

# Korištenje paradigme "digitalnih blizanaca" u razvoju proizvoda

---

Vlah, Daria

Master's thesis / Diplomski rad

2018

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

*Permanent link / Trajna poveznica:* <https://urn.nsk.hr/urn:nbn:hr:235:870979>

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

*Download date / Datum preuzimanja:* **2024-07-14**

*Repository / Repozitorij:*

[Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# **DIPLOMSKI RAD**

**Daria Vlah**

Zagreb, 2018.

SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# DIPLOMSKI RAD

Mentor:

Prof. dr. sc. Mario Štorga, dipl. ing.

Student:

Daria Vlah

Zagreb, 2018.

Izjavljujem da sam ovaj rad izradio samostalno koristeći znanja stečena tijekom studija i navedenu literaturu.

Zahvaljujem se mentoru prof. dr. sc. Mariju Štorgi na zadavanju ovako izazovnog zadatka te za strpljenje, podijeljeno znanje i dane savjete tokom izrade rada. Također se zahvaljujem kolegi Nikoli Horvatu za pomoć oko tehničke izvedbe zadatka te Alessandru Toliu, bez čijih savjeta u vezi softverske podrške ovaj rad možda ne bi bio uspješno realiziran. Posebno se zahvaljujem kolegi Matiji Kurtoiću i svojoj obitelji za strpljenje i podršku pruženu tokom izrade ovog rada, ali i tokom svih godina mog studiranja.

Daria Vlah



SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE



Središnje povjerenstvo za završne i diplomske ispite  
Povjerenstvo za diplomske ispite studija strojarstva za smjerove:  
procesno-energetski, konstrukcijski, brodostrojarški i inženjersko modeliranje i računalne simulacije

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa:	
Ur.broj:	

## DIPLOMSKI ZADATAK

Student: **Daria Vlah**

Mat. br.: **0035190746**

Naslov rada na hrvatskom jeziku: **Korištenje paradigme „digitalnih blizanaca“ u razvoju proizvoda**

Naslov rada na engleskom jeziku: **The utilisation of the „digital twin“ paradigm in product development**

Opis zadatka:

„Digitalni blizanac“ može se definirati kao digitalni zapis ponašajnih karakteristika tehničkog sustava koji nastaje tijekom korištenja sustava te pomaže pri optimiranju njegovih radnih performansi. Jedna od mogućih primjena implementacija paradigme „digitalnih blizanaca“ temelji se na prikupljanju podataka o radu tehničkog sustava i njegovoj radnoj okolini u realnom vremenu, te integraciji s CAD okolinom u kojoj se prikupljeni podaci analiziraju putem računalnih simulacija rada tehničkog sustava. Na taj se način paradigma „digitalnih blizanaca“ može iskoristiti za dobivanje ključnih spoznaja potrebnih pri donošenju odluka u razvoju proizvoda.

U okviru rada potrebno je:

- Istražiti značajke paradigme „digitalnih blizanaca“ i mogućnosti primjene u razvojnom procesu, te za kasnije faze životnog ciklusa tehničkih sustava.
- Realizirati tehnologiju „digitalnih blizanaca“ korištenjem Arduino elektroničke platforme za prikupljanje podataka o radu tehničkih sustava, te 3DEXPERIENCE radne okoline za modeliranje, simulaciju i analizu rada tehničkih sustava.
- Na konkretnom primjeru pokazati mogućnosti prikupljanja i pohrane podataka o radu tehničkog sustava u stvarnom vremenu, te njihove analize korištenjem metoda za obradu velikih skupova podataka.

Opseg implementacije dogovoriti tijekom izrade rada.

U radu navesti korištenu literaturu i eventualno dobivenu pomoć.

Zadatak zadan:

3. svibnja 2018.

Zadatak zadao:

Prof. dr. sc. Mario Štorga

Rok predaje rada:

5. srpnja 2018.

Predviđeni datumi obrane:

11., 12. i 13. srpnja 2018.

Predsjednica Povjerenstva:

Prof. dr. sc. Tanja Jurčević Lulić

## SADRŽAJ

SADRŽAJ .....	1
POPIS SLIKA .....	3
POPIS OZNAKA .....	5
SAŽETAK.....	6
SUMMARY .....	7
1. UVOD.....	8
1.1. Industrija 4.0 .....	8
1.1.1. Četvrta industrijska revolucija .....	8
1.1.2. Tehnologije/paradigme Industrije 4.0.....	9
1.1.3. Karakteristike Industrije 4.0.....	11
1.1.4. Posljedice za industriju .....	12
2. Digitalni blizanci .....	14
2.1. Definicija.....	14
2.2. Povijest razvoja paradigme digitalnih blizanaca.....	15
2.3. Digitalni blizanci u razvojnom procesu .....	16
2.4. Komponente digitalno-fizičkih sustava .....	17
2.5. Koraci za kreiranje digitalnih blizanaca.....	19
2.6. Primjeri primjene digitalnih blizanaca .....	20
2.6.1. Digitalni blizanci zrakoplova .....	21
2.6.2. Digitalni blizanac brdskog bicikla .....	22
3. Primjer razvoja digitalnih blizanaca .....	23
3.1. Funkcionalnost uređaja .....	24
3.2. Prototip.....	24
3.3. Komponente uređaja .....	25
3.3.1. Nosiva konstrukcija .....	26
3.3.2. Upravljačka jedinica .....	29
3.3.3. Senzori .....	30

---

3.3.4. Aktuatori .....	31
3.3.5. Solarni panel .....	32
3.4. Upravljanje .....	33
4. Modeliranje digitalnog blizanca .....	34
4.1. Modeliranje digitalno-fizičkog sustava .....	34
4.2. 3DExperience .....	35
4.2.1. CATIA .....	35
4.2.2. RFLP .....	36
4.2.3. DYMOLA .....	37
4.3. Povezivanje uređaja i virtualnog modela na 3DExperience platformi .....	38
4.3.1. Logička shema uređaja .....	38
4.3.2. Logička jedinica Arduino mikrokontrolera .....	39
4.3.3. Ulazne logičke jedinice .....	40
4.3.4. Logičke jedinice za pretvorbu varijabli .....	42
4.3.5. Logička jedinica 3D modela mehanizma .....	43
4.4. Serijska komunikacija .....	47
4.5. Pokretanje uređaja i digitalnog blizanca .....	48
5. Prikupljanje i analiza podataka .....	50
5.1. Veliki skupovi podataka u Industriji 4.0 .....	50
5.2. Analiza podataka na izvedenom primjeru .....	52
6. ZAKLJUČAK .....	56
LITERATURA .....	58
PRILOZI .....	60

