

Primjena ARM arhitekture u automatizaciji

Ladiš, Marino

Undergraduate thesis / Završni rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zadar / Sveučilište u Zadru**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:162:742479>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-06-30**



Sveučilište u Zadru
Universitas Studiorum
Jadertina | 1396 | 2002 |

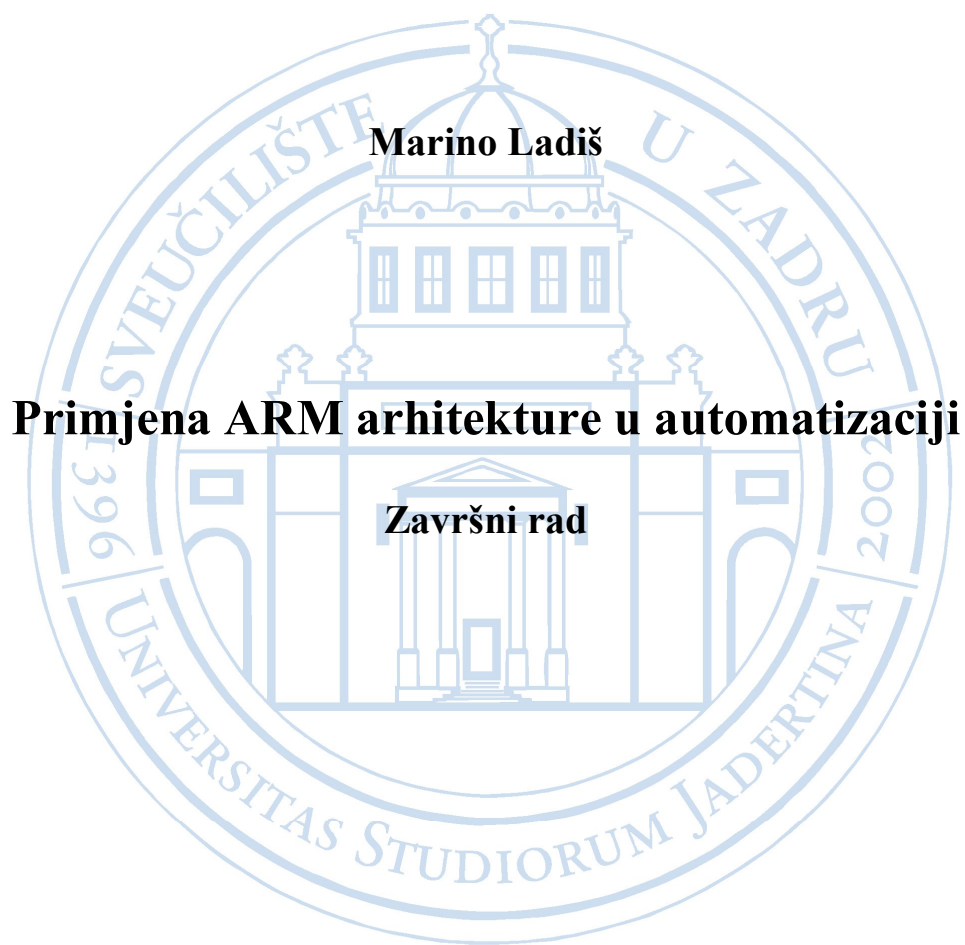
Repository / Repozitorij:

[University of Zadar Institutional Repository](#)



Sveučilište u Zadru

Stručni preddiplomski studij Informatičke tehnologije



Marino Ladiš

Primjena ARM arhitekture u automatizaciji

Završni rad

Zadar, 2021.

Sveučilište u Zadru

Stručni preddiplomski studij Informatičke tehnologije

Primjena ARM arhitekture u automatizaciji

Završni rad

Student/ica:

Marino Ladiš

Mentor/ica:

dr. sc. Frane Urem

Zadar, 2021.



Izjava o akademskoj čestitosti

Ja, **Marino Ladiš**, ovime izjavljujem da je moj **završni** rad pod naslovom **Primjena ARM arhitekture u automatizaciji** rezultat mojega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na izvore i radove navedene u bilješkama i popisu literature. Ni jedan dio mojega rada nije napisan na nedopušten način, odnosno nije prepisan iz necitiranih radova i ne krši bilo čija autorska prava.

Izjavljujem da ni jedan dio ovoga rada nije iskorišten u kojem drugom radu pri bilo kojoj drugoj visokoškolskoj, znanstvenoj, obrazovnoj ili inoj ustanovi.

Sadržaj mojega rada u potpunosti odgovara sadržaju obranjenoga i nakon obrane uređenoga rada.

Zadar, 19. siječanj 2021.

Primjena ARM arhitekture u automatizaciji

Završni rad primjene ARM (engl. Advanced RISC Machines) arhitekture u automatizaciji objasnit će osnovne karakteristike ARM arhitekture i osnovne koncepte industrijske automatizacije. Sagledat će se povijesni napredak tih dviju, možda na izgled nepovezanih, industrija. Objašnjena je osnovna funkcionalnost postojećeg modula PLC (engl. Programmable logic controller) uređaja, kako bi se stekao uvid u trenutne zahtjeve tržišta. Kroz jednostavan primjer, pokazat će se korištenje mikrokontrolera koji je baziran na ARM arhitekturi. Unutar primjera bit će razvijen dio funkcionalnosti koja se primjenjuje u odabranom modulu PLC uređaja. Primjena ARM arhitekture u automatizaciji će biti potkrijepljena postojećim solucijama tvrtke STMicroelectronics koja se prilagodila ovom novom rastućem tržištu.

Ključne riječi

ARM, automatizacija, pulsno-širinska modulacija, programabilni logički kontroler

Popis korištenih kratica

ARM	Advanced RISC Machines
PLC	Programmable logic controller
PWM	Pulse-width modulation
RISC	Reduced instruction set computer
CISC	Complex instruction set computer
SSD	Solid state drive
LCD	Liquid crystal display
LED	Light emitting diode
DC	Direct current
AC	Alternating current
UART	Universal asynchronous receiver-transmitter
CAN	Controller area network
TQM	Total Quality Management
GSM	Global System for Mobile Communications

Sadržaj

1. Uvod	1
1.1. Povijest procesora ARM	1
1.2. Karakteristike arhitekture ARM	2
1.3. Industrijska automatizacija.....	4
1.4. Programabilni logički kontroler.....	5
2. Rasprava.....	9
2.1. Karakteristike jezgri ARM Cortex.....	9
2.2. Pulsno-širinska modulacija	10
2.3. Pulsno-širinska modulacija koristeći ARM Cortex-M4	11
2.3.1. Opis najznačajnijih registara	11
2.3.2. Programski kod	14
2.3.3. Objašnjenje programskog koda	15
2.3.4. Primjer rada programa.....	16
2.4. Izlazni modul 4402 PWM	17
2.4.1. Napajanje modula i spajanje trošila	20
2.4.2. Mod pulsno-širinska modulacija.....	21
2.4.3. Mod Dither kontrola.....	21
2.4.4. Početno stanje uređaja	22
2.4.5. Primjene modula	23
2.5. Fieldbus i industrijski ethernet	24
3. Zaključak.....	26
Summary	27
Literatura.....	28

1. Uvod

1.1. Povijest procesora ARM

ARM (engl. Advanced RISC Machines) je naziv tvrtke formirane 1990. godine kao zajednički pothvat tvrtki Apple Computers, Acorn Computer Group i VLSI Technology. ARM je nastao kao odgovor na dominaciju 8-bitnih procesora s kompleksnim skupom naredbi zvanih CISC (engl. Complex Instruction Set Computers). Ta dominacija je bila prekinuta trima projektima: IBM 801, Berkeley RISC i Stanford MIPS. Kroz sva tri projekta predstavljen je novi pristup projektiranju procesora, temeljen na jednostavnim naredbama koje se izvode velikom brzinom. RISC (engl. Reduced Instruction Set Computers) procesori su za razliku od CISC procesora pokazivali izuzetne performanse uz jeftiniju sklopovsku izvedbu. Projektanti u Acorn Computer Group su vidjeli prednost u takvom tipu arhitekture kako bi stvorili svoj procesor bez potrebe dugogodišnjeg razvoja te angažiranja velikog broja projektanata i financijskih sredstava. Tako je nastao prvi procesor ARM čija je ime bila kratica od Acorn RISC Machine. Kako je arhitektura prvog procesora ARM postala izuzetno uspješna na tržištu, tako je od dijela tvrtke Acorn Computers Group osnovana nova tvrtka Advanced RISC Machines (Kovač, M. 2015.).



