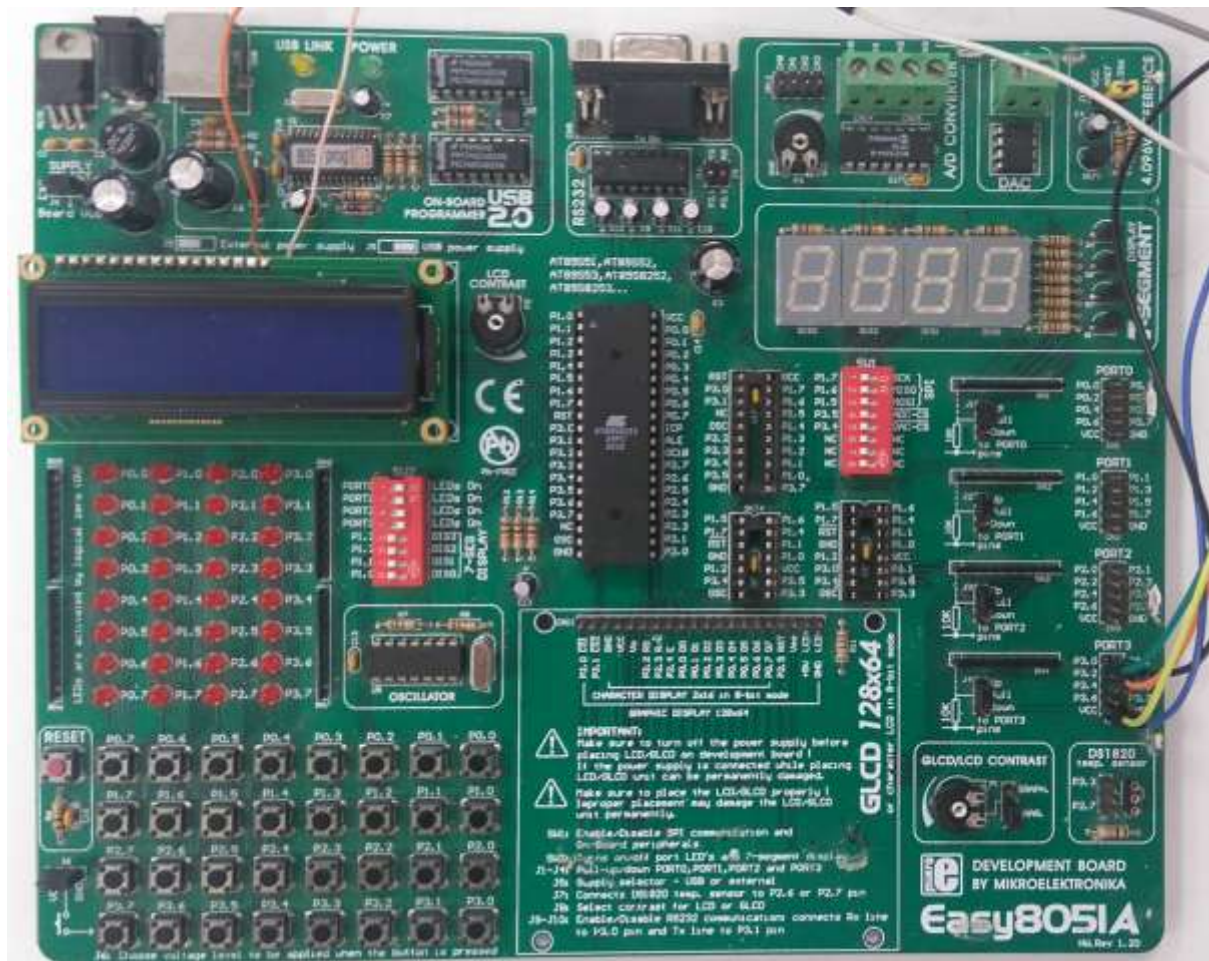


Figura 2.1. Pllaka zhvillimore Easy8051A

2.2. Njësia për komunikim

Të gjithë ngasësit nga Siemens përdorin një interfejs serik si standard. Interfejsi serik përdorë RS-485 me dy tela për lidhje. Distanca ndërmjet mikrokontrollerit dhe njësisë ngasëse mund të jetë deri në 1200m, prandaj gjen aplikim në industri. Deri në 31 njësi ngasëse mund të lidhen në një lidhje të vetme të RS-485, dhe njësitet mund të adresohen individualisht ose të gjithë përnjëherë në modin broadcast. Protokolli USS është zhvilluar nga Siemens dhe është i përbashkët për të gjitha produktet ngasëse të Siemens. Funkcionon deri në 19kBaud, në rastin tonë kemi përdorur 2400Baud. Skema e lidhjes është paraqitur në figurën 2.2a kurse lidhja fizike është paraqitur në figurën 2.2c. Konvertimi i sinjalit është bërë përmes modulit TTL në RS-485, *ST485CN*, e paraqitur në figurën 2.2b.