## POPIS SLIKA

Slika 1.	Prikaz četiri industrijske revolucije [1] .....	9
Slika 2.	Industrija 4.0 [1] .....	10
Slika 3	Shematski prikaz modela digitalnog blizanca [4] .....	14
Slika 4	Digitalno fizički sustav proizvodne linije [6].....	18
Slika 5	Koraci oblikovanja arhitekture digitalnog blizanca [6].....	20
Slika 6	Prikaz mjerenih podataka na motoru zrakoplova u digitalnog okolini [10].....	21
Slika 7	Digitalni blizanac brdskog bicikla [11] .....	22
Slika 8	Uređaj korišten za izradu digitalnog blizanca .....	23
Slika 9	Rana faza fizičkog prototipa.....	25
Slika 10	3D printana inačica uređaja .....	26
Slika 11	Komponente nosive konstrukcije .....	27
Slika 12	3D model uređaja .....	28
Slika 13	Pogled na stražnji dio uređaja – nosiva konstrukcija i ožičenje.....	29
Slika 14	Arduino mikrokontroler (desno) i eksperimentalna pločica (lijevo).....	30
Slika 15	Fotootpornici postavljeni na uređaj .....	30
Slika 16	Aktuatori postavljeni na uređaj .....	31
Slika 17	Uređaj sa postavljenim solarnim panelom i digitalnim voltmetrom .....	33
Slika 18	Apstraktni model digitalno-fizičkog sustava.....	34
Slika 19	Catia V6 sučelje unutar 3DExperience platforme sa ekspaniranim stablom značajki.....	36
Slika 20	RFLP radna okolina.....	37
Slika 21	DYMOLA radna okolina.....	38
Slika 22	RFLP logička radna okolina.....	38
Slika 23	Logička shema realizirang uređaja.....	39
Slika 24	Četiri logičke komponente vezane uz jedan fizički konektor .....	40
Slika 25	Ulazna logička jedinica .....	40
Slika 26	Povezivanje ulaznih logičkih jedinica i jedinice logičkog Arduina .....	41
Slika 27	Prikaz trenutnih vrijednosti očitavanih parametara.....	41
Slika 28	Shematski prikaz logičke jedinice za pretvorbu i normizaciju signala Converter 1 .....	42



---

Slika 29	Shematski prikaz logičke jedinice za pretvorbu, normizaciju i modifikaciju područja očitavanja signala Converter 2 .....	43
Slika 30	Povezivanje fizičkog i logičkog modela naredbom Implement relations .....	44
Slika 31	Mehanizam sa pripadajućim stablom značajki.....	45
Slika 32	Povezivanje pretvorbenih logičkih jedinica i logičke jedinice mehanizma .....	45
Slika 33	Kreiranje logičke jedinice mehanizma postupkom mapiranja .....	46
Slika 34	Logička reprezentacija mehanizma .....	47
Slika 35	Definiranje ulaznog USB porta .....	48
Slika 36	Definiranje načina izvođenja simulacije uz uključenu opciju vremenske zadržke	49
Slika 37	Simulacija rada uređaja sa prikazom mjerenih podataka .....	49
Slika 38	Shematski prikaz prikupljanja i strukturiranja podataka [21] .....	51
Slika 39	Prikaz pojednostavljenog modela neuronske mreže [23] .....	52
Slika 40	Različiti položaji uređaja naspram izvora svjetlosti.....	53
Slika 41	Digitalni blizanac uređaja u trenutku kad je usmjeren od izvora svjetlosti.....	54
Slika 42	Digitalni blizanac uređaja u trenutku kad je usmjeren prema izvoru svjetlosti ....	54

**POPIS OZNAKA**

<b>Oznaka</b>	<b>Jedinica</b>	<b>Opis</b>
$P_{el}$	W	Izlazna električna snaga
$P_{sol}$	W	Snaga zračenja (najčešće Sunčevog)
$U$	V	Efektivna vrijednost izlaznog napona
$I$	A	Efektivna vrijednost izlazne struje
$E$	W/m <sup>2</sup>	Specifična snaga zračenja
$A$	m <sup>2</sup>	Površina

## SAŽETAK

U ovom radu prikazan je postupak izrade digitalnog blizanca u sklopu 3DExperience platforme na primjeru pametnog uređaja opremljenog sensorima. Prije same izvedbe uređaja u uvodu je dan pregled pojmova vezanih uz Industriju 4.0, njene glavne karakteristike i tehnologije te je objašnjeno značenje digitalno-fizičkih sustava. U nastavku je koncept digitalnog blizanca definiran kao digitalni zapis ponašanja elementa i dinamike sustava koji nastaje tokom rada sustava te pomaže pri optimizaciji njegovih radnih performansi. Da bi se ponašanje proizvoda moglo reflektirati na njihovu virtualnu inačicu u realnom vremenu, oni moraju biti opremljeni odgovarajućim sensorima, a digitalna okolina mora omogućavati prikupljanje i pretvaranje ulaznih podataka u oblik pogodan za daljnju upotrebu. U radu su navedeni koraci za realizaciju digitalnih blizanaca koji su detaljno razrađeni kroz primjer pokretne solarne platforme koja se prilagođava okolini na način da se nastoji maksimizirati učinkovitost prikupljanja energije. Pojednostavljena verzija uređaja je izrađena tako da su ostvarene glavne funkcionalnosti digitalno-fizičkih sustava te koristi Arduino mikrokontroler za upravljanje, prikupljanje i prijenos podataka o radu uređaja i stanju okoline. Arduino povezuje fizičku i digitalnu okolinu unutar 3DExperience platforme preko odgovarajuće logičke jedinice koja pretvara ulazne signale u oblik koji omogućuje pokretanje 3D modela. Time je kreirana virtualna kopija fizičkog proizvoda. Za daljnji razvoj ovog proizvoda potrebno je skupljati podatke o performansama solarnog panela te ih analizirati u kombinaciji sa podacima iz drugih izvora kako bi se rezultati analize mogli koristiti u prediktivne svrhe i za optimizaciju njegovih radnih performansi.