Slika 1.1.1. Prvo sjedište ARM-a

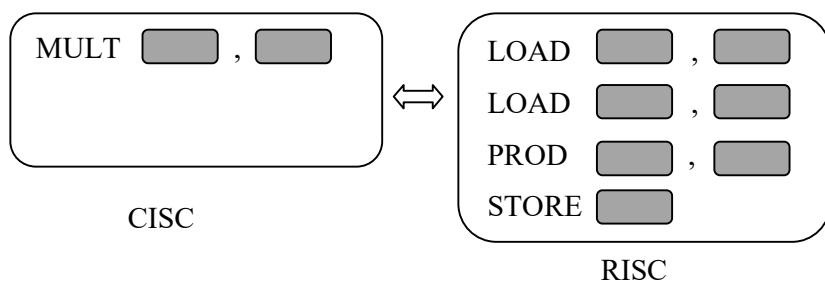
(Walshe, B. A Brief History of Arm: Part 1. 2015)

Ciljevi novonastale tvrtke su bili širenje tržišta i daljnji razvoj arhitekture ARM. ARM je tvrtka čiji se poslovni model zasniva na licenciranju arhitekture svojih procesora. Takav poslovni model je bio rijedak u to vrijeme, a temeljio se na tome da partneri otkupljuju licence kako bi proizvođili procesore. Prvi veći partner koji je koristio ARM licence je Texas Instruments. Time je ARM mogao ulagati u daljnji razvoj isplativih proizvoda. Daljnje ponude od tvrtke Samsung i Sharp su pridonijele u rastu popularnosti ARM procesora. Naporan rad ARM-a se isplatio 1994. godine, za vrijeme mobilne revolucije, kada su mali mobilni uređaji postali stvarnost. Tvrtka Texas Instruments je savjetovala Nokiu da koriste ARM procesor za njihov novi GSM (engl. Global System for Mobile Communications) mobilni telefon. Nokia 6110 je bio veliki uspjeh, što je doprinijelo daljnjem napretku ARM-a (Walshe, B. A Brief History of Arm: Part 1. 2015).

1.2. Karakteristike arhitekture ARM

ARM procesori su bazirani na RISC arhitekturi. Iz tog razloga, važno je razumjeti RISC arhitekturu i njezine prednosti i nedostatke u odnosu na CISC arhitekturu.

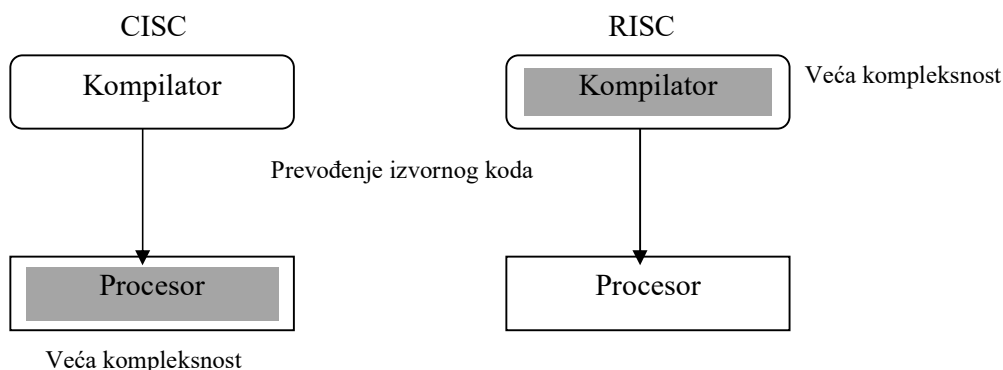
Na početku razvoja računarstva procesori su bili jednostavni, ali im je kompleksnost brzo rasla. Kompleksnost im je rasla zbog sve većeg broja procesorskih naredbi, koje su olakšavale prevođenje programa iz viših programskih jezika, štedjele memoriju zbog manjeg broja naredbi i ubrzale rad zbog manjeg broja dohvata naredba iz memorije. Za svaku dodatnu naredbu bio je potreban sve veći broj tranzistora. Time je nastalo povećanje troškova izrade procesora, povećano trošenje energije i grijanje procesora. Bitna karakteristika CISC arhitekture je ta da procesorske naredbe u memoriji imaju različitu dužinu.



Slika 1.2.1. Usporedba CISC i RISC naredbi

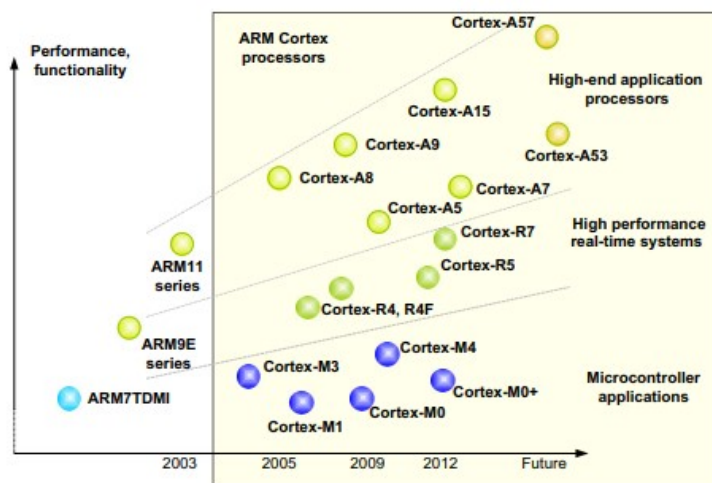
Za razliku od CISC arhitekture, RISC arhitektura sadrži vrlo jednostavne i malobrojne naredbe. S nekoliko naredba RISC arhitekture može se izvesti ista funkcija koju izvodi

pojedina CISC naredba. Smanjenjem broja procesorskih naredbi i ujednačavanjem njihove dužine, uspjela se povećati brzina izvođenja programa i troškovi izrade. Budući da se smanjio broj naredbi povećala se potrošnja memorijskog prostora za pohranu programskog koda i otežalo se prevođenje koda viših programskih jezika. Smanjenjem cijene memorije i razvojem prevoditelja koda, RISC arhitektura postala je sve zastupljenija.



Slika 1.2.2. Kompleksnost kompilatora u RISC i CISC arhitekturi

Arhitektura procesora ARM je prepoznatljiva po svojoj jednostavnosti te po izvedbi visokih performansa. Zbog toga su proizvedeni uređaji niske potrošnje, što odgovara za proizvodnju uređaja napajanih baterijom. Iz tog razloga procesori arhitekture ARM mogu se pronaći od 60% ukupnog broja proizvedenih mobilnih uređaja. No procesori ARM arhitekture nisu samo zastupljeni u mobilnim uređajima, već se koriste u televizijama, prijenosnim računalima, automobilima, mrežnim uređajima itd. Kako se spomenuti uređaji razlikuju po potrebama procesora, ARM je kategorizirao procesore u nekoliko skupina (Yiu, J. 2013.).



Slika 1.2.3. Kategorije ARM procesora (Yiu, J. 2013.)

1.3. Industrijska automatizacija

Industrijska automatizacija je upotreba upravljačkih sustava, poput računala ili robota i informacijskih tehnologija za rukovanje različitim procesima i strojevima u industriji koji zamjenjuju čovjeka. To je drugi korak izvan mehanizacije u opsegu industrijalizacije.

Ranija svrha automatizacije je bila povećati produktivnost i smanjiti troškove ljudskih resursa. Međutim, danas se fokus automatizacije preusmjerio na povećanje kvalitete i fleksibilnosti u proizvodnom procesu. U automobilskoj industriji ugradnja klipova u motor nekad se izvodila ručno sa stopom pogrešaka 1-1,5%. Trenutno se ovaj zadatak izvodi pomoću automatiziranih strojeva sa stopom pogrešaka od 0,00001%.

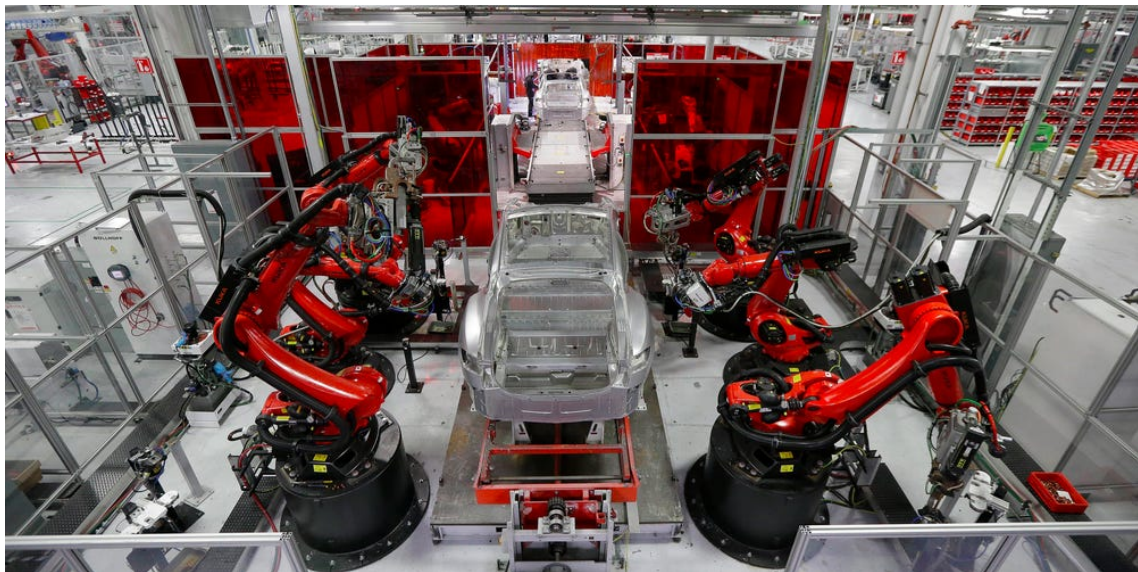
Prednosti industrijske automatizacije:

- Niži operativni troškovi
- Visoka produktivnost
- Visoka kvaliteta
- Visoka razina fleksibilnost
- Visoka točnost informacija
- Visoka sigurnost

Mana industrijske automatizacije:

- Visoki početni trošak

([What is Industrial Automation?. 2013](#)).