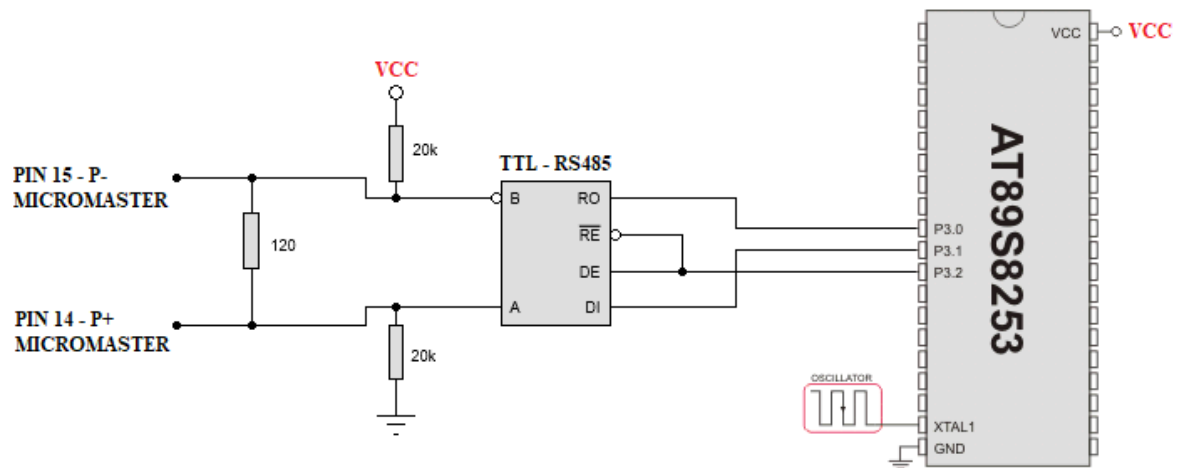


Figura 2.2a. Skema e lidhjes ndërmjet mikrokontrollerit dhe njësisë ngasëse



Figura 2.2b. ST485CN

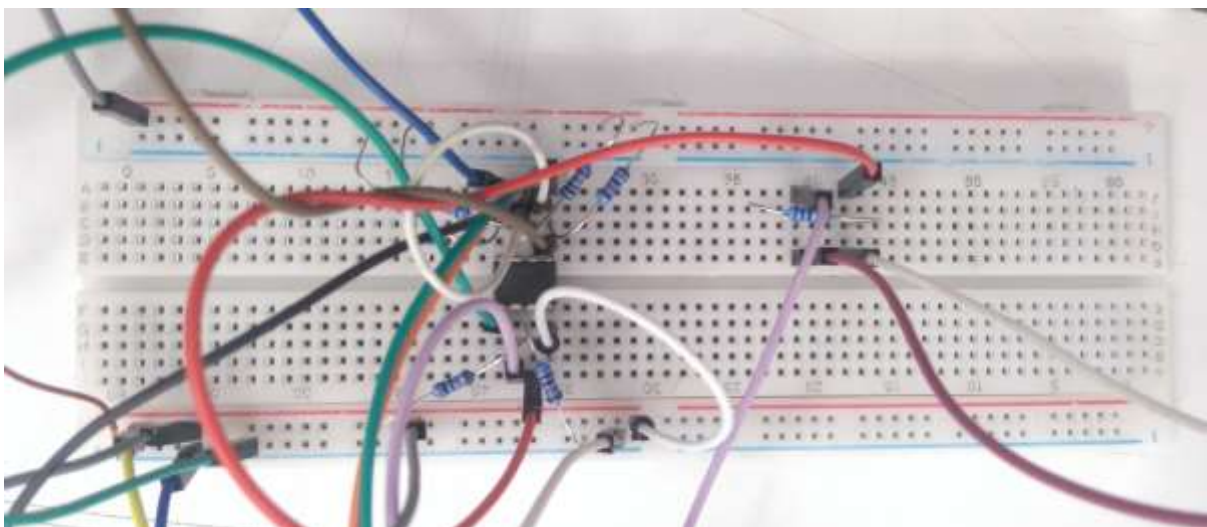


Figura 2.2c. Lidhja fizike e modulit ST485CN

2.3. Njësia ngasëse

Ngasësi me frekuencë të ndryshueshme (VFD) është një pajisje që kontrollon shpejtësinë e motorit duke kontrolluar tensionin dhe frekuencën furnizuese të tij. Për të kontrolluar shpejtësinë e një motori induktiv AC kërkohej një kontrollues më kompleks, i quajtur inverter. Inverteri është një konvertues elektronik i cili konverton rrymën e vazhduar (DC) në rrymë alternative (AC) si në figurën 2.3. Ngasësit e motorëve AC shndërrojnë furnizimin AC në DC duke përdorur drejtues të valës dhe pastaj e kthejnë atë përsëri në frekuencë dhe tension të ndryshueshëm AC duke përdorur një inverter. Ky konvertim arrihet duke përdorur transistor të fuqisë IGBT me metodën e modulimit me gjerësi të pulseve PWM.

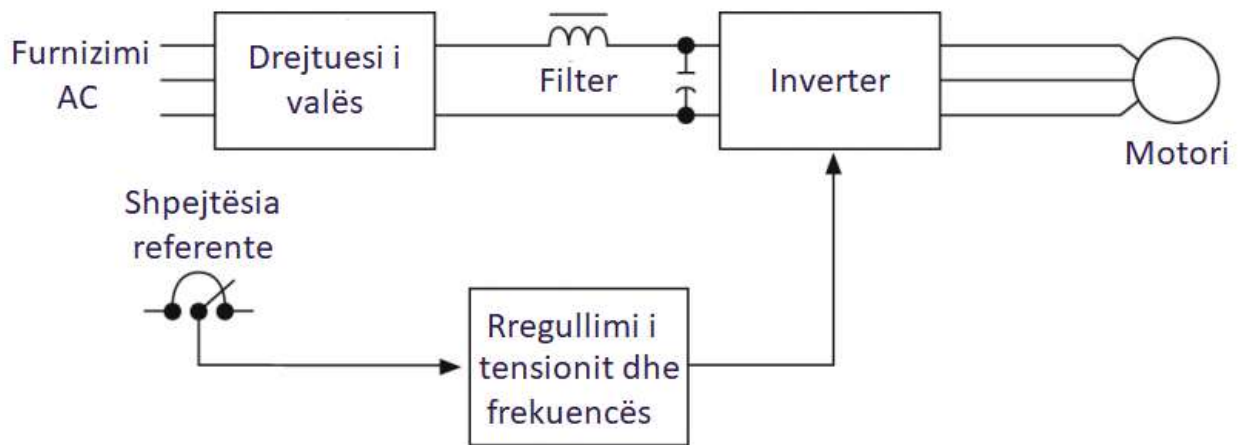


Figura 2.3. Bllok diagrami i një VFD

Qarku elektrik i VFD është paraqitur në figurën 2.4.

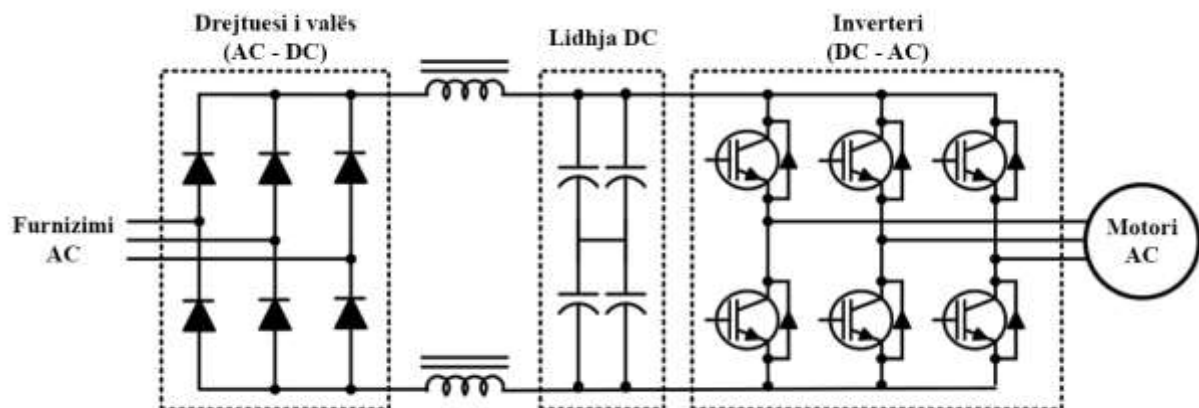


Figura 2.5. Qarku elektrik kryesor i VFD

VFD që kemi përdorur për kontrollim të shpejtësisë së motorit është *Micromaster 420* i prodhuar nga Siemens i paraqitur në figurën 2.6. Ky inverter punon me fuqi në hyrje prej 120W njëfazore deri në 11kW trefazore. *Micromaster 420* mund të përdoret në aplikacione të thjeshta të kontrollit të motorit duke përdorur vetëm cilësimet standarde, kurse për aplikacione më të zgjeruara të kontrollit të motorit përmes listës së plotë të parametrave. *Micromaster 420* mund të përdoret si i pavarur gjithashtu mund të integrohet në sistemet e automatizimit.



Figura 2.6. *Micromaster 420 dhe motori elektrik*

2.4. Ekrani LCD 2x16 dhe butonat

Për paraqitjen e frekuencës aktuale të inverterit, caktimin e frekuencës, ndryshimin e adresës së masterit, ndryshimin e vlerave të parametrave është përdorur LCD 2x16 dhe disa butona si në figurën 2.7. Tastiera është e lidhur në modin 4 bitësh me mikrokontrollerin përmes portit 0.