Ključne riječi: Industrija 4.0, digitalno-fizički sustavi, digitalni blizanci, pametni proizvodi, senzori, Arduino, 3DExperience, 3D model, razvoj proizvoda

## **SUMMARY**

This thesis presents the process of creating a digital twin of a smart product equipped with sensors using the 3DExperience platform. Before the physical realization of a product an introduction to Industry 4.0, its main characteristics and technologies was given along with the description of cyber-physical systems. The concept of digital twin was defined as a digital representation of system's behavior and dynamics which is created during runtime, which is later on used for the optimization of the product's performance. To create a real time virtual representation of a product's behavior, it needs to be equipped with a number of sensors, and the digital environment has to have the ability to enable data collection and transformation so that they can be used for further analyses. Steps for the creation of a digital twin are explained in details with a use case of a solar platform that can adapt to its surrounding in order to maximize its energy production rate. The simplified version of the device was designed, that has all of the main functionalities of a cyber-physical system which uses Arduino microcontroller for control and collection of the product's performance and working conditions data. Arduino connects physical and digital environment through 3DExperience platform using a matching logical unit that translates input signals into a form that enables the motion of a 3D model, which finally represents a virtual twin of a physical product. For further development of this product performance data of the solar panel need to be analyzed together with the data from other sources so that the results can be used for predictive purposes and performance optimization.

Key words: Industry 4.0, cyber-physical systems, digital twin, smart products, sensors, Arduino, 3DExperience, 3D model, product development

## 1. UVOD

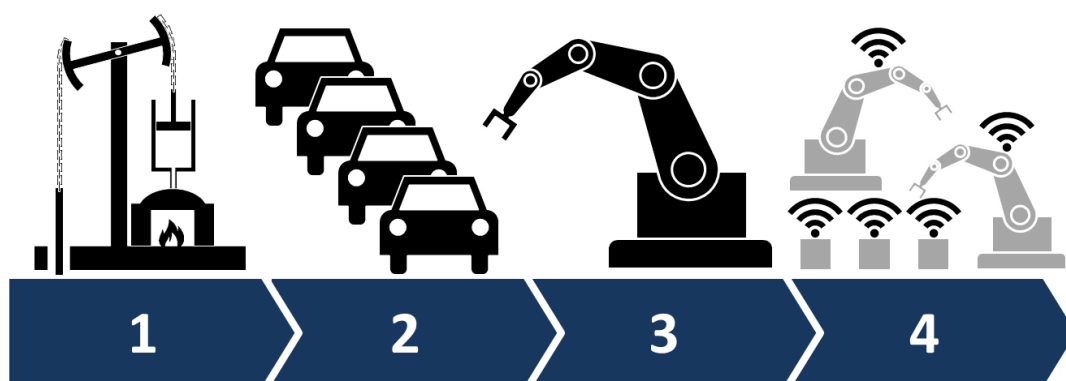
U okvirima neprestano rastuće Industrije 4.0 pojam digitalnog blizanca postaje sve rašireniji. U sklopu ovog rada biti će prikazane karakteristike Industrije 4.0 te će se razmatrati uvjeti potrebni za realizaciju digitalnog blizanca, koji će biti potkrijepljeni pojednostavljenim eksperimentalnim primjerom.

### 1.1. Industrija 4.0

U posljednjih nekoliko godina sve češće se susrećemo s pojmom Industrije 4.0 koja se temelji na velikim promjenama koje već sad znatno utječu na sve segmente industrijskog poslovanja, od poslovnih modela i procesa, načina proizvodnje, radnih mjesta, pa sve do načina komunikacije unutar i među kompanijama. U nastavku će pobliže biti opisane karakteristike Industrije 4.0 kao 4. industrijske revolucije, glavne tehnologije koje se vežu uz Industriju 4.0 kao što su Internet objekata (eng. *Internet of Things*), digitalno-fizički sustavi (eng. *Cyber-Physical Systems*) i analiza velikih kupova podataka (eng. *Big Data*), te će biti navedene osnovne smjernice implementacije Industrije 4.0 u tvrtke i njihova najčešća primjena.

#### 1.1.1. Četvrta industrijska revolucija

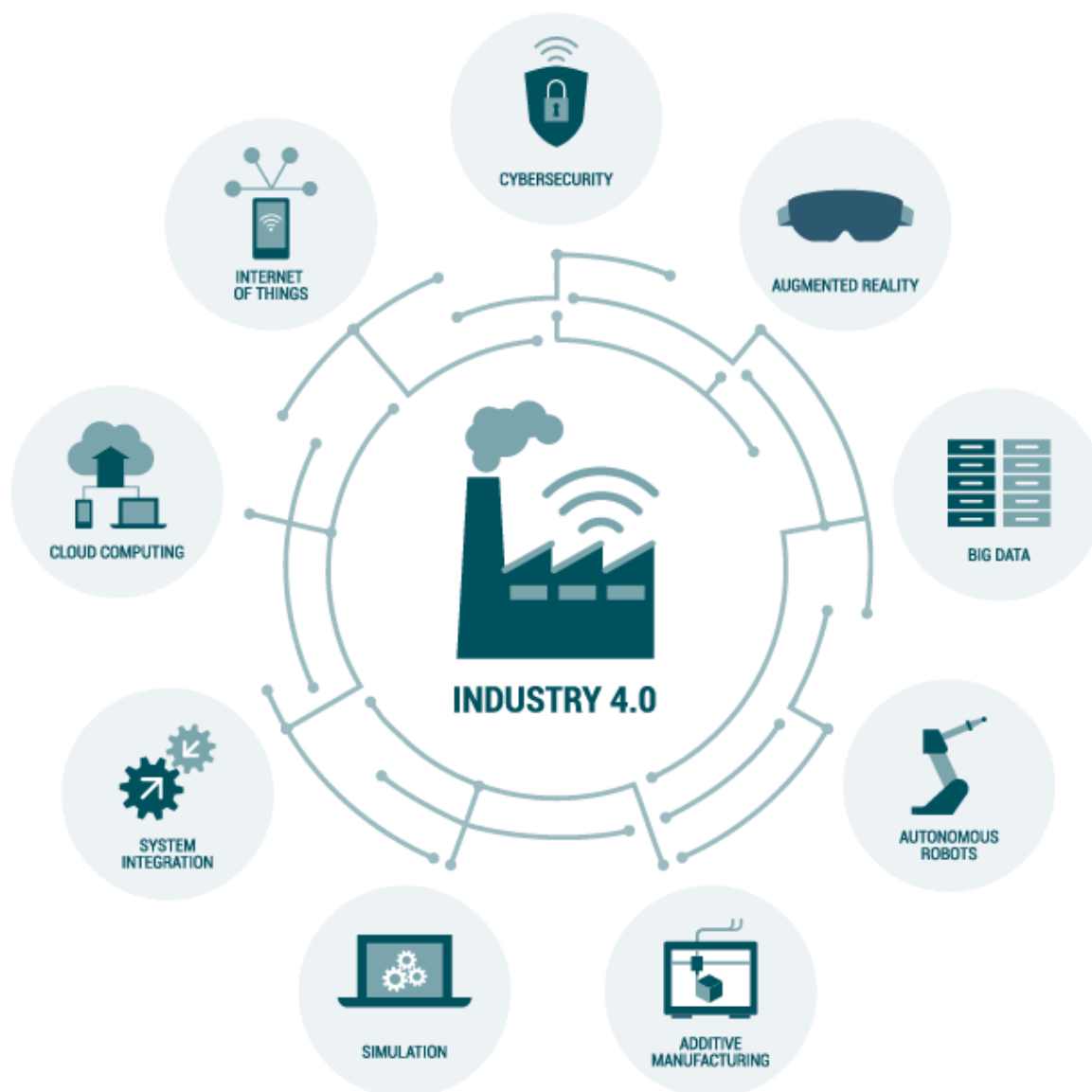
Prva industrijska revolucija započela je izumom parnog stroja, a obilježava je mehanizacija radnog procesa. Drugu industrijsku revoluciju pokrenula je pojava rada na proizvodnoj liniji, te otkriće električne energije uz koje je započela masovna proizvodnja dobara. Treća industrijska revolucija veže se uz pojavu računala i razvoj elektronike te ju je obilježila automatizacija proizvodnje. Iako je neki smatraju tek nastavkom treće industrijske revolucije, četvrtu industrijsku revoluciju obilježava međusobno povezivanje digitalno-fizičkih sustava i ubrzani razvoj umjetne inteligencije čime se postiže visok stupanj autonomije tehničkih sustava. [Slika 1] Korištenjem tih novih tehnoloških dostignuća stvaraju se pametne tvornice (eng. *Smart factory*) koje karakterizira modularna struktura, međusobno povezani digitalno-fizički sustavi koji upravljaju radnim procesima, postojanje virtualne kopije fizičkih objekata koje očitavaju trenutne podatke čime se pospješuje decentralizirano donošenje odluka. Korištenjem Interneta objekata (eng. *Internet of Things*) digitalno-fizički sustavi komuniciraju i surađuju međusobno, ali i s ljudima, u stvarnom vremenu, na internoj razini, kao i kroz ostale razine lanca vrijednosti (eng. *value chain*) [1].