Slika 1.3.1. Roboti izrađuju Tesla Model S automobil

(<https://www.businessinsider.com/a-rare-look-inside-teslas-electric-car-factory-2016-2>)

1.4. Programabilni logički kontroler

Programabilni logički kontroler (engl. Programmable logic controller ili skraćeno PLC) je najčešće korištena tehnologija za industrijsku kontrolu procesa. PLC je računalo industrijske namjene koje je moguće programirati s ciljem izvođenja kontrolnih funkcija. Nastao je kao zamjena za relejnu logiku.

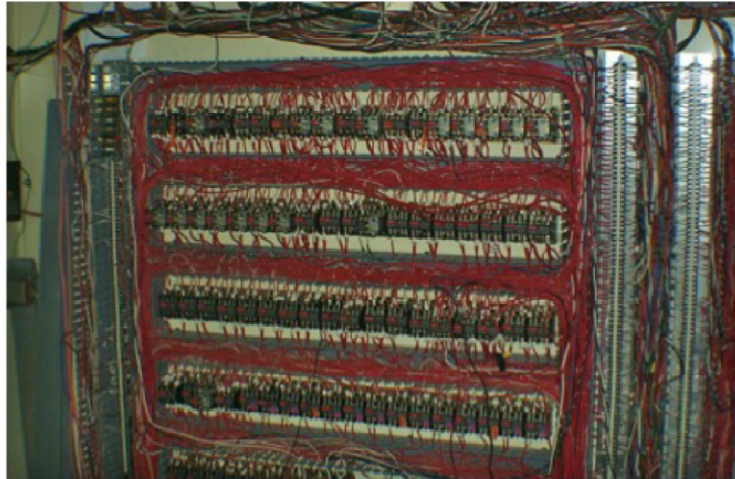


Slika 1.4.1. Razni tipovi programabilnih logičkih kontrolera

(Petruzella, F. D. 2017.)

Prednosti PLC-a u odnosu na kontrolu industrijskog procesa relejnom logikom su:

- manji troškovi
- poboljšana pouzdanost
- veća fleksibilnost
- veća brzina odaziva
- lakše otkrivanje kvara
- mogućnost komunikacije s drugim uređajima



Slika 1.4.2. Relejna kontrolna ploča

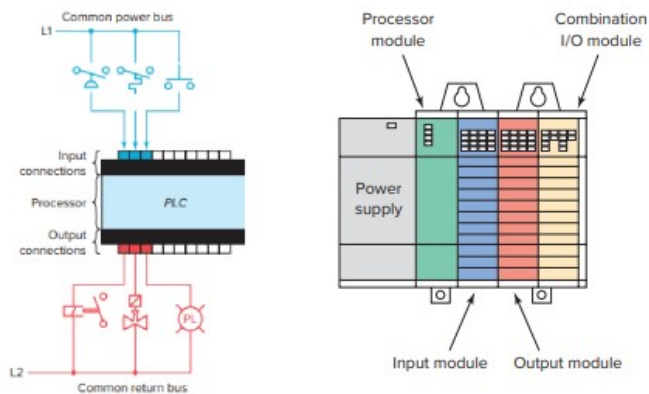
(<https://www.innovativeautomation.com/plc-programming-automation>)



Slika 1.4.3. PLC kontrolna ploča

(<http://rinktechengineering.com/plc-based-control-panels.html>)

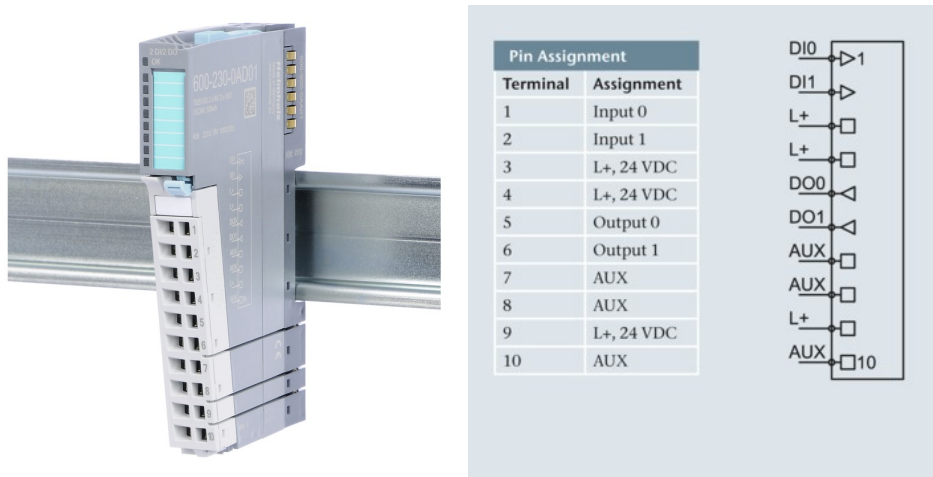
Najčešće PLC možemo podijeliti na procesorsku jedinicu, sekciju ulaza i izlaza, napajanje i uređaj za programiranje. PLC može imati otvorenu ili zatvorenu arhitekturu. Otvorena arhitektura je ona u kojoj se može ostvariti komunikacija između uređaja i programa različitih proizvođača. Zatvorena arhitektura je ona u kojoj proizvođač svojim standardima ne dozvoljava komunikaciju sa uređajima i programima različitog proizvođača. Najzastupljeniji na tržištu je PLC zatvorene arhitekture.



Slika 1.4.4. Fiksni (s lijeva) i modularni (s desna) PLC

(Petruzella, F. D. 2017.)

PLC dijelimo i na modularne i fiksne. Fiksni PLC sadrži zajedno upakirane ulaze i izlaze, koji se ne mogu proširivati. Takav PLC je pristupačniji cijenom, no većinom se koristi za upravljanje jednostavnim postrojenjima. Modularni PLC je preko sabirnice povezan sa modulima koji na sebi sadrže ulaze i izlaze (Petruzella, F. D. 2017.).



Slika 1.4.5. Modul s dva digitalna izlaza i ulaza

(<https://www.helmholz.de/en/products/distributed-fieldbus-io-system/digital-inputoutput-modules/digital-mix-modules/dio-24-v-dc-500-ma>)

Princip rada PLC-a je vrlo jednostavan. Procesor odlučuje ovisno o programu napisanom od strane korisnika. Kako bi PLC koristio program, on komunicira s različitim uređajima zaduženih za nadzor i kontrolu. PLC radi tako što kontinuirano skenira programe i ponavlja

taj isti proces mnogo puta u sekundi. U trenutku paljenja PLC-a, provode se programske i sklopovske provjere sa ciljem otkrivanja grešaka. Ako nijedan problem nije detektiran, PLC počinje s ciklusom rada. On se sastoji od četiri koraka: čitanje ulaza, izvršavanje programa, dijagnostika i komunikacija, osvježavanje izlaza ([The Basic of PLC Operation. 2014.](#)).



Slika 1.4.6. Ciklus rada PLC-a

2. Rasprava

2.1. Karakteristike jezgri ARM Cortex

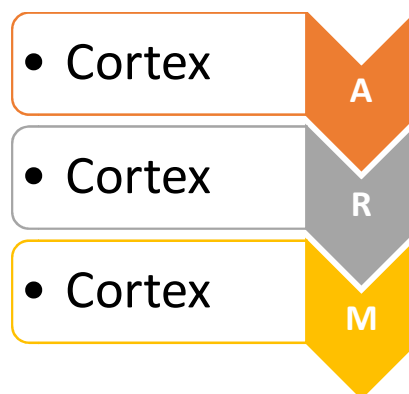
Seriya jezgri ARM Cortex obuhvaća vrlo širok raspon skalabilnih mogućnosti izvedbe nudeći dizajnerima velik izbor i priliku da koriste jezgru koja najbolje odgovara za njihovu primjenu bez prisiljavanja na jednoznačno rješenje. Cortex serija je podijeljena u tri kategorije:

- Cortex-A – aplikacijske procesorske jezgre za sustave koji zahtijevaju visoke performanse
- Cortex-R – jezgre visokih performansi za aplikacije u stvarnom vremenu
- Cortex-M – jezgre mikrokontrolera za širok raspon ugrađenih sustava

Cortex-A procesori nude niz rješenja za uređaje koji koriste operativni sustav kao Linux ili Android i koriste se u širokom spektru aplikacija, od jeftinih mobilnih telefona do pametnih telefona, tablet računala, set-top box uređaja i profesionalnih mrežnih oprema.

Cortex-R procesori su prilagođeni za aplikacije visokih performansi u stvarnom vremenu kao što su kontroleri tvrdog diska, kontroleri SSD-a (engl. Solid state drive), mrežna oprema, profesionalni pisači, a primjenjuje se i u automobilskoj industriji za zračne jastuke, sustave kočenja, upravljače motora.

Cortex-M serija je dizajnirana za vrlo konkurentno tržište mikrokontrolera. Ta serija je postala 32-bitni industrijski standard. Neki od mikrokontrolera unutar Cortex-M serije su optimizirani za energetska efikasnost, neki za visoke performanse, a neki su skrojeni za posebne tržišne segmente poput pametnog mjerenja ([Which ARM Cortex Core is Right for Your Application: A, R or M?. Silicon Labs.](#)).