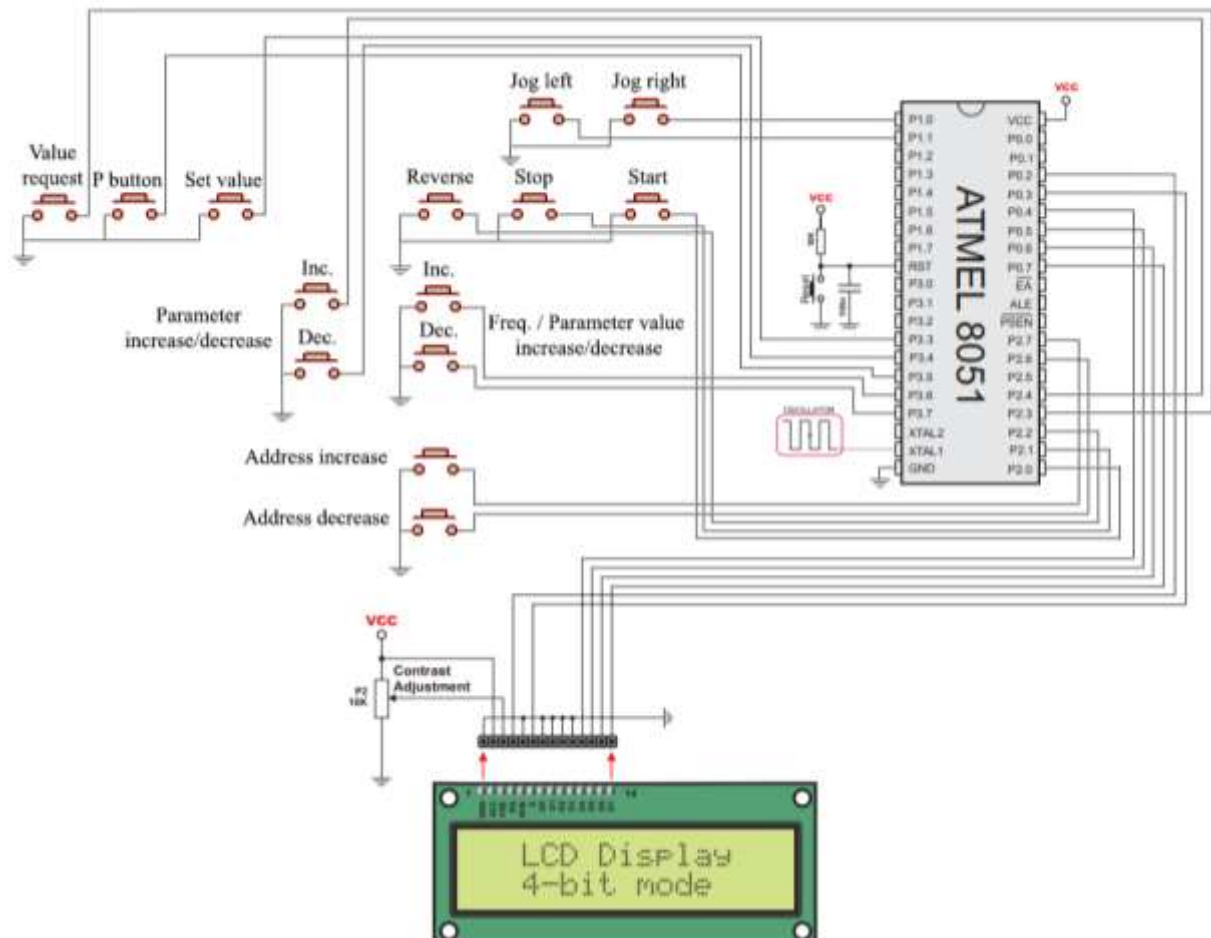


Figura 2.7. Skema e lidhjes së butonave dhe LCD Display në modin 4-bit

3. Softueri i projektit

Në këtë kapitull do të diskutohet për pjesën softuerike të realizimit të protokollit, gjithashtu do të paraqiten format valore të sinjaleve të transmetimit dhe të pranimit të të dhënave. Pjesa e kontrollës në mikrokontrollerin AT89S8253 është programuar në gjuhën programuese C.

3.1. Transmetimi i të dhënave

Për dërgim të të dhënave nga masteri kemi përdorur interruptin e timerit 0 i cili dërgon gjatë gjithë kohës telegrame në slave. Telegrami është caktuar me gjatësi fikse. Gjatësia e PKW është 4 fjalë (8 bajt), ndërsa gjatësia e PZD 2 fjalë (4 bajt), gjithsej 16 bajt duke përfshirë bajtin për start STX, për adresë ADR, për gjatësi LGE dhe BCC. Kalkulimi kohës së karakterit për 2400Baud është bërë përmes formulës: $tch = \frac{11 \cdot 1000}{2400} = 4.853[ms]$.

Funksionet për dërgim të të dhënave:

```
void Transmit_data(uchar tx_data)
{
    ACC = tx_data;
    TB8 = P; // Ngarkohet parity bit
    SBUF = tx_data; // Ngarkohet karakteri në SBUF register
    while (TI == 0); // Pret derisa të dërgohet karakteri
    TI = 0; // Clear TI flag
}

void String(char *str)
{
    int i;
    for (i = 0; i < frame_size - 1; i++) // Dërgohet secili karakter deri në fund të frame
    {
        Transmit_data(str[i]); // Thirret funksioni transmit data
    }
    Transmit_data(BCC(str)); // Thirret funksioni për dërgim të bajtit BCC
}
```

Funksioni i interruptit të timerit 0 për dërgim të frame:

```
void Timer0_ISR (void) interrupt 1
{
    RS485En = 1;                // Për dërgim duhet të vendoset RS-485 pini
    TH0 = 0xE8;
    TL0 = 0x1F;                // Delay 2 kohë të karakterit (2tch)
    TR0 = 1;
    while (TF0 == 0);
    String(frame);              // Thirret funksioni për dërgim të frame
    RS485En = 0;                // Ndalohet pasi të përfundohet transmetimi
    TH0 = 0x00;
    TL0 = 0x00;
    TR0 = 1;
    TF0 = 0;
}
```

Funksioni për llogaritje të BCC:

```
uchar BCC(uchar *frame)
{
    uchar BCC_value = 0;        // Vlera fillestare e BCC është 0
    int i;
    for (i=0;i<frame_size-1;i++)
    {
        BCC_value ^= frame[i];  // Çdo bajt i ardhshëm bëhet XOR me bajtin paraprak
    }
    return BCC_value;
}
```

3.2. Pranimi dhe analizimi i të dhënave

Pranimi i të dhënave bëhet përmes interruptit serik, ku fillimisht pranohen të dhënat pastaj analizohet se a ka ndonjë gabim në paritet. Funkzioni për pranimin dhe analizimin e të dhënave:

```
void Serial_ISR() interrupt 4
{
    if (TF0 == 0){
    if (RI == 1){
        ACC = SBUF;
        rx_data = SBUF;           // Pranimi i të dhënave
        RI = 0;
        if (RB8 == P)             // Pariteti është në rregull për sistemet me paritet çift
        {
            if (data_count == 0){
                if (rx_data == 0x02)           // Frame i ri
                {
                    ReplyFrame[data_count++] = rx_data; // Ngarkohet bajti i parë
                }
            }
            else {
                ReplyFrame[data_count++] = rx_data; // Ngarkohen bajtat e mbetur
                if (data_count >= frame_size){      // deri në fund të frame
                    if (ReplyFrame[data_count-1] == BCC_rx(ReplyFrame))
                    {
                        freq = match(ReplyFrame[data_count-3],ReplyFrame[data_count-2]);
                        if (count == 0){
                            val_select=match(ReplyFrame[data_count-7],ReplyFrame[data_count-6]);
                        }
                    }
                }
                if (freq > 10000){
                    direction = 1;
                }
            }
        }
    }
}
```

```
        freq = 20000 - freq;
    }
    else {
        direction = 0;
    }
    data_count = 0;
    TFO = 1;
}
}
}
}
}
```

3.3. Kontrollimi dhe monitorimi i inverterit (PZD)

Për kontrollim të inverterit nga USS RS-485, parametrin P0700 duhet ti vendoset vlera 5, dhe për vendosje të vlerave të frekuencës parametri P1000 po ashtu duhet të jetë me vlerë 5.