Slika 1. Prikaz četiri industrijske revolucije [1]

### 1.1.2. Tehnologije/paradigme Industrije 4.0

Industrija 4.0 temelji se na tehnologiji koja omogućuje povezivanje stvarnog i digitalnog svijeta. Osnovne paradigme koje vežemo uz taj pojam su Internet objekata, Računalni oblaci (eng. *Cloud computing*), Digitalno-fizički sustavi i Veliki skupovi podataka (eng. *Big Data*). Internet objekata je mreža uređaja povezanih preko interneta čime je omogućena razmjena podataka među njima. Za prikupljanje i izmjenu podataka povezani uređaji moraju biti opremljeni s raznim elektroničkim komponentama, sensorima, aktuatorima te odgovarajućom računalnom podrškom koja uključuje razne programe i aplikacije. Povezivanjem se omogućuje međusobna interakcija uređaja i sustava čime se olakšava njihova kontrola i praćenje, ali se i otvara prostor za stvaranje novih usluga. Komunikacija može biti ostvarena između uređaja, između uređaja i ljudi, ali i između digitalno-fizičkih sustava koji uključuju oboje.



Slika 2. Industrija 4.0 [1]

Digitalno-fizički sustavi (eng. *Cyber-Physical Systems*, u nastavku CPS) su sustavi koje se sastoje od inteligentnih fizičkih komponenti, objekata ili sustava sa integriranim memorijskim i računalnim modulima, koji povezani u mreže omogućuju realizaciju pametne tvornice (eng. *Smart factory*). Tako povezani sustavi mogu međusobno komunicirati čime se otvaraju nove mogućnosti u proizvodnji, na području praćenja trenutnog stanja sustava, praćenja pojave grešaka, upravljanja i kontrole s udaljenosti, održavanja itd. Definicija CPS se u pogledu industrijske proizvodnje često preklapa sa definicijom Interneta objekata te će se u nastavku spominjati kao CPS. CPS-ovi obuhvaćaju i simulacije, digitalne blizance (eng. *Digital twin*),

pametne analitičke i logističke sustave, sustave za samo-optimizaciju i drugo. Glavne karakteristike CPS-ova slijedeće: 1) povezivanje digitalnog i fizičkog svijeta, koje karakterizira spajanje informacijskih i operacijskih tehnologija, 2) komunikacija, CPS-ovi su povezani u mreže a upravljani su pomoću inteligentnih sustava, 3) mogu biti pojedinačni identificirani putem IP adrese, 4) sadrže mehatroničke i elektroničke komponente (kontroleri, senzori, aktuatori), 5) osnovni su gradivni elementi Industrije 4.0 i omogućavaju dodatna svojstva kao što su praćenje i upravljanje na daljinu. Ključni element koji omogućuje postavljanje ovakvih pametnih sustava je Internet objekata putem povezanih senzora i aktuatora, komunikacijskih veza te prikupljanja i analize podataka [1].

*Big Data* je tehnologija koja omogućava prikupljanje i obradu velikih količina strukturiranih i nestrukturiranih podataka u realnom vremenu [2]. Glavne karakteristike velikih skupova podataka (eng. *Big data*) su količina, raznolikost i brzina [3]. Prikupljeni podaci moraju biti obrađeni sa naprednim alatima za analizu i algoritmima da bi mogli kreirati korisne informacije. Uzevši u obzir vidljive i nevidljive probleme u industriji, algoritam za generiranje informacija mora moći detektirati i prepoznati nevidljive probleme kao što su npr. trošenje komponenti strojeva i slično.

### **1.1.3. Karakteristike Industrije 4.0**

Kako bi se omogućilo uspostavljanje industrije 4.0 potrebno je ispunjenje slijedećih uvjeta: interoperabilnost, transparentnost informacija, orijentiranost uslugama, decentraliziranost i autonomno donošenje odluka, mogućnost djelovanja u realnom vremenu te modularnost [1].

Interoperabilnost se odnosi na svojstvo sustava unutar kojeg su strojevi, uređaji, senzori i ljudi povezani posredstvom interneta objekata (eng. *Internet of Things*, u nastavku IoT) i interneta ljudi (eng. *Internet of people*).