Slika 2.1.1. Kategorije Cortex jezgri

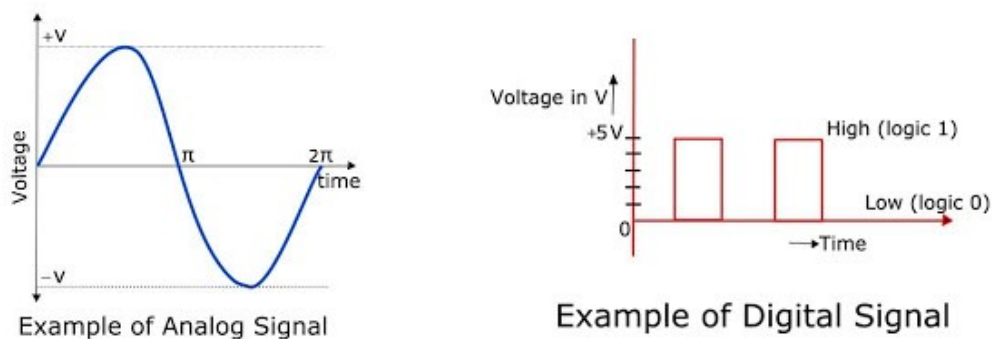
2.2. Pulsno-širinska modulacija

Pulsno-širinska modulacija (engl. Pulse-width modulation ili skraćeno PWM) je tehnika kontroliranja analognih krugova koristeći procesorske digitalne izlaze. Koristi se u mnoge svrhe, od mjerenja i komunikacije do kontrole snage i pretvorbe napona.

Analogni signal sadrži kontinuirano promjenjivu vrijednost, s beskonačnom razlučivošću i u vremenu i u veličini. Baterija od 9V primjer je analognog uređaja jer se njen izlazni napon mijenja tijekom vremena i može poprimiti bilo koju stvarno brojčanu vrijednost. Slično tome, količina struje koja se izlazi iz baterije nije ograničena na konačan skup mogućih vrijednosti. Analogni se signali razlikuju od digitalnih jer digitalni signali uvijek uzimaju vrijednosti iz skupa unaprijed definiranih mogućnosti, poput skupa $\{0V, 5V\}$.

Analogni signali se mogu koristiti za izravno upravljanje stvarima, poput kontrola glasnoće radija unutar automobila. U jednostavnom analognom radiju, tipka je spojena na promjenjivi otpornik. Okretanjem gumba otpor se povećava ili smanjuje, te se time struja kroz otpornik ponaša isto. To mijenja količinu struje koja pokreće zvučnike, povećavajući ili smanjujući glasnoću. Analogni krugovi mogu biti osjetljivi na buku, jako se zagrijavati i skuplji su za projektiranje nego digitalni.

Digitalnim kontroliranjem analognih krugova, troškovi sistema i potrošnja struje može biti drastično smanjena. Mnogi mikrokontroleri uključuju PWM kontrolere, čime se implementacija pojednostavljuje (Barr, M. *Introduction to Pulse Width Modulation (PWM)*. 2001.).



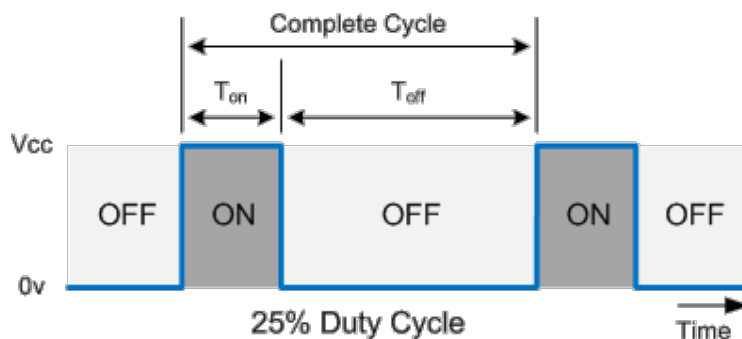
Slika 2.2.1. Analogni i digitalni signal

(<https://www.schoolelectronic.com/2012/01/difference-between-analog-and-digital.html>)

Tajmer je skup sličnih impulsa. U pulsno-širinskoj modulaciji se širina impulsa mijenja ovisno o situaciji. PWM sa 100% radnim ciklusom (engl. Duty cycle) je uvijek uključen

(logički 1), a PWM s 0% radnim ciklusom je uvijek isključen (logički 0). Radni ciklus predstavlja prosječnu vrijednost signala.

PWM signali se najčešće koriste u svrhu kontrole intenziteta LED (engl. Light emitting diode) diode, upravljanja servo motorima, kontroliranju brzine DC (engl. Direct current) motora i kontrole svjetline LCD-a (engl. Liquid crystal display) (White, E. 2011.).



Slika 2.2.2. PWM signal

(Using the PWM Expansion. Onion Docs.)

$$\text{radni ciklus} = \frac{T_{on}}{\text{Complete Cycle}}$$

$$\text{frekvencija} = \frac{1}{\text{Complete Cycle}}$$

$$\text{Complete Cycle} = T_{on} + T_{off}$$

(Using the PWM Expansion. Onion Docs.)

2.3. Pulsno-širinska modulacija koristeći ARM Cortex-M4

Primjer pulsno-širinske modulacije koristeći STM32F407VG Discovery razvojnu pločicu. STM32F407VG mikrokontroler pripada ARM Cortex-M4 obitelji mikrokontrolera koji je proizveden od ARM-ovog partnera STMicroelectronics.

2.3.1. Opis najznačajnijih registara

- CNT (counter)

Brojač koji broji svaki puls tajmera. Ovisno o postavljenom modu broji:

0-ARR (rastući mod)

ARR-0 (padajući mod)

- ARR (Auto-reload register)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARR[31:16] (depending on timers)															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

Slika 2.3.1.1. Bitovi 32-bitnog ARR registra

(RM0090 Reference manual)

ARR sadrži vrijednost do koje CNT broji ili od koje CNT broji, ovisno o postavljenom modu.

- PSC (Prescaler register)

16-bitni registar za skaliranje frekvencije izvora. Rezultat skaliranja je CK_CNT.

$$f_{CK_CNT} = \frac{f_{CK_PSC}}{Prescaler\ value + 1}$$

- CCR (Capture/Compare register)

Registar koji se postavlja za svaki potrebn kanal. Vrijednost ovog registra utječe na radni ciklus PWM signala.

$$RadniCiklus = \frac{CCR}{ARR} \times 100$$

- TIMx_CR1 (TIMx control register 1)

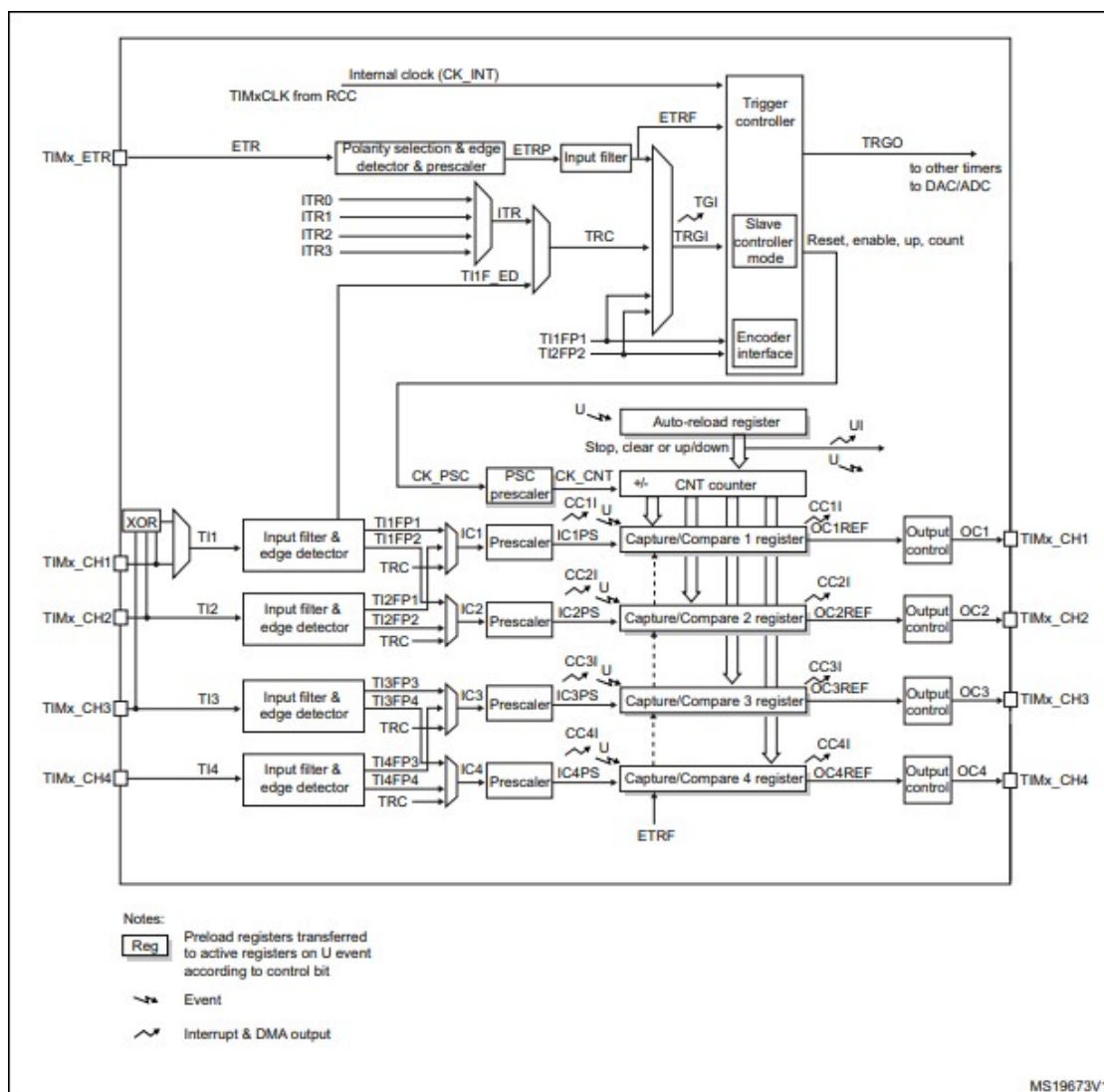
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CKD[1:0]	ARPE	CMS		DIR	OPM	URS	UDIS	CEN	
						rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

Slika 2.3.1.2. Bitovi 16-bitnog TIMx_CR1 registra

(RM0090 Reference manual)

Registar TIMx_CR1 se koristi za upravljanje tajmerom.