Komanda në fjalën e kontrollit (STW) për JOG djathtas është 057E (hex), JOG majtas 067E, stop 047E, start 047F dhe për lëvizje në kahjen e kundërt me lëvizjen aktuale 0D7F.

Kahja e ngasjes së motorit mund të ndryshohet drejtpërdrejtë duke mos e ndalur inverterin. Për ndërrim të kahjes së JOG pa e ndalur përdoren bitat e kontrollit 8 dhe 9, p.sh. nëse dërgojmë 067E 0000 pas 057E 0000 inverteri do ta ndërroj drejtimin e JOG prej majtas në djathtas pa e ndalur.

Vlerat në setpoint mund të dërgohen si përqindje ose vlera absolute. Mënyra e dërgimit mund të caktohet në parametrin P2009. Në rastin tonë kemi caktuar $P2009 = 1$, d.m.th. me vlera absolute. Në figurën 3.1. është paraqitur forma valore e sekuencës së transferimit të telegrameve ku si shembull kemi zgjedhur frame: *02 0E 01 00 00 00 00 00 00 00 00 00 00 03 E8 33*, ku është dërguar vetëm fjala e dytë e PZD pra vlera setpoint (HSW) me vlerën 1000 (03E8 hex), që i bie 10.00Hz, ndërsa inverteri është i ndalur pasi që nuk është dërguar ndonjë komandë në fjalën e kontrollit (STW). Përgjigja e kthyer nga slave është: *02 0E 01 00 00 00 00 00 00 00 00 FA 31 00 00 C6*. Pasi që baud rate është 2400, vonesa kohore e fillimit duhet të jetë më e madhe se 9.17ms. Vonesa kohore e fillimit është caktuar 2 kohë të karakterit (tch)

dhe siç shihet në figurën 3.1 koha e kthimit të përgjigjes nga slave është 3tch. Koha e kthimit të përgjigjes duhet të jetë më e madhe se vonesa kohore e fillimit, pra e plotëson këtë kusht.

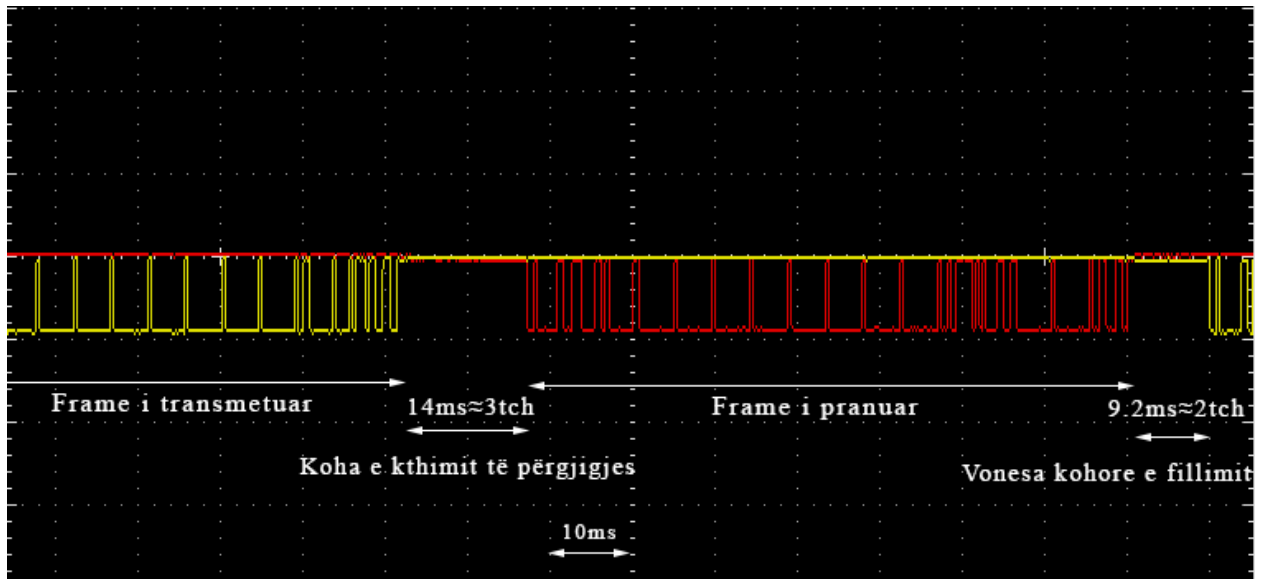


Figura 3.1. Forma valore në osciloskop e transferimit të telegrameve kur inverteri është i ndalur

Pjesa e programit për kontrollimin e pjesës së PZD me butona:

```

if (jog_R == 0){                                     // Kur shtypet butoni në pinin P1.0 vendosen vlerat
    frame[11] = 0x05;                                // për jog right në bajtat përkatës
    frame[12] = 0x7E;
}
else if (jog_L == 0){                                // Pini P1.1 për jog left
    frame[11] = 0x06;
    frame[12] = 0x7E;
}
else if (stop == 0){                                  // Pini P2.1 për stop
    frame[11] = 0x04;
    frame[12] = 0x7E;
}
else if (start == 0){                                 // Pini P2.0 për stop
    frame[11] = 0x04;
    frame[12] = 0x7F;
}
    
```

```

}
else if (reverse == 0){           // Pini P2.2 për stop
    frame[11] = 0x0D;
    frame[12] = 0x7F;
}
if (pcount == 0){               // Nuk lejon të ndryshohet frek. kur përdoret P button
    if (freq_inc == 0){
        split += 100;           // Frekuenca rritet për 1Hz pas çdo shtypje të butonit
    }
    else if (freq_dec == 0){
        split -= 100;           // Frekuenca zvogëlohet për 1Hz
    }
}
frame[13] = split >> 8;
frame[14] = split & 0xFF;       // Frekuenca e ndryshuar vendoset në
}                               // bajtat për setpoint

```

Frame i transmetuar: 02 0E 01 00 00 00 00 00 00 00 00 04 7E 03 E8 9C.

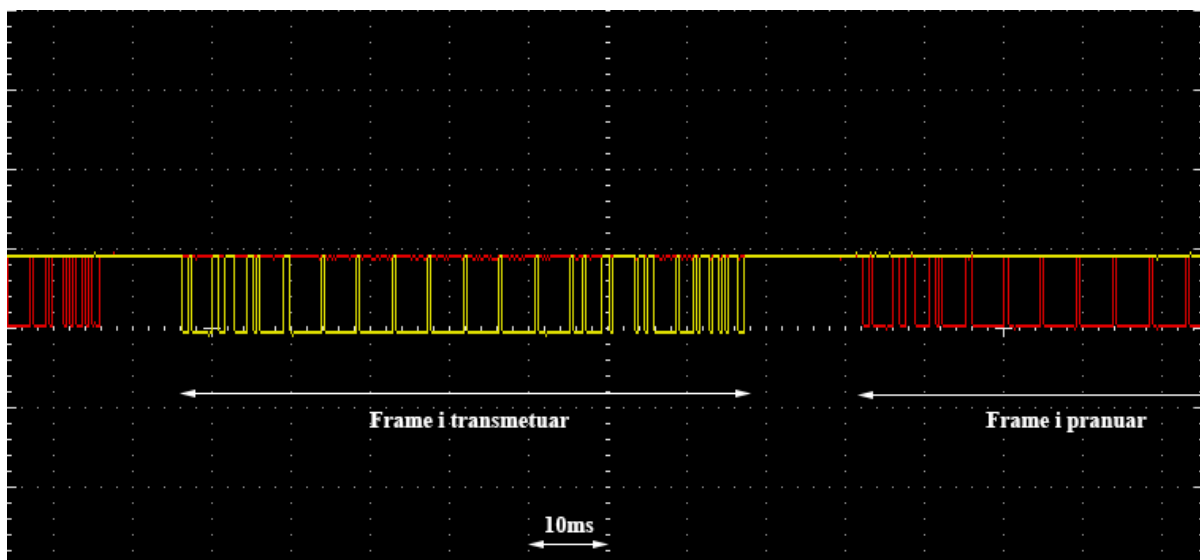


Figura 3.2. Forma valore transmetimit të telegramit për startim me frekuencë 10.00Hz

Në figurën 3.3 është paraqitur forma valore e frame të pranuar kur motori është me drejtim djathtas dhe e ka arrit vlerën e caktuar të frekuencës (10.00Hz).