Transparentnost informacija odnosi se na mogućnost generiranja vjerodostojne digitalne kopije realnih sustava pomoću digitalnih modela nastalih temeljem podataka prikupljenih sensorima. Korištenjem velikog broja senzora ugrađenih u međusobno umrežene sustave i podsustave omogućeno je prikupljenim podacima dati dimenziju konteksta u kojem su prikupljeni. Tako je moguće izvući cjelovitije informacije o ponašanju sustava usred djelovanja konkretnih radnih uvjeta.

Uslijed sve prisutnije personalizacije proizvoda i širenja ekonomije temeljene na uslugama javlja se potreba prilagodbe proizvodnih procesa personalizaciji proizvoda dodajući im



dimenziju usluge. Na taj se način mijenja cjelokupan proces proizvodnje od onoga u kojem su proizvođači osmišljavali proizvode do najsitnijih detalja ka onome unutar kojeg je kupac taj čija osobna zamisao i želja, kroz proces personalizacije, daje završnu riječ u samom procesu proizvodnje. Okretanje industrije uslugama također se očituje kroz potrebe proizvođača za razvijanjem novih usluga, koje su bazirane na podacima podignutima na razinu informacija, s ciljem ostvarivanja dobiti putem uslužnih servisa. Najpogodnija područja za implementaciju takvih procesa su tehnička podrška i održavanje. Unutar spomenutih je branši moguće iz temelja promijeniti način poslovanja iskorištavajući puni potencijal koji IoT nudi.

Decentraliziranost i autonomno donošenje odluka odnose se na decentralizirano prikupljanje podataka sa raznih izvora koji su kasnije dostupni putem IoT servisa, koji je kao takav decentraliziran. Zahvaljujući decentraliziranosti digitalno-fizičkih sustava, a u kombinaciji s implementiranim određenim stupnjem umjetne inteligencije, pred njih se stavljaju zahtjevi autonomnog donošenja odluka uslijed njihove međusobne interakcije kao i uslijed interakcije s čovjekom.

Jedna od karakteristika Industrije 4.0 svakako je i djelovanje u realnom vremenu, što se odnosi na mogućnost osjetilnih komponenti sustava, tj. senzora i tehničke podrške, da prikupljaju i pohranjuju podatke o trenutnom stanju proizvoda i njegove okoline, koji se odmah mogu koristiti u raznim analizama koje rezultiraju pravovremenim reakcijama sustava na vanjske podražaje. Prikupljanjem trenutnih podataka o proizvodu dobiva se jasnija i cjelokupna slika o njegovom stvarnom ponašanju te se time potpomaže donošenje odluka prilikom razvoja.

Već spomenuti trend personalizacije proizvoda i usluga razlog je sve veće potrebe za razvojem modularnih proizvoda. Korisnici žele da proizvodi posjeduju točno one karakteristike koje njima trebaju, što od proizvođača zahtjeva sposobnost da naprave fleksibilne i agilne proizvode koji se mogu brzo prilagođavati novim trendovima u takvoj dinamičkoj okolini.

Tvrtke koje posjeduju sve navedene karakteristike ili teže njihovom uvođenju i integraciji u svoje poslovne procese zadovoljavaju uvjete za uvođenje Industrije 4.0. Tako mogu konstantno unaprjeđivati svoje proizvode i usluge te time ostvarivati povećanje profita i osiguravanje svojeg mjesta na tržištu.

#### ***1.1.4. Posljedice za industriju***

Nije upitno da će Industrija 4.0 utjecati na mnoge aspekte industrije, od kojih su najčešće spominjani: usluge i modeli poslovanja, pouzdanost i produktivnost, IT sigurnost i zaštita

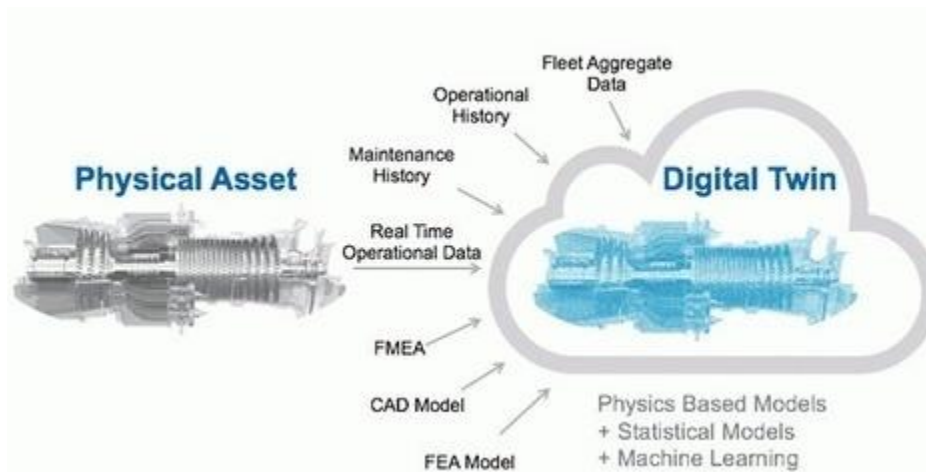
podataka, sigurnost strojeva s kojima su ljudi u kontaktu, životni ciklusi proizvoda, razvoj novih proizvoda i optimizacija radnih procesa, lanac vrijednosti u industriji, edukacija i vještine radnika, socioekonomski utjecaji itd.

Nove informacijske tehnologije za komunikaciju (*CPS, Big Data, Cloud computing*) pomažu nam pri bržem otkrivanju naznaka nepravilnosti i nedostataka proizvoda, čime omogućuju njihovu prevenciju, ali i povećanje produktivnosti, kvalitete i fleksibilnosti (agilnosti) proizvoda čime se može steći značajna kompetitivna prednost. S povećanjem težnje za personalizacijom proizvoda i usluga, poboljšanom interakcijom korisnika i proizvođača, kao i težnjom da se smanje neučinkovitosti, irelevantne informacije i troškovi posredovanja, u digitalnom lancu dobave omogućava se lakše zadovoljavanje sve većih zahtjeva kupaca, kao što su brzina, niža cijena i dodatna vrijednost inovativnih proizvoda i usluga.