- CEN – bit za uključivanje tajmera
 - DIR – postavljanje moda brojanja: rastući/padajući
 - OPM – postavljanje moda jednog pulsa
 - CMS – postavljanje moda centralno poravnatog brojanja
- (RM0090 Reference manual)



Slika 2.3.1.3. Blok dijagram tajmera za opću namjenu unutar STM32F407 mikrokontrolera
(RM0090 Reference manual)

PWM mod	Mod brojanja	Logički 1	Logički 0
Mod 1	Rastući	$CNT < CCR$	$CNT \geq CCR$
Mod 1	Padajući	$CNT \leq CCR$	$CNT > CCR$

Mod 2	Rastući	CNT >= CCR	CNT < CCR
Mod 2	Padajući	CNT > CCR	CNT >= CCR

Tablica 2.3.1.1. Logičke vrijednosti u ovisnosti o odabranim modovima
(Yifeng, Z. 2017.)

PWM mod 1 – ako je brojač manji od referentnog signala, izlaz je logička jedinica, a u suprotnom nula.

PWM mod 2 – ako je brojač veći od referentnog signala, izlaz je logička jedinica, a u suprotnom nula (Yifeng, Z. 2017.).

2.3.2. Programski kod

Program koristi standardnu perifernu biblioteku STM32F4xx_StdPeriph.

```
#include "stm32f4xx.h"

int main(void)
{
    uint16_t ARR = 665;
    uint16_t CCR1 = 166;
    uint16_t PRS = (uint16_t) (SystemCoreClock / 1000000) - 1;

    GPIO_InitTypeDef GPIO_PortInitStructure;
    TIM_OCInitTypeDef TIM_ChannelInitStructure;
    TIM_TimeBaseInitTypeDef TIM_InitStructure;

    RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM3, ENABLE);
    RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOC, ENABLE);

    GPIO_PortInitStructure.GPIO_Pin = GPIO_Pin_6;
    GPIO_PortInitStructure.GPIO_Mode = GPIO_Mode_AF;
    GPIO_PortInitStructure.GPIO_Speed = GPIO_Speed_100MHz;
    GPIO_PortInitStructure.GPIO_OType = GPIO_OType_PP;
    GPIO_PortInitStructure.GPIO_PuPd = GPIO_PuPd_UP ;
    GPIO_Init(GPIOC, &GPIO_PortInitStructure);
    GPIO_PinAFConfig(GPIOC, GPIO_PinSource6, GPIO_AF_TIM3);

    TIM_InitStructure.TIM_Prescaler = PRS;
    TIM_InitStructure.TIM_CounterMode = TIM_CounterMode_Up;
    TIM_InitStructure.TIM_Period = ARR;
    TIM_InitStructure.TIM_ClockDivision = 0;
    TIM_TimeBaseInit(TIM3, &TIM_InitStructure);
```

```

TIM_ChannelInitStructure.TIM_OCMode = TIM_OCMode_PWM1;
TIM_ChannelInitStructure.TIM_OutputState = TIM_OutputState_Enable;
TIM_ChannelInitStructure.TIM_OCPolarity = TIM_OCPolarity_High;
TIM_ChannelInitStructure.TIM_Pulse = CCR1;
TIM_OC1Init(TIM3, &TIM_ChannelInitStructure);

TIM_OC1PreloadConfig(TIM3, TIM_OCPreload_Enable);

TIM_ARRPreloadConfig(TIM3, ENABLE);

TIM_Cmd(TIM3, ENABLE);

while (1){

}
}

```

2.3.3. Objašnjenje programskog koda

Deklariranje struktura za inicijalizaciju tajmera, kanala tajmera i GPIO porta.

```

GPIO_InitTypeDef GPIO_PortInitStructure;
TIM_OCInitTypeDef TIM_ChannelInitStructure;
TIM_TimeBaseInitTypeDef TIM_InitStructure;

```

Uključivanje sabirnice do GPIO porta C i tajmera TIM3.

```

RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM3, ENABLE);
RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOC, ENABLE);

```

Inicijalizacija GPIO porta C pina 6 kao pina alternativne funkcije i spajanje pina 6 s tajmerom TIM3.

```

GPIO_PortInitStructure.GPIO_Pin = GPIO_Pin_6;
GPIO_PortInitStructure.GPIO_Mode = GPIO_Mode_AF;
GPIO_PortInitStructure.GPIO_Speed = GPIO_Speed_100MHz;
GPIO_PortInitStructure.GPIO_OType = GPIO_OType_PP;
GPIO_PortInitStructure.GPIO_PuPd = GPIO_PuPd_UP ;
GPIO_Init(GPIOC, &GPIO_PortInitStructure);
GPIO_PinAFConfig(GPIOC, GPIO_PinSource6, GPIO_AF_TIM3);

```

Inicijalizacija TIM3 brojača. Postavljanje moda brojanja, vrijednosti PSC i ARR registara.

```

TIM_InitStructure.TIM_Prescaler = PRS;
TIM_InitStructure.TIM_CounterMode = TIM_CounterMode_Up;
TIM_InitStructure.TIM_Period = ARR;
TIM_InitStructure.TIM_ClockDivision = 0;
TIM_TimeBaseInit(TIM3, &TIM_InitStructure);

```

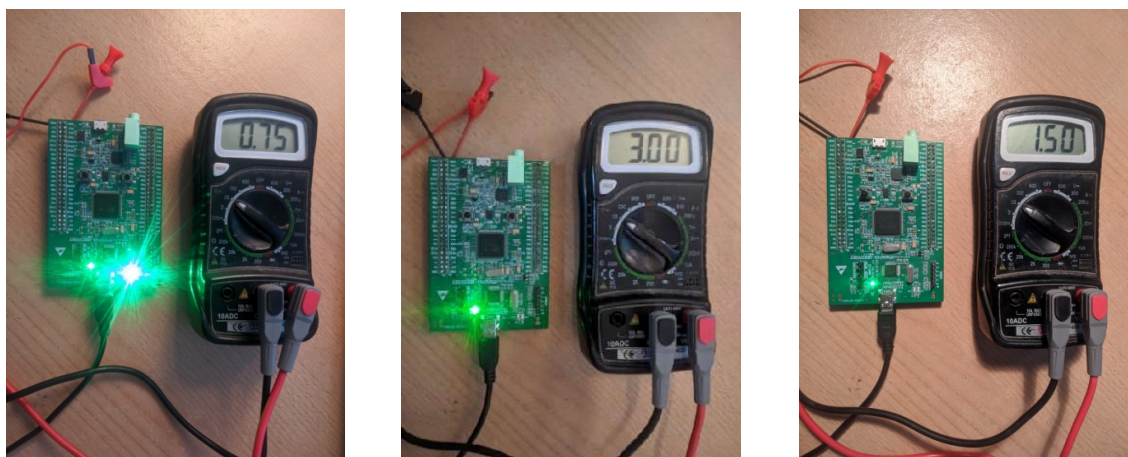
Inicijalizacija prvog kanala TIM3 brojača. Postavljanje vrijednosti CCR1 registra.

```
TIM_ChannelInitStructure.TIM_OCMode = TIM_OCMode_PWM1;  
TIM_ChannelInitStructure.TIM_OutputState = TIM_OutputState_Enable;  
TIM_ChannelInitStructure.TIM_OCPolarity = TIM_OCPolarity_High;  
TIM_ChannelInitStructure.TIM_Pulse = CCR1;  
TIM_OC1Init(TIM3, &TIM_ChannelInitStructure);
```

2.3.4. Primjer rada programa

Cilj primjera je pokazati korištenje PWM modulacije na mikrokontroleru.