Frame i pranuar: 02 0E 01 00 00 00 00 00 00 00 00 FB 34 03 E8 09.

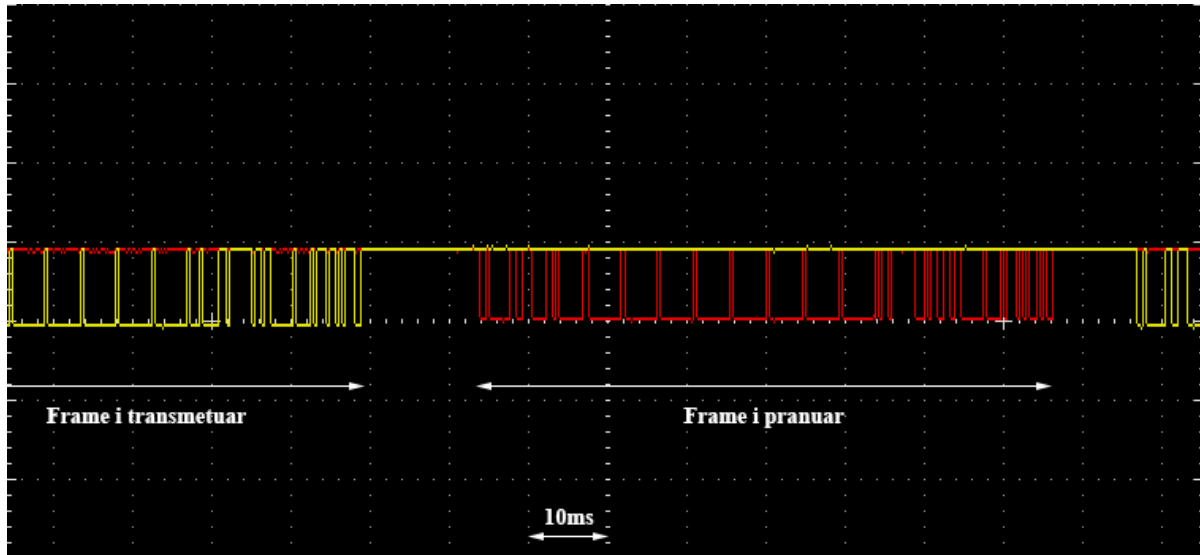


Figura 3.3. Forma valore e frame të pranuar kur motori është me drejtim djathtas me frekuencë 10.00Hz

Pjesa e kodit për paraqitje të vlerës së caktuar dhe vlerës aktuale të frekuencës së inverterit:

```
sprintf(Msg, "%5.2f", (float) freq / 100); // Konvertimi i frekuencës në float
if (direction == 1)
{
    Lcd4_Set_Cursor(1,10);
    Lcd4_Write_String("-");
}
else // Nëse drejtimi është djathtas largohet "-“
{
    Lcd4_Set_Cursor(1,10);
    Lcd4_Write_String(" ");
}
Lcd4_Set_Cursor(1,11);
Lcd4_Write_String(Msg); // Vlera aktuale paraqitet në display
sprintf(Msg, "%3d", split / 100); // Konvertimi në decimal me shenjë
Lcd4_Set_Cursor(1,2);
Lcd4_Write_String(Msg); // Vlera e caktuar paraqitet në display
```

Paraqitja në LCD Display e vlerës aktuale dhe vlerës së caktuar të frekuencës kur inverteri është i ndalur është paraqitur në figurën 3.4, dhe në figurën 3.5 është paraqitur kur inverteri e ka arritur frekuencën e caktuar dhe motori është drejtuar djathtas.

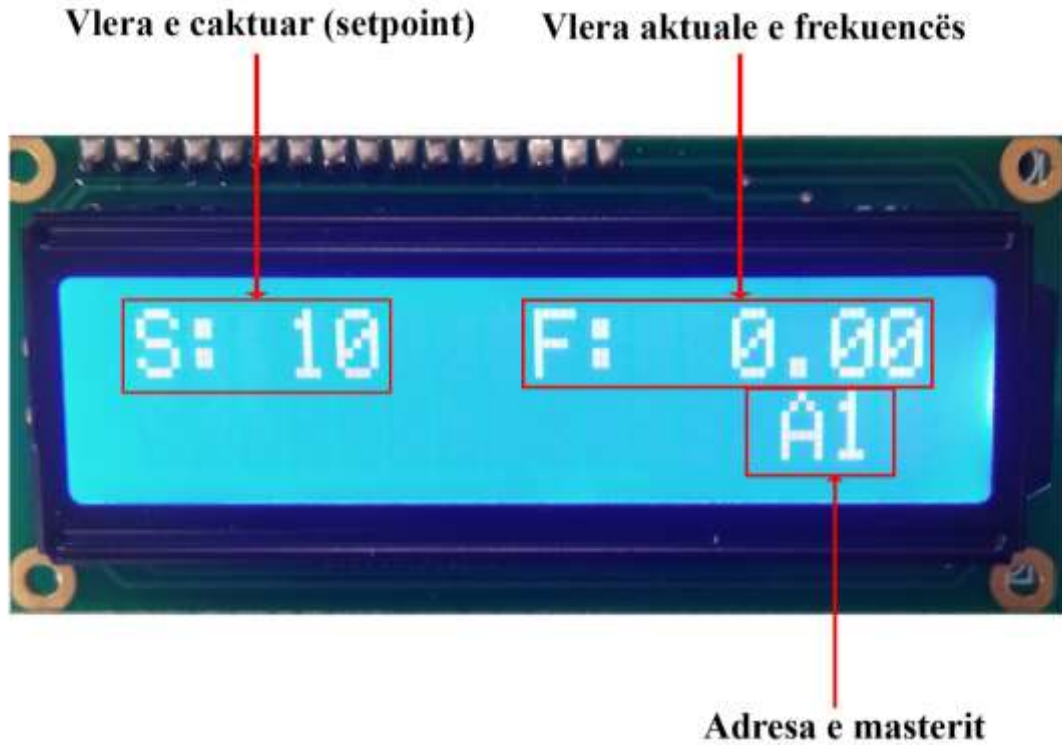


Figura 3.4. Paraqitja në LCD Display e vlerave të frekuencës kur inverteri është i ndalur



Figura 3.5. Paraqitja në LCD Display e vlerave të frekuencës kur inverteri e ka arritur frekuencën e caktuar dhe motori është drejtuar djathtas

Adresa ADR në frame është parazgjedhur me vlerë 1. Ndryshimi i adresës së masterit mund të bëhet duke shtypur butonat që janë të lidhur në pinat P2.6 dhe P2.7 të mikrokontrollerit. Butoni i lidhur në pinin P2.6 e rrit adresën për një njësi kurse butoni i lidhur në pinin P2.7 e zbritet për një njësi. Adresa e masterit është paraqitur në display si në figurën 3.3.