## 2. Digitalni blizanci

### 2.1. Definicija

Digitalni blizanci su virtualne replike fizičkih proizvoda koje modeliraju stvarno ponašanje proizvoda u realnom vremenu, a služe za detektiranje mogućih nedostataka i problema, provedbu raznih analiza, testiranje novih funkcija i simuliranje mogućih scenarija koje bi bilo skupo ili čak nemoguće provesti na fizičkom proizvodu. Oni uz pomoć senzora i mnogobrojnih tehnologija u digitalnoj okolini prikazuju elemente i dinamiku proizvoda koji su povezani preko Interneta objekata, čime se omogućuje praćenje stanje uređaja i njegove okoline, te način na koji se oni mijenjaju tokom životnog ciklusa. Uloga senzora je dobivanje ispravnih podataka o trenutnom stanju uređaja i okoline, koji se pohranjuju na računalo. Osim fizičkih izvora podataka, u računalima se pohranjuju i podaci kreirani u digitalnoj okolini, kao što su primjerice konstrukcijske specifikacije, CAD modeli, liste komponenti i materijala, rezultati proračuna, simulacija i slično. Kombinacijom tih podataka sa podacima iz prikupljenima putem senzora dobiva se cjelovita slika rada i performansi sustava koja omogućuje pravovremeno djelovanje u fazi korištenja i održavanja, ali i bolju prilagodbu proizvoda stvarnim radnim uvjetima. [Slika 3]



Slika 3 Shematski prikaz modela digitalnog blizanca [4]

Uvođenje digitalnih blizanaca promijeniti će način na koji će konstrukcija proizvoda, njihova građa i procesi kroz koje prolaze utjecati na životni vijek proizvoda. U fazi konstruiranja proizvoda za Industriju 4.0 trebaju se uvažiti sve potrebne fizičke komponente i odgovarajuću računalnu podršku tako da se komponente mogu povezati u funkcionalni sustav. Kako je već

rečeno, jedna od glavnih karakteristika proizvoda Industrije 4.0 je modularnost, što povlači za sobom potrebu da sustavi imaju mogućnost obavljanja velikog broja različitih funkcija istovremeno, zbog čega će dodatno povećavati stupanj složenosti sustava. Građa proizvoda odnosi se na utjecaj pojedinih komponenti na fizičku izdržljivost cijelog proizvoda, te na tolerancije koje je potrebno zadovoljiti. Praćenjem stvarnih podataka vezanih uz stanje proizvoda može se dobiti realniji uvid u stvarno opterećenje i trošenje, čime se pruža mogućnost za smanjivanje potrebnih faktora sigurnosti i finiju prilagodbu tolerancija, a time i troškova. Tokom životnog ciklusa proizvodi obavljaju različite operacije prilikom čega prolaze kroz različite radne uvjete, njihove komponente doživljavaju mnoge promjene te se posljedično tome i troše. Praćenjem procesa kroz koje proizvodi prolaze tokom svog životnog vijeka te njihovih parametara pomoću digitalnih blizanaca se može uvidjeti prostor za njihovu optimizaciju.

Krajnja svrha upotrebe digitalnih blizanaca je smanjenje ukupnih troškova, pravovremena upozorenja kod nepravilnosti rada uređaja čime se optimizira održavanje, stvaranje mogućnosti za razvoj novih proizvoda i usluga, poboljšanje postojećih proizvoda u vidu optimizacije komponenti ili smanjenja vremena čekanja između pojedinih operacija, te povezivanje raznih podsustava u mrežu radi dobivanja jasnije slike o ponašanju kompleksnih sustava čime se olakšava donošenje odluka.

## **2.2. Povijest razvoja paradigme digitalnih blizanaca**

U periodu uvođenja pojma digitalnog blizanca (eng. *Digital Twin*) s početka 2000-ih godina tehnologija virtualnog prikaza proizvoda bila je relativno nova i nerazrađena, dok se prikupljanje informacija o fizičkim proizvodima u proizvodnom procesu svodilo na ručno zapisivanje podataka na papiru. U periodu koji je uslijedio programska podrška razvoju virtualnih proizvoda konstantno se nastojala unaprijediti, kao i mogućnosti u proizvodnji fizičkih proizvoda. Prikupljanje podataka počinje se odvijati digitalnim putem korištenjem raznovrsnih nerazornih tehnologija kao što su senzori i mjerni uređaji, vizijski sustavi i sl.

Spomenutim se tehnologijama omogućuje povezivanje fizičkog i virtualnog proizvoda tvoreći tako cjeloviti digitalno-fizički sustav koji se sastoji od fizičkog proizvoda u realnom prostoru, virtualnog modela u virtualnom prostoru te protoka podataka iz fizičke u virtualnu okolinu.

Daljnijim je razvojem tehnologije značajno povećana mogućnost obrade podataka te su razvijeni softveri koji uz vizualizaciju proizvoda omogućavaju provođenje zahtjevnih računalnih simulacija s ciljem što preciznijeg određivanja njihovog ponašanja u realnim uvjetima. Nadalje

je napravljen značajan pomak u načinu zapisivanja modela u pojednostavljenim formatima smanjujući tako potrebu za računalnom snagom prilikom obrade podataka te posljedično omogućavajući izvršavanje simulacija u realnom vremenu (eng. *real time*). Isto je tako implementacijom raznih osjetilnih podsustava omogućeno praćenje i pohranjivanje raznih parametara sustava kao što su npr. sile i momenti koji se javljaju prilikom korištenja uređaja [5].

Usprkos navedenim tehnološkim iskoracima mogućnosti povezivanja fizičke i virtualne okoline nisu u potpunosti iskorištene. Kao najjednostavniji primjer može se uzeti izrada 2D crteža iz 3D virtualnog modela s ciljem pretakanja virtualnog modela na papir u procesu pripreme proizvodnje poluproizvoda. Taj se crtež kasnije koristi kako bi čovjek napisao program za obradni stroj, u virtualnoj okolini, za proizvodnju željenog poluproizvoda.

Cilj povezivanja virtualnog i fizičkog proizvoda je omogućiti komunikaciju između njih na način da se korištenjem implementiranih osjetnika u realnom vremenu prati stanje fizičkog sustava te se neprestano evaluira i uspoređuje s digitalnim, idealiziranim slučajem kako bi se osiguralo ispunjenje svih željenih kriterija kao što su npr. dimenzijska točnost obratka, optimiranje procesa i performansi sustava i sl., te da se prikupljeni podaci pohrane za daljnju obradu s ciljem optimizacije daljnjeg rada sustava, kao i što kvalitetnijeg razvoja nove generacije uređaja i sustava.

### **2.3. Digitalni blizanci u razvojnom procesu**

U kontekstu razvojnog procesa zanimljivo je istaknuti kako tehnologija digitalnog blizanca podupire tri najbitnije karakteristike ljudskog razmišljanja: konceptualizaciju, sposobnost uspoređivanja te suradnju [5].