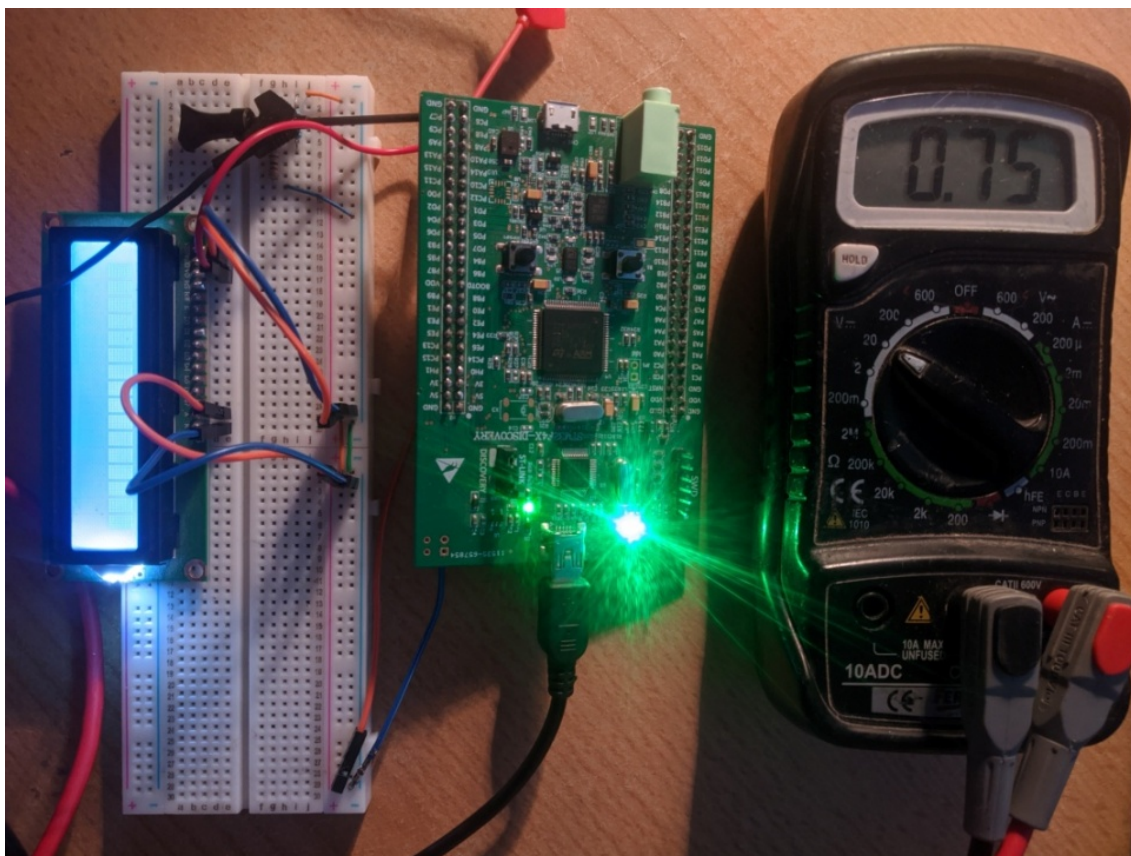
PWM signal izlazi s GPIO porta C pod pinom 6.



Slika 2.3.4.1. Efektivna voltaža PWM signala pri različitim radnim ciklusima

Pozicija na slici 2.3.1.	Lijevo	Sredina	Desno
Efektivni napon [V]	0,75	3,00	1,50
Radni ciklus	25%	100%	50%

Tablica 2.3.4.1. Efektivni napon i pripadajući radni ciklusi

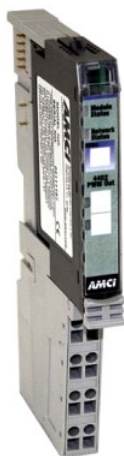


Slika 2.3.4.2. Upotreba modularanog signala za kontrolu kontrasta LCD-a s kontrolerom Hitachi HD44780

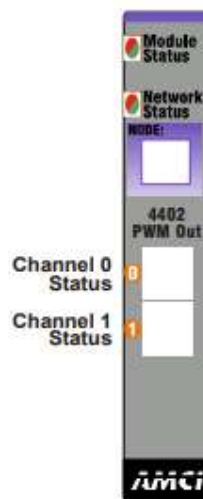
2.4. Izlazni modul 4402 PWM

4402 je najnoviji modul u rastućoj liniji proizvoda trvke AMCI. On nudi dva izlazna PWM kanala s promjenjivom frekvencijom i promjenjivim radnim ciklusom. Ovaj modul je programiran i kontroliran preko mrežnih podataka poslanih od glavnog kontrolera. Radni status modula se prenosi natrag do glavnog kontrolera koristeći 6 16-bitnih riječi.

Programiranje modula se ostvaruje koristeći 8 16-bitnih riječi. Jednom modulu je potrebno 15 ms za osvježavanje izlaza, dok taj broj iznosi 50 ms za blok od 5 modula (4402 PWM Output Module for POINT I/O. 2015.).



Slika 2.4.1. 4402 PWM izlazni modul



Slika 2.4.2. LED diode modula 4402

(4402 PWM Output Module for POINT I/O. 2015)

Na modulu se nalaze četiri LED diode koje označavaju statuse modula (slika 2.4.2.).

STANJE	OPIS
Zelena	Modul uredan
Crvena bljeskajuća	Greška programiranja
Crvena	Greška spremanja programa u memoriju

Tablica 2.4.1. Status modula LED

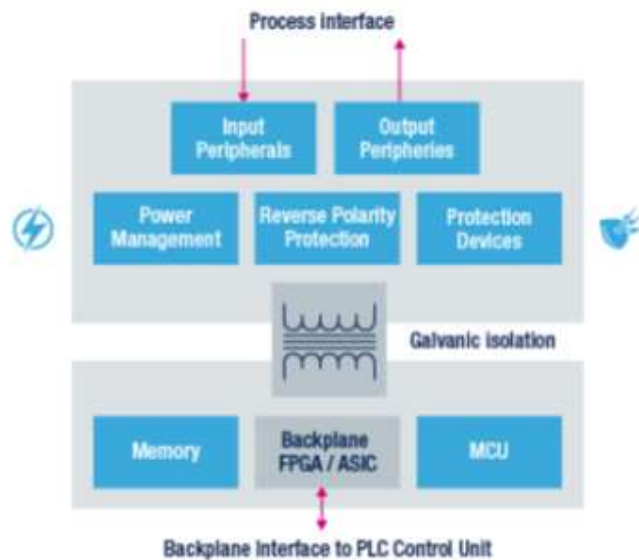
(4402 PWM Output Module for POINT I/O. 2015)

STANJE	OPIS
Ugašena	Komunikacija ne postoji
Zelena	Aktivna komunikacija
Crvena	Greška u komunikaciji

Tablica 2.4.2. Status mreže LED
(4402 PWM Output Module for POINT I/O. 2015)

STANJE	OPIS
Ugašena	Radni ciklus je postavljen na automatsku vrijednost
Narančasta	Radni ciklus je različit od automatske postavljene vrijednosti
Narančasta bljeskajuća	Radni ciklus je različit od automatske vrijednosti i „dihtering“ mod je uključen

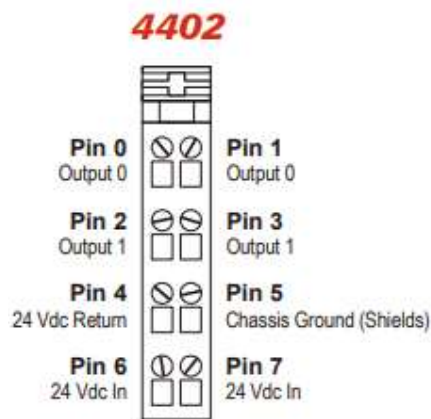
Tablica 2.4.3. Status kanala LED
(4402 PWM Output Module for POINT I/O. 2015)



Slika 2.5.2.1. Blok dijagram ulazno/izlaznog modula
(<https://www.st.com/en/applications/factory-automation/input-output-modules.html>)

2.4.1. Napajanje modula i spajanje trošila

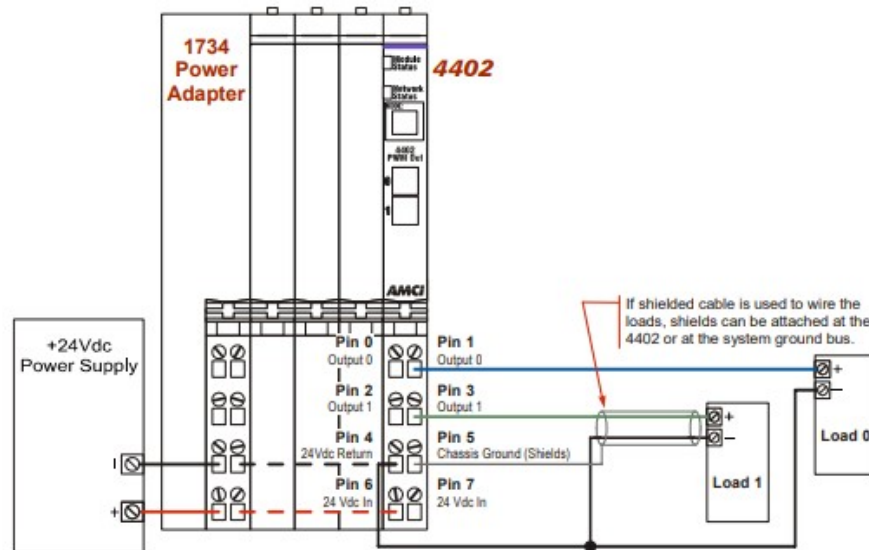
Modul 4402 se napaja spajanjem na 24 V DC napajanje preko POINT I/O sučelja. POINT I/O sučelje je razvijeno od strane tvrtke Allen-Bradley. Tako sučelje nudi jednostavan i modularni pristup napajanja modula. Dizajn POINT I/O sučelja dopušta da moduli budu zamjenjeni ili uklonjeni bez utjecaja na ožičenje drugih modula. POINT I/O uz napajanje modula služi i za komunikaciju između glavnog kontrolera na njega spojenih modula (Tierney, S. *Point I/O vs Flex I/O*. 2019.).



Slika 2.4.1.1. Ulazi i izlazi modula 4402

(4402 PWM Output Module for POINT I/O. 2015)

- Pin 0 i 1 su međusobno električki spojeni unutar modula tako da za spajanje trošila možemo iskoristiti bilo koji od tih izlaza
- Pin 2 i 3 su međusobno električki spojeni unutar modula tako da za spajanje trošila možemo iskoristiti bilo koji od tih izlaza
- Pin 6 i 7 su međusobno električki spojeni i spojeni sa POINT I/O naponskim konektorima
- Ukoliko se trošilo spaja kabelom koji sadži zaštitu od interferencija, ta zaštita se može spojiti na pin 5 (4402 PWM Output Module for POINT I/O. 2015.).