Pjesa e kodit për ndryshim të adresës së masterit:

```
if (addr_up == 0){  
    ADR += 1;                // Rritet adresa e masterit për 1 njësi  
    Delay(8);  
}  
else if (addr_low == 0){  
    ADR -= 1;                // Zbritet adresa e masterit për 1 njësi  
    Delay(8);  
}  
if (ADR > 31){               // Kufizon rritjen e adresës deri në 31  
    ADR = 0;                // dhe nëse e tejkalon 31 kthehet në 0  
}  
if (ADR < 0){               // Kufizon zbritjen e adresës deri në 0  
    ADR = 31;              // dhe nëse zbritet nën 0 kthehet në 31  
}  
sprintf(Msg, "A%B", ADR);  // Konvertohet në unsigned decimal  
Lcd4_Set_Cursor(2, 13);    // Zgjedhet se në cilën pjesë të display paraqitet  
Lcd4_Write_String(Msg);    // Paraqet adresën në display
```

3.4. Leximi dhe shënimi i vlerave të parametrave (PKW)

PKW mund të përdoret në master me gjatësi 4 fjalë edhe kur P2013 është me vlerë variabile, pra P2013 = 127, por mund të ndryshoj kur slave kthen përgjigje në varësi të tipit të vlerës së parametrin, prandaj gjatësinë e PKW kemi zgjedhur 4 fjalë në parametrin P2013, pra P2013 = 4. Kur gjatësia e PKW është 4, dhe vlerat e parametrin janë një fjalëshe atëherë do të paraqiten në PWE2 dhe jo në PWE1. Për leximin e vlerave të parametrave duhet të përdoret Task ID 1 “kërkesa për vlerë të parametrave”. Reply ID do të jetë 1 (gjatësi një fjalë), 2 (dy fjalë) ose 7 (gabim). Në punim kemi zgjedhur vetëm parametrat: P0700, P1000, P2011, r2024, r2025.

Numrat me parashenjën “r” janë parametra vetëm për lexim, vlera e të cilëve mund të lexohet por jo edhe të shkruhet.

Në figurën 3.6 është paraqitur forma valore në osciloskop kur dërgohet kërkesa për lexim të vlerës së parametrut P0700. Paraprakisht është dërguar edhe komanda për stop 04 7E (hex).

Për lexim Task ID duhet të jetë 1 dhe Parameter ID 0700 = 2BC (hex). Në këtë rast është transmetuar frame: `02 0E 01 12 BC 00 00 00 00 00 00 04 7E 03 E8 32`.

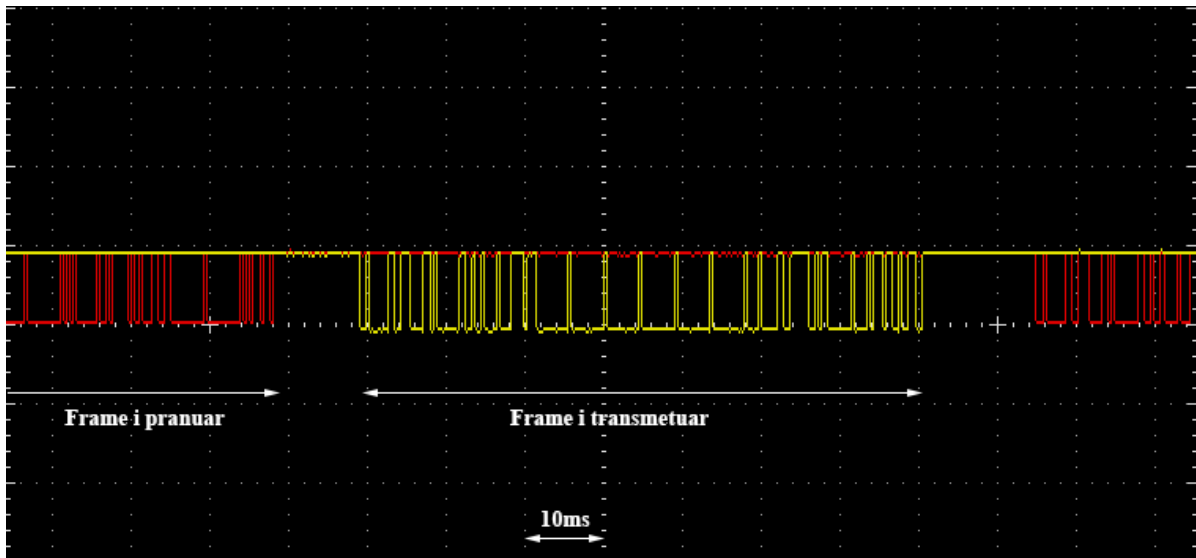


Figura 3.6. Forma valore në osciloskop kur bëhet kërkesa për lëxim të vlerës së parametrut P0700

Ndërsa frame i pranuar do të jetë: `02 0E 01 12 BC 00 00 00 00 00 05 FA 31 00 00 6D`. Në figurën 3.7 është paraqitur forma valore e frame-it të pranuar. Nga përgjigjja shihet se parametri P0700 ka vlerë me gjatësi një fjalë dhe vlera e tij është 0005. Nëse gjatësia e PKW është variable, pra P2013 = 127, vlera e parametrut do të shfaqet në PWE1 dhe PWE2 nuk do të ekzistojë fare. PKW i pranuar do të jetë: `12 BC 00 00 00 05`. Pasi që ne kemi zgjedhur gjatësinë e PKW = 4, PKW i pranuar është `12 BC 00 00 00 00 05`.

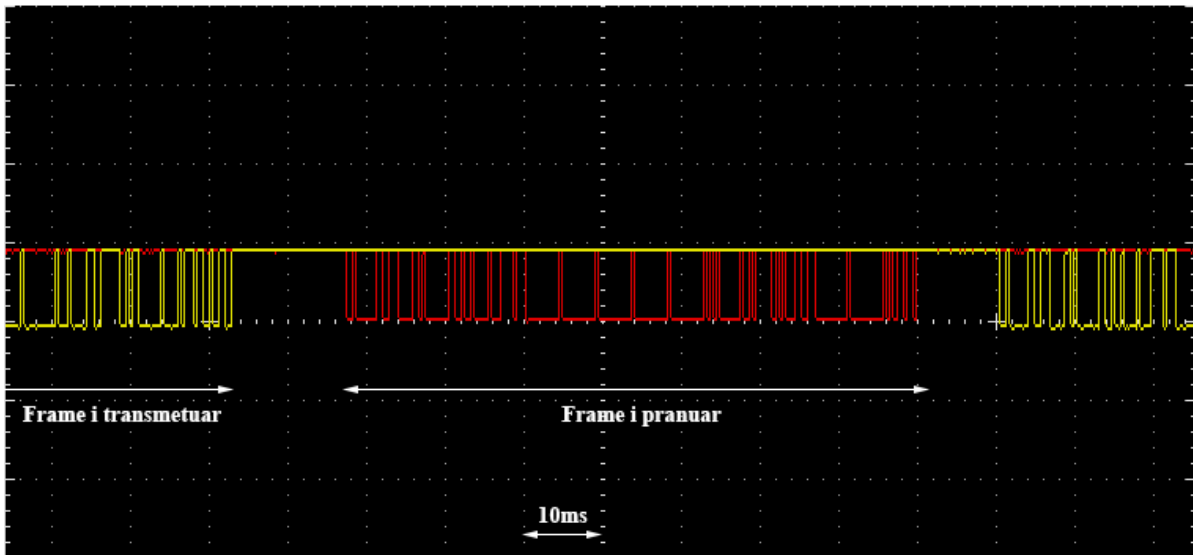


Figura 3.7. Forma valore në osciloskop e frame-it të pranuar për vlerën e parametrut P0700