Za razliku od računala koja informacije obrađuju serijski, korak po korak, ljudi informacije procesuiranju u vezi s kontekstom u kojem su nastale. Iz tog je razloga procesuiranje velike količine podataka nepovoljno za ljudski način razmišljanja koji te podatke procesuiraju u manjim skupinama, razlažući ih na osnovne faktore kao što su slova i brojke na papiru a potom ih ponovo povezuje tvoreći nove koncepte. Iz opisanog razloga često dolazi do gubitka značajnog dijela primljenih informacija uz vremensku neefikasnost procesa.

U slučaju digitalnog blizanca digitalnim se putem zaobilaze međukoraci potrebi da bi se iz niza detalja stvorila cjelovita slika promatranog procesa već se pomoću simulacije digitalnog blizanca može pratiti, po potrebi i u realnom vremenu, odvijanje različitih procesa i promjene

određenih parametara i karakteristika. Spomenuto ponajviše dolazi do izražaju promatra li se kompleksan sustav podijeljen na manje kompleksne konstrukcijske značajke od kojih se svaka zasebno može uspoređivati sa njoj odgovarajućim izmjerenim vrijednostima.

Model digitalnog blizanca moguće je implementirati na način da se usporedba ključnih parametara i nominalne vrijednosti provodi pomoću vizualizacije vodeći računa o granicama tolerancijskog polja pojedinih značajki. Vizualizacijom npr. bojama, gdje zeleno označava da izmjereni podatak ulazi u zadano tolerancijsko polje, a crveno izlazak iz njega, korisniku se omogućava učinkovitije praćenje trendova promjena promatranih značajki kao i definiranje kritičnih točaka procesa. Implementacija opisanog modela ima za posljedicu učinkovitiju identifikaciju i uklanjanje uzročnika neprihvatljivih odstupanja kao i efikasnije optimiranje budućih procesa.

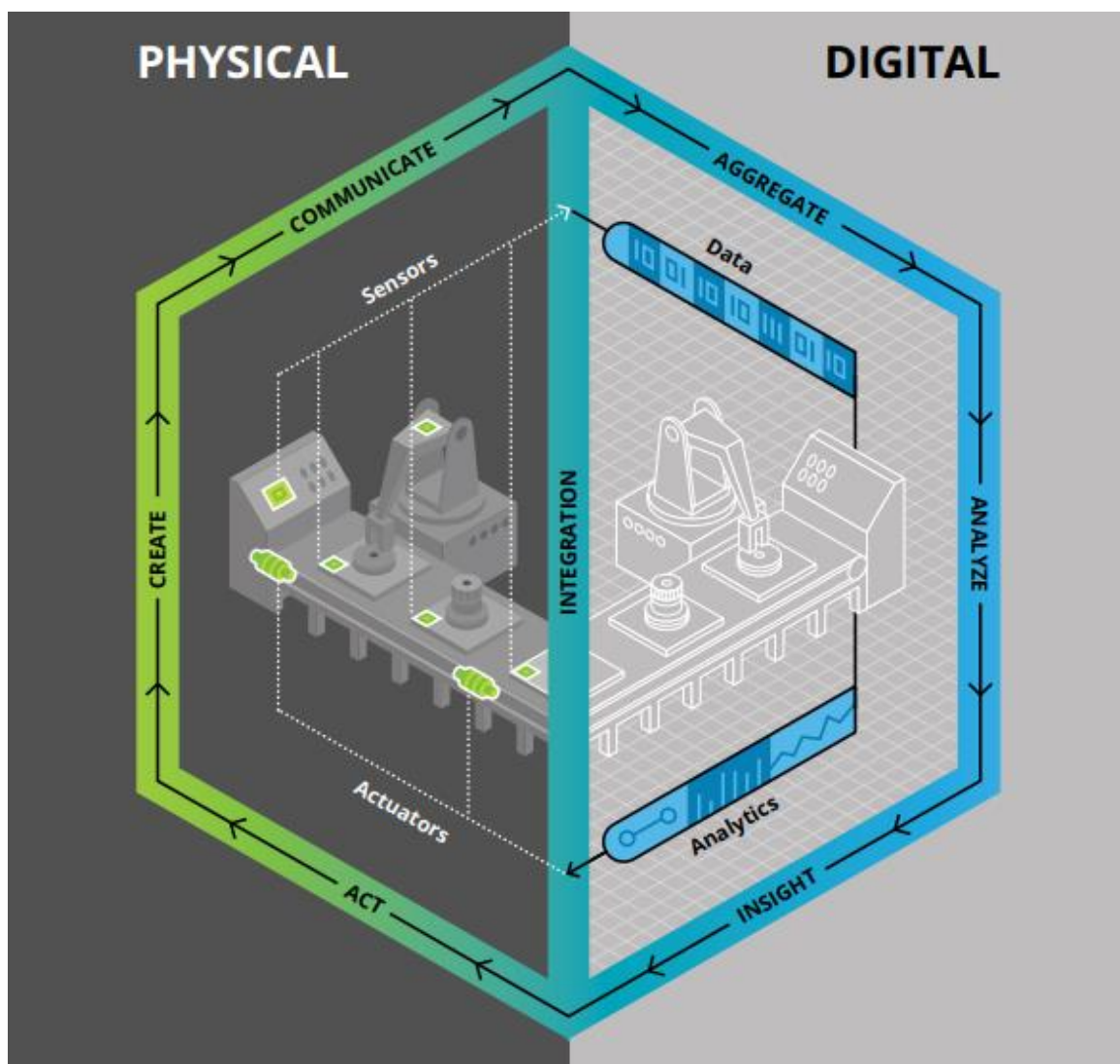
Pretvaranjem fizičke stvarnosti u virtualnu projiciranjem fizičkih parametara na digitalni model omogućava se da osobe koje surađuju na danom projektu u isto vrijeme dobivaju na isti način konstruirane i prikazane podatke. Na taj se način izbjegava mogućnost da svaki od aktera na sebi svojstven način interpretira priložene informacije čime se osigurava pravovremenost i sljedivost toka informacija te se pospješuje ostvarenje kolaboracije dislociranih timova.