Slika 2.4.1.1. Ožičenje modula 4402
(4402 PWM Output Module for POINT I/O. 2015.)

2.4.2. Mod pulsno-širinska modulacija

Dva izlaza su dostupna na modulu 4402. Ovi izlazi imaju nezavisne PWM frekvencije i radne cikluse te se tako mogu i postaviti.

Frekvencija PWM signala je programabilna i može se postaviti od 1 Hz do 20 kHz. U postavljanju frekvencije PWM signala vrijedi obratiti pozornost na rezoluciju od 1 Hz.

Radni ciklus PWM signala se može postaviti od 0% do 100% ako je frekvencija kanala programirana ispod 10 kHz. Ako je frekvencija kanala veća od 10 kHz, radni ciklus se može postaviti na vrijednost 0%, 100% i od 5% do 95% (4402 PWM Output Module for POINT I/O. 2015.).

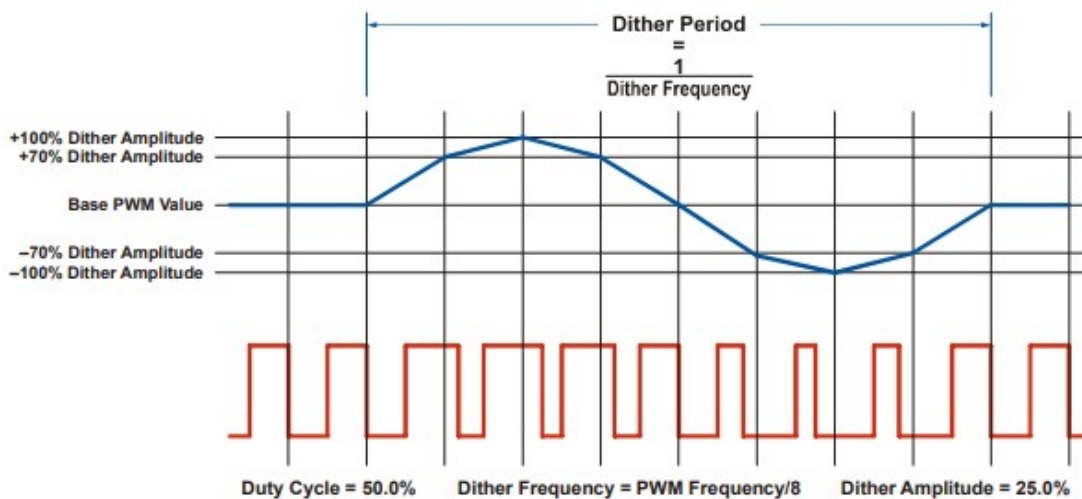
2.4.3. Mod Dither kontrola

Cilj ovog moda je proizvesti AC (engl. Alternating current) signal koristeći pulsno-širinsku modulaciju. To se postiže redovnim mijenjanjem radnog ciklusa. Ako se takav signal promatra preko osciloskopa, uz dodatnu filtraciju, trebao bi poprimiti oblik AC signala. Signal se kontrolira mijenjajući parametar frekvencije i parametar amplitude.

Modul 4402 sadrži dva parametra frekvencije, za svaki kanal po jedan parametar. S ovim parametrom postavljamo frekvenciju AC signala, koji želimo proizvesti. Ona može poprimiti vrijednost od 0 Hz do PWM frekvencija/4.

Unutar modula 4402 postoje i dva parametra za određivanje amplitude (po 1 za svaki kanal). Kao i kod radnog ciklusa, amplituda je cjelobrojna vrijednost koja predstavlja postotak od

programiranog vremenskog perioda. Rezolucija parametra amplitude iznosi 0,1%. Parametri mogu poprimiti vrijednost od 0 do {radni ciklus/2}. Ako vrijednost amplitude iznosi 0, ovaj mod je isključen za taj kanal (4402 PWM Output Module for POINT I/O. 2015.).



Slika 2.4.3.1. Reprezentacija AC signala pulsno-širinskom modulacijom (4402 PWM Output Module for POINT I/O. 2015.)

$$\text{radni ciklus}(t) = \text{Radni ciklus} + \text{Dither amplituda} \times \sin(\omega t)$$

Dither amplituda = određena vrijednost amplitude za odabrani kanal

Radni ciklus = određena vrijednost radnog ciklusa za odabrani kanal

t = vrijeme u sekundama

ω = Dither frekvencija

2.4.4. Početno stanje uređaja

Modul 4402 koristi flash memoriju. Ona sprema i pohranjuje podatke iako je uređaj isključen. Unutar memorije zapisane su predodređene vrijednosti registara (4402 PWM Output Module for POINT I/O. 2015.).

Parameter	Setting
Output State on Net Loss	Maintain Current Output
CH0 PWM Frequency	250 Hz
CH0 Duty Cycle	10.0% (Integer value of 100)
CH0 Dither Frequency	0 Hz (Disabled state)
CH0 Dither Amplitude	0.0% (Disabled state)
CH0 RT Update Enable	Disabled
CH1 PWM Frequency	250 Hz
CH1 Duty Cycle	10.0% (Integer value of 100)
CH1 Dither Frequency	0 Hz (Disabled state)
CH1 Dither Amplitude	0.0% (Disabled state)
CH1 RT Update Enable	Disabled

Slika 2.4.4.1. Početna stanja parametara modula 4402
(4402 PWM Output Module for POINT I/O. 2015.)

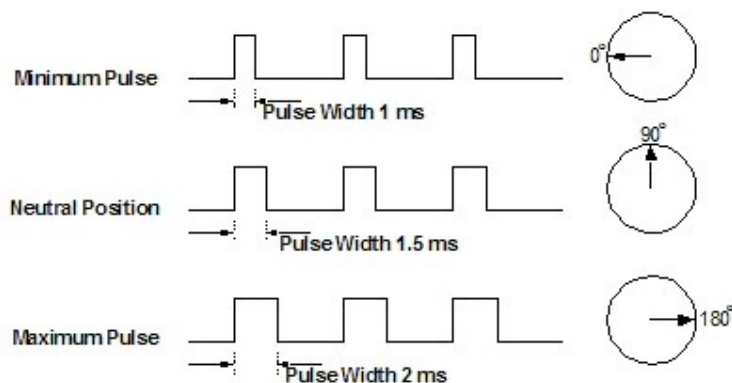
2.4.5. Primjene modula

PWM modul svoju primjenu najčešće nalazi u kontroli izlazne voltaže te time služi za kontrolu intenziteta LED diode, jačine zvuka alarma ili zvona, brzine okretanja DC motora. Korištenje modula u te svrhe je intuitivno. Neizostavni razlog korištenja PWM modula je kontrola servo motora. Servo motori se primjenjuju u industrijskoj proizvodnji, robotici i proizvodnji na pokretnoj traci. Za razliku od klasičnog DC motora, servo motori su namjenjeni za preciznu kontrolu rotacije u oba smijera (How Servo Motors Work.).



Slika 2.4.5.1. 24 V DC servo motor snage 400W
(<https://www.ato.com/400w-dc-servo-motor>)

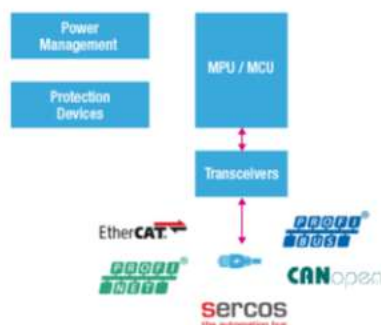
Servo uređaji se kontroliraju slanjem PWM impulsa preko upravljačke žice servo uređaja. Neutralni položaj servo uređaja je taj u kojem servo ima potpuno jednaku količinu potencijalne rotacije u smjeru kazaljke na satu kao i u suprotnom smjeru. Različiti servo uređaji su različito ograničeni u rotaciji, no svi imaju neutralni položaj koji se poprimi slanjem signala pulsa širine 1,5 ms. Servo očekuje puls svakih 20 ms, a duljina impulsa odredit će dokle se okreće motor. Kada se pošalje impuls koji je manji od 1,5 ms, servo motor se zakreće u smjeru suprotnom od kazaljke na satu. Kad je impuls širi od 1,5 ms događa se suprotno (**Servo FAQs**).



Slika 2.4.5.2. Kontrola servo motor
(**Servo FAQs**)

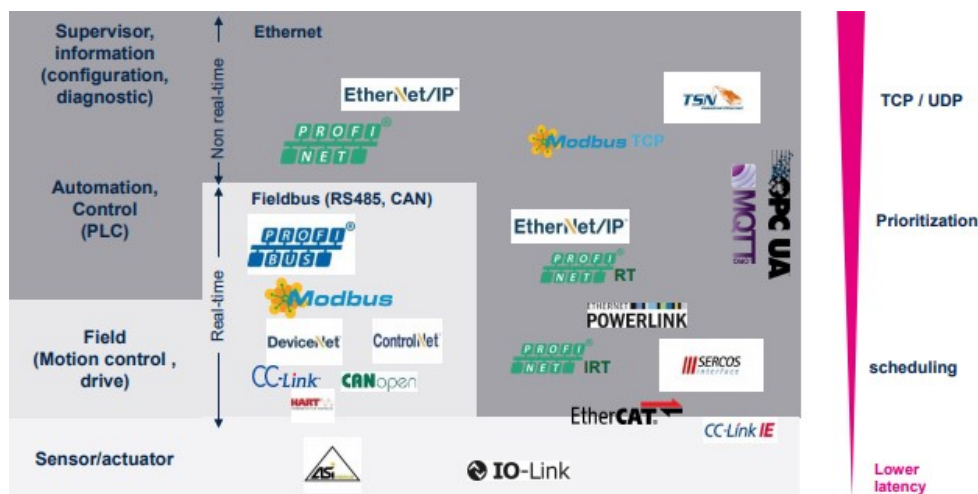
2.5. Fieldbus i industrijski ethernet

U tvorničkoj automatizaciji presudna je povezanost podataka posebno kada koncept Industrije 4.0 proširuje svoj opseg. Brojni standardi se koriste za podršku posebno razvijenih sabirnica uključujući CAN (Modbus i CANopen), RS-485 (PROFIBUS, CC-Link i Modbus), te industrijski Ethernet.