Në vazhdim për shënimin e vlerave të parametrut kemi zgjedhur parametrin P1000, ku forma valore e tij është paraqitur në figurën 3.8. Për shënim të vlerës së parametrut me gjatësi një fjalë, Task ID duhet të jetë 2 për ruajtje të përkohshme (vetëm në RAM) ose 14 (E hex) për ruajtje të përhershme (RAM dhe EEPROM) dhe Parameter ID 1000 = 3E8 (hex). Frame i transmetuar është: `02 0E 01 23 E8 00 00 00 00 00 01 00 00 03 E8 2C`. Pra kemi dërguar frame për shënimin e vlerës 1 në parametrin P1000. Për shënim të vlerave kur gjatësia e PKW është variabile P2013 = 127 vlen e njëjta rregull, kur vlera e parametrut është një fjalë vlera duhet të shënohet në PWE1 e jo në PWE2 si në rastin tonë. P.sh. për parametrin 1000 vlera e PKW shënohet në formën `23 E8 00 00 00 00 01 00`.

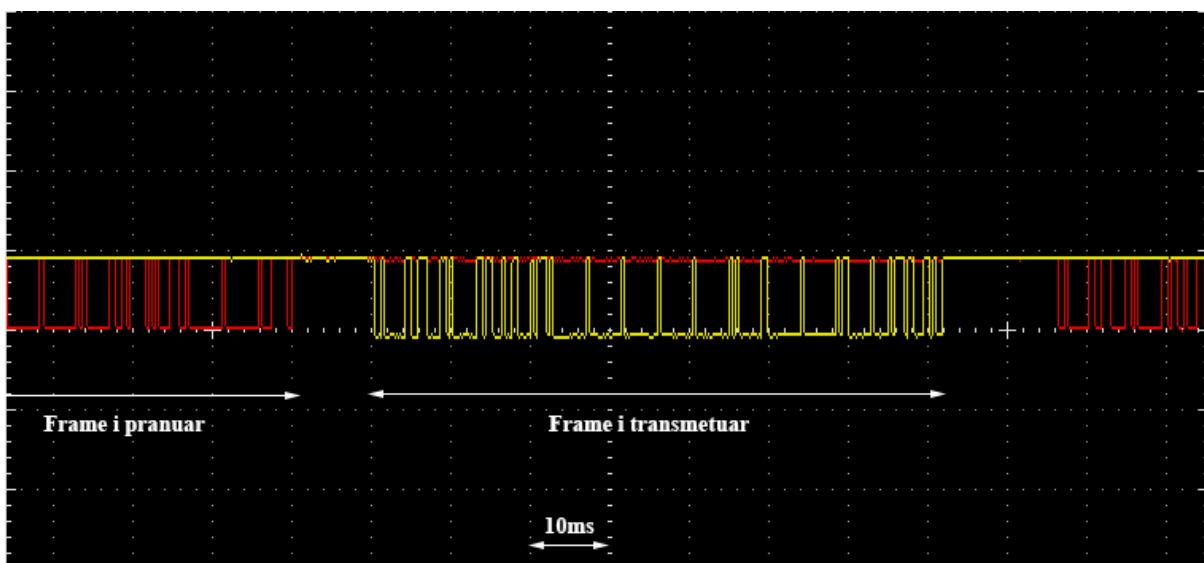


Figura 3.8. Forma valore në osciloskop e frame-it për shënim të vlerës 1 në P1000

Përgjigjja e kthyer nga slave do të jetë: `02 0E 01 13 E8 00 00 00 00 00 01 FA 31 00 00 3C`.

Forma valore është paraqitur në figurën 3.9. Për leximin apo shënimin e vlerave në parametrat prej 2000 deri në 3999, duhet të vendoset PNU në fjalën e dytë (IND). P.sh. për leximin e vlerës së parametrat P2011 duhet të dërgohet PKW në formën: `100B 8000 0000 0000`. Pra `2011 = 00B` (hex) dhe biti i 15 i IND. Nëse parametri P2011 ka vlerën 1, përgjigjja do të jetë: `100B 8000 0000 0001`.

Për të zgjedhur parametrat programimi është bërë në atë mënyrë që së pari duhet të shtypet butoni që është i lidhur në pinin P3.5 të mikrokontrollerit dhe pastaj do të shfaqet shkronja “P” në LCD display, nëse shtypet për të dytën herë shkronja “P” dhe parametrat e paraqitur do të fshihen nga LCD.

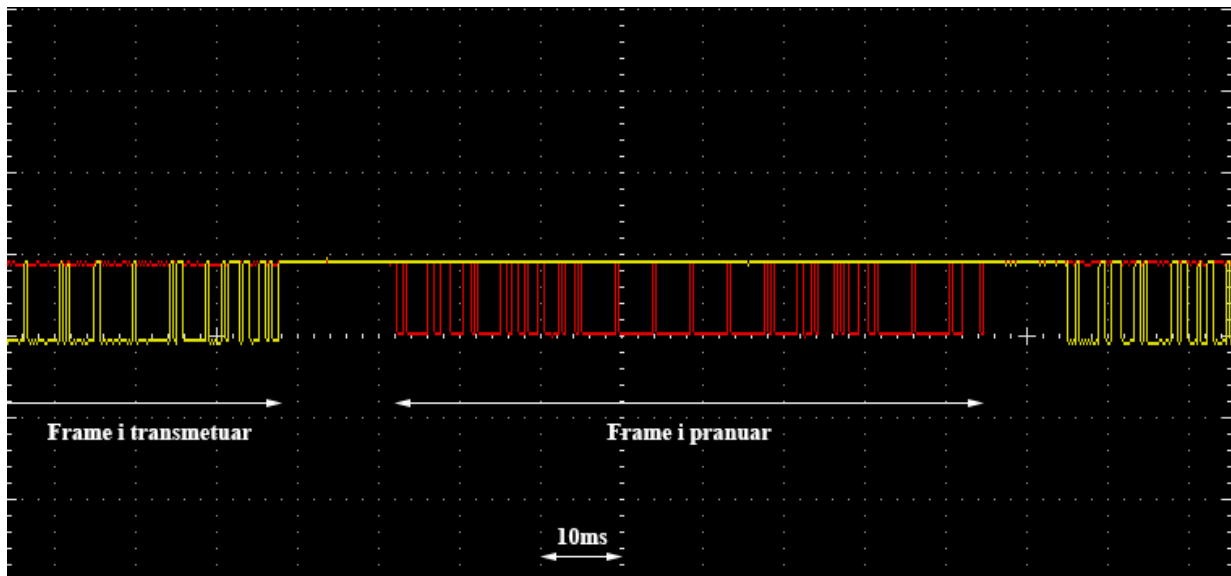


Figura 3.9. Forma valore në osciloskop e frame-it të pranuar pas kërkesës për shënimin e vlerës 1 në parametrin P1000