#### **2.4. Komponente digitalno-fizičkih sustava**

Digitalni se blizanci baziraju na mnogobrojnim, kumulativnim, stvarnim podacima prikupljenim sa različitih izvora, mjerenima u realnom vremenu. Pomoću tih podataka kreira se dinamički model proizvoda ili procesa u digitalnom obliku, koji može dati bolji uvid u performanse sustava te time potaknuti promjene koje je potrebno provesti na fizičkom sustavu u svrhu njegovom poboljšanja. Zahvaljujući brzom razvitku računalnih tehnologija posljednjih godina, performanse računalnih sustava postale su veće uz istodobno smanjivanje njihove cijene, te je postalo moguće u kratkom vremenu izvoditi kompleksne algoritme sa velikim brojem podataka u svrhu provođenja raznih analiza i simulacija [6].

Slika 4 prikazuje primjer digitalno-fizičkog sustava proizvodne linije, koji se sastoji od fizičkog dijela i digitalnog blizanca. Na njemu su prikazane ključne komponente koje svaki sustav mora sadržavati da bi bilo moguće izraditi njegovu digitalnu kopiju. Kao dio fizičke okoline proizvod mora biti opremljen sensorima koji šalju signale o trenutnom stanju uređaja i okoline, te aktuatorima pomoću kojih se ostvaruju željene reakcije. Ono što povezuje fizički i digitalni dio su podaci. Sensori ih očitavaju iz fizičke okoline te ih šalju u računalo gdje se oni pohranjuju u

digitalnom obliku. Proces u kojem se povezuju fizička i digitalna okolina te se omogućava izmjena podataka među njima naziva se integracija. Osim fizičkih izvora podataka, u računalima se pohranjuju i podaci kreirani u digitalnoj okolini, kao što su primjerice konstrukcijske specifikacije, CAD modeli, liste komponenti i materijala, rezultati proračuna, simulacija i slično. Pohranjene je podatke potrebno prevesti u oblik u kakvom će se moći koristiti u kombinaciji sa podacima koji potječu iz različitih izvora. Tek tada se mogu provoditi razne analize podataka kroz algoritamske simulacije i vizualizacije čime se pomoću digitalnih blizanaca stvara jasnija slika i pruža bolji uvid u rad i ponašanje sustava. Rezultati provedenih analiza upućivat će na moguće nepravilnosti ili prostor za optimizaciju na temelju kojih se nastoji poboljšati ili korigirati ponašanje sustava pomoću aktuatora.



Slika 4 Digitalno fizički sustav proizvodne linije [6]

## 2.5. Koraci za kreiranje digitalnih blizanaca

Kreiranje digitalnih blizanaca može se podijeliti na dva dijela [6]:

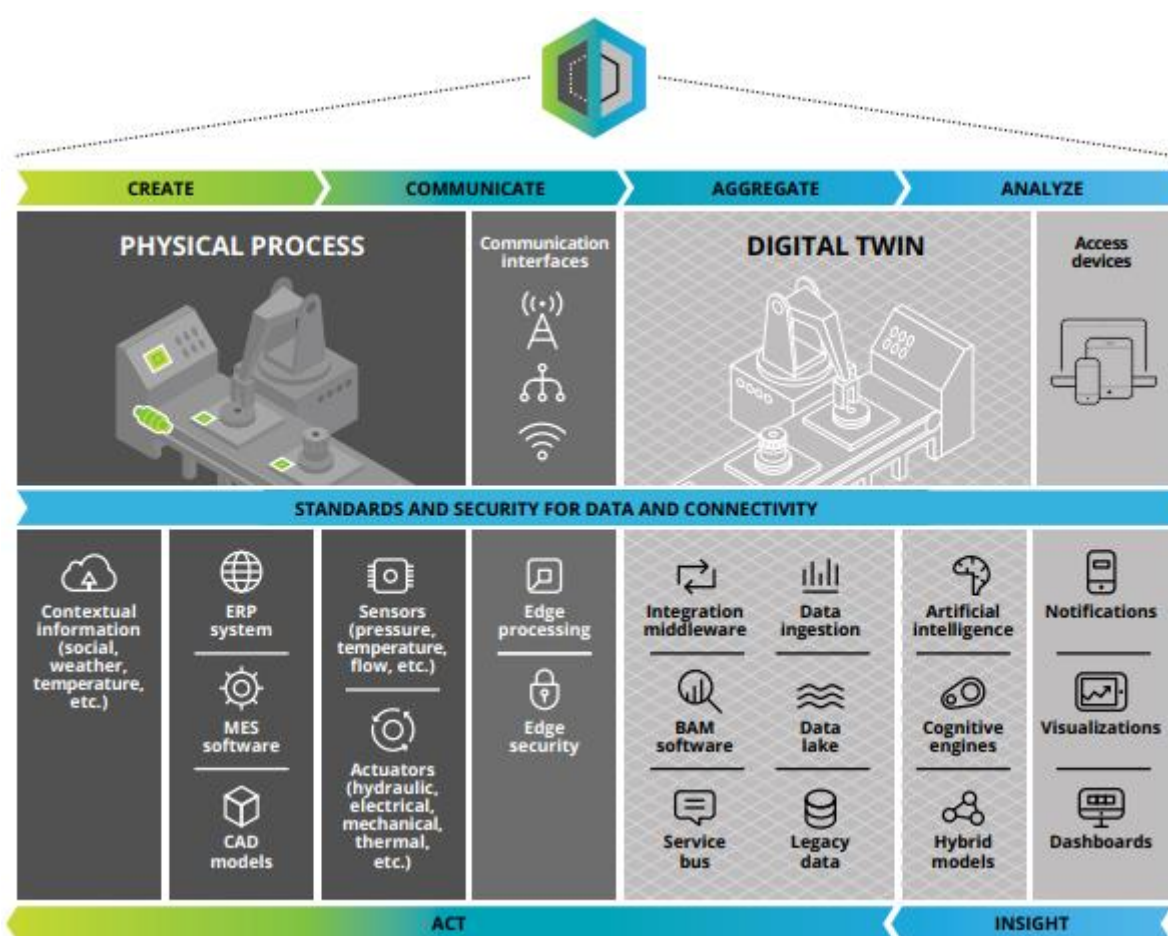
- 1) Određivanje procesa i zahtjeva za informacije tokom životnog vijeka proizvoda koje digitalni blizanac mora sadržavati
- 2) Razvoj tehnologije koja omogućava prikupljanje, pohranu i korištenje podataka sa senzora i drugih podataka sa računala, kojom će se integrirati fizički proizvod i njegova digitalna kopija

U prvom dijelu potrebno je definirati što se želi postići izradom digitalnog blizanca nekog proizvoda. Kao željeni cilj najčešće se navodi optimizacija troškova i/ili vremena potrebnog za obavljanje operacija te povećanje učinkovitosti postojećih proizvoda. U toj se fazi primjenjuju standardne