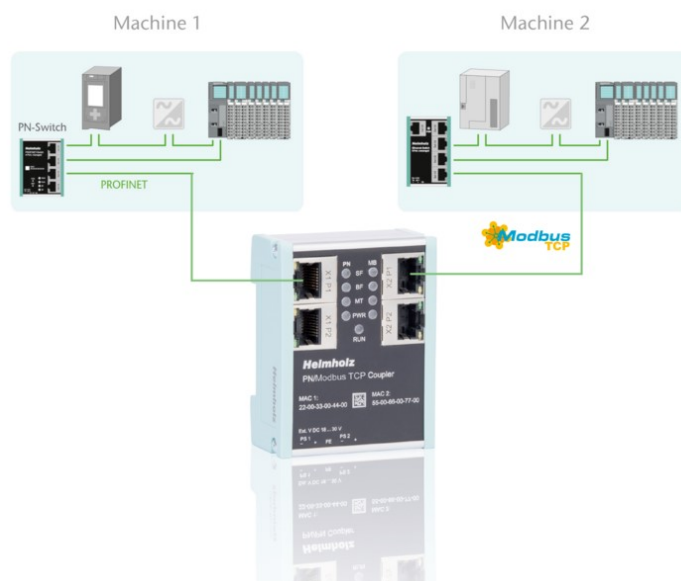


Slika 2.5.2.2. Blok dijagram uređaja Fieldbus i industrijskog Etherneteta
(**Fieldbus and Industrial Ethernet. STMicroelectronics**.)

Tvrtka STMicroelectronics nudi široku paletu 32-bitnih mikrokontrolera baziranih na ARM arhitekturi s velikim brojem perifernih uređaja koji podržavaju implementaciju širokog spektra protokola, uključujući UART (engl. Universal asynchronous receiver-transmitter), CAN, CAN-FD i industrijskog Ethernet (Fieldbus and Industrial Ethernet. STMicroelectronics.).



Slika 2.5.2.3. Podržani protokoli unutar ekosustava STM32
(Embedded software solutions ST, 3rd parties and open source)



Slika 2.5.2.4. Helmholz Profinet-ModbusTCP coupler
(<https://www.helmholz.de/en/products/fieldbus-applications/modbus/couplersgateways/pnmodbustcp-coupler>)

3. Zaključak

Unutar razdoblja od 1970. do 1990. godine svjedoci smo revolucionarnog vala poboljšanja produktivnosti u proizvodnji. Nove tehnologije, uključujući osobna računala, numerički upravljane strojeve, programabilne logičke kontrolere i računalno potpomognutu proizvodnju, u kombinaciji s novim idejama za poboljšanje procesa poput TQM (engl. Total Quality Management), Just-In -Time i Six Sigma postigli su sjajne nove razine produktivnosti i učinkovitosti u mnogim proizvodnim industrijama. U ovom desetljeću vidimo sličnu konvergenciju tehnologija i inicijativa za poboljšanje procesa s potencijalom da se radikalno poboljša način na koji proizvođači povezuju procese i sustave u cijelom lancu vrijednosti proizvoda kako bi kupcima pružili više mogućnosti i novih modela usluga (Conrad, Leiva. *On the Journey to a Smart Manufacturing Revolution*. 2015.).

Industrijska automatizacija je specifična po zatvorenoj arhitekturi. Sve do sada većina izrađenih uređaja nisu koristila zajedničke standarde. Time se znatno otežao daljnji napredak tehnologije i procesa. Tržište sve do sada nije uvidjelo taj problem jer nije postojao razlog za povezivanjem više tehnologija. No danas, se stvorila potreba za zajedničkim standardima koji će pomoći u povezivanju svih uređaja s ciljem optimizacije procesa i smanjenja troškova. Time se otvorila prilika za stvaranje novih uređaja koji će implementirati moderne metode i tako jednoga dana doći do cilja pametne tvornice. Takva tvornica će moći logički i proizvodno funkcionirati bez ljudske intervencije. Tvrtka ARM je prepoznala novu priliku tržišta te je prilagodila seriju proizvoda za ovaj inovativni pokret. Procesori serije za automatizaciju su dizajnirani s posebnim naglaskom na performanse, vrijeme odaziva i sigurnost. ARM će i dalje doprinosom u razvoju arhitekture procesora pridonijeti razvoju industrije 4.0.

Application of ARM architecture in automation

Summary

This bachelor final paper on the application of ARM architecture in automation will explain basic characteristics of ARM architecture and the basic concepts of industrial automation. The historical progress of these two, perhaps seemingly unrelated industries, will also be considered. The basic functionality of the existing PLC module is explained, as an insight into the current market demands. Part of the functionality of selected PLC module will be developed within an example. The application of the ARM architecture in automation will be supported by existing solutions from STMicroelectronics, which has adapted to the new growing market.

Keywords: ARM, automation, pulse-width modulation, programmable logic controller

Literatura

1. Kovač, M. (2015). *Arhitektura računala*. Zagreb: Fakultet elektrotehnike i računarstva.
2. Petruzella, F. D. (2017). *Programmable Logic Controllers (Fifth edition.)*. New York: McGraw-Hill Education.
3. Yiu, J. (2013). *The Definitive Guide to ARM® Cortex®-M3 and Cortex-M4 processors (Third edition.)*. Kidlington, Oxford: Newnes.
4. Walshe, B. *A Brief History of Arm: Part 1*. (2015). URL: <https://community.arm.com/developer/ip-products/processors/b/processors-ip-blog/posts/a-brief-history-of-arm-part-1> (2020-10-10)
5. *The Basic of PLC Operation*. (2014). URL: <https://www.techtransfer.com/blog/basics-plc-operation> (2020-10-10)
6. *Which ARM Cortex Core is Right for Your Application: A, R or M?*. Silicon Labs. URL: <https://www.silabs.com/documents/public/white-papers/Which-ARM-Cortex-Core-Is-Right-for-Your-Application.pdf> (2020-10-10)
7. White, E. (2011). *Making Embedded Systems: Design Patterns for Great Software (First edition.)*. O'Reilly Media.
8. Barr, M. *Introduction to Pulse Width Modulation (PWM)*. (2001). URL: <https://barrgroup.com/Embedded-Systems/How-To/PWM-Pulse-Width-Modulation> (2020-10-11)
9. *Using the PWM Expansion*. URL: <https://docs.onion.io/omega2-docs/using-pwm-expansion.html> (2020-10-11)
10. *RM0090 Reference manual (Rev. 18)*. STMicroelectronics. URL: https://www.st.com/resource/en/reference_manual/dm00031020-stm32f405-415-stm32f407-417-stm32f427-437-and-stm32f429-439-advanced-arm-based-32-bit-mcus-stmicroelectronics.pdf (2020-10-12)
11. Yifeng, Z. (2017). *Embedded Systems with ARM Cortex-M Microcontrollers in Assembly Language and C (Third edition.)*. E-man Press LLC.
12. *What is Industrial Automation?*. (2013). URL: <https://www.surecontrols.com/what-is-industrial-automation> (2020-10-12)
13. *4402 PWM Output Module for POINT I/O*. (2015). AMCI. URL: https://www.amci.com/files/4114/6538/5039/4402_manual_04030.pdf (2020-10-14)

14. Tierney, S. *Point I/O vs Flex I/O*. (2019). URL: <https://theautomationblog.com/point-i-o-vs-flex-i-o> (2020-10-14)
15. *How Servo Motors Work*. URL: <https://www.jameco.com/jameco/workshop/howitworks/how-servo-motors-work.html> (2020-10-15)
16. *Servo FAQs*. URL: <https://www.servocity.com/servo-faqs> (2020-10-15)
17. Conrad, Leiva. *On the Journey to a Smart Manufacturing Revolution*. (2015). URL: <https://www.industryweek.com/technology-and-iiot/systems-integration/article/21967056/on-the-journey-to-a-smart-manufacturing-revolution?page=2> (2020-10-15)
18. *Fieldbus and Industrial Ethernet*. URL: <https://www.st.com/en/applications/connectivity/fieldbus-and-industrial-ethernet.html> (2020-10-15)
19. *Embedded software solutions ST, 3rd parties and open source*. URL: https://www.st.com/content/ccc/resource/sales_and_marketing/presentation/product_presentation/30/8f/78/6a/b9/a4/45/ab/stm32-stm8_embedded_software_solutions.pdf/files/stm32-stm8_embedded_software_solutions.pdf/jcr:content/translations/en.stm32-stm8_embedded_software_solutions.pdf (2020-10-15)