Pjesa e kodit për paraqitjen e shkronjës “P”:

```
if (pbutton == 0) // Kur shtypet butoni në pinin P3.5
{
    Lcd4_Set_Cursor(2,0); // Zgjedhet se në cilën pjesë të display
    pcount++; // Rritet për 1
    switch (pcount)
    {
        case 1:
```

```

        Lcd4_Write_String("P");           // Shfaqet shkronja "P" në display
        break;
    default:                               // Nëse pcount e ndryshme prej 1
        pcount = pselect = val_select = 0; // vlerat rikthehen në zero
        Lcd4_Write_String("      ");      // fshihet çdo parametër nga display
        break;
    }
    Delay(10);
}

```

Në figurën 3.10 është paraqitur LCD display kur shtypet butoni në pinin P3.5



Figura 3.10. Paraqitja e shkronjës "P" në LCD display kur shtypet butoni në pinin P3.5

Kalimi nëpër parametra bëhet përmes butonave të lidhur në pinat P2.4 dhe P3.4 të mikrokontrollerit. Kur shtypet butoni në pinin P2.4 bëhet kalimi në parametrin vijues, ndërsa kur shtypet butoni në pinin P3.4 bëhet kthimi në parametrin paraprak. Kalimi në parametrin P0700 është paraqitur në figurën 3.11. Kërkesa për lexim të vlerës së parametrin bëhet në momentin që kalohet në ndonjë parameter. Kodi për kalimin nëpër parametra:

```

if (pcount == 1){
    Lcd4_Set_Cursor(2,0);           // Zgjedhet se në cilën pjesë të display
    if (inc == 0){                  // Kur shtypet butoni në pinin P2.4

```

```
pselect++; // kalohet në parametrin vijues
count = 0;
parameters();
Delay(13);
}
if (dec == 0){ // Kur shtypet butoni në pinin P3.4
    pselect--; // kalohet në parametrin paraprak
    count = 0;
    parameters();
    Delay(13);
}
}
void parameters(){
switch (pselect){
    case 1:
        param_h = 0x12;param_l = 0xBC;
        PNU = 0x00;
        Lcd4_Write_String("P0700: ");
        break;
    case 2:
        param_h = 0x13;param_l = 0xE8;
        PNU = 0x00;
        Lcd4_Write_String("P1000: ");
        break;
    case 3:
        param_h = 0x10;param_l = 0x0B;
        PNU = 0x80;
        Lcd4_Write_String("P2011: ");
        break;
    case 4:
        param_h = 0x10;param_l = 0x18;
        PNU = 0x80;
        Lcd4_Write_String("r2024: ");
        break;
```



```
case 5:
    param_h = 0x10;param_l = 0x19;
    PNU = 0x80;
    Lcd4_Write_String("r2025: ");
    break;
default:
    param_h = 0x00;param_l = 0x00;
    PNU = 0x00;
    val_select = 0;
    Lcd4_Write_String("P      ");
    break;
}
if (pselect > 5){
    pselect = 0;
}
if (pselect < 0){
    pselect = 6;
}
}
```



Figura 3.11. Paraqitja në LCD kur bëhet kalimi në parametrin P0700

Pas leximit të vlerës së parametrut, vlera e tij do të shfaqet në LCD pasi të shtypet butoni i lidhur në pinin P2.3, në këtë mënyrë ndalohej dërgimi i kërkesës për lexim të vlerës por lejohet kalimi nëpër vlera të parametrut përmes butonëve në pinat që janë përdorur për ndryshim të setpoint të inverterit. Vlera e setpoint nuk mund të ndryshohet në momentin që është shtypur butoni për shkronjën “P”. Pas zgjedhjes së vlerës e cila do të shënohet në atë parametër duhet të shtypet butoni në pinin P3.3, me shtypjen e të cilit bëhet kërkesa për ndryshim të vlerës. Në figurën 3.12 është paraqitur LCD display kur shfaqet vlera e lexuar parametrut.



Figura 3.11. Paraqitja në LCD e vlerës së lexuar nga parametri P0700

Kodi për vendosjen e vlerave të parametrut:

```
if (pselect != 0){
if (cont == 0){                                     // Kur shtypet butoni në pinin P3.3
    count++;
    Delay(10);
}
if (count > 1){
    count = 0;
}
if (count == 0){
    Lcd4_Set_Cursor(2,6);
    Lcd4_Write_String("  ");
}
```

```
}
if (count == 1){
    if (freq_inc == 0){
        val_select++;
        Delay(5);
    }
    else if (freq_dec == 0){
        val_select--;
        Delay(5);
    }
}
if (pselect == 1){
    if (set == 0){ // Kur shtypet butoni në pinin P3.3
        param_h=0x22;param_l=0xBC;
    }
    else {
        param_h=0x12;param_l=0xBC;
    }
    max_v = 6; // Vlera maksimale e parametrut
}
else if (pselect == 2){
    if (set == 0){
        param_h = 0x23;param_l = 0xE8;
    }
    else {
        param_h = 0x13;param_l = 0xE8;
    }
    max_v = 6;
}
else if (pselect == 3){
    if (set == 0){
        param_h = 0x20;param_l = 0x0B;
        PNU=0x80; // Nëse parametri > 1999 vendoset bit 15
    }
    else {
```

```
    param_h = 0x10; param_l = 0x0B;
    PNU = 0x80;
}
    max_v = 31;
}
if (val_select > max_v){
    val_select = 0;
}
if (val_select < 0){
    val_select = max_v;
}
Lcd4_Set_Cursor(2,6);
sprintf(Msg, "%d ", val_select);           // Bëhet konvertimi në vlera decimale
Lcd4_Write_String(Msg);                   // Paraqitet në LCD vlera e parametrin
}
}
```

4. Konkluzioni

Në këtë temë u prezentua realizimi i protokollit USS për komunikim me njësitë ngasëse të Siemens ku si njësi ngasëse është marrë *Micromaster 420*. Protokolli është realizuar duke përdorur mikrokontrollerin AT89S8253 i cili është programuar në gjuhën programuese C. Ky softuer është i aftë për kryerjen e funksioneve të nevojshme për ngasjen e njësisë duke e zëvendësuar panelin kontrollues me një kontrollim më të lehtë duke përdorur më shumë butona.

E gjithë puna lidhur me të është diskutuar për të kuptuar më mirë se si funksionon protokolli USS dhe se me pak përmirësime mund të përdoret edhe në seri tjera të njësive ngasëse të Siemens.

U tregua për komunikimin ndërmjet njësive. Këto njësi mund të komunikojnë në largësi deri në 1200m duke përdorur standardin për komunikim serik RS-485. Gjatë punimit u shpjegua transferimi i të dhënave sipas rregullave të protokollit USS. Pastaj janë paraqitur grafika për raste të ndryshme të cilët shpjegojnë më mirë vonesën kohore të fillimit për transmetim të të dhënave dhe për kohën e kthimit të përgjigjes.

Në versionet e ardhshme mund të shqyrtohet mundësia e realizimit të protokollit duke përdorur kristalin e kuarcit të mikrokontrollerit që më është i përshtatshëm për komunikim serik me shpejtësi më të madhe të transferimit të të dhënave pa gabime.

5. Literatura

- [1] Universal Serial Interface Protocol, USS Protocol, Siemens, Specification, 1999.
- [2] AT89S8253, Microcontroller Datasheet, Atmel Corporation, 2010.
- [3] Using USS Protocol with Micromaster MM420, 2001.
- [4] “Variable Frequency Drives, Energy Efficiency Reference Guide”, Natural Resources Canada, 2009.
- [5] Siemens Micromaster 420 Manual, User Documentation, 10/2006.
- [6] Parameters Micromaster 420, 2001.
- [7] Siemens Micromaster 420, Getting Started Guide, 09/2000.
- [8] Martin Brown, “Siemens Standard Drives, Application Handbook”, 12/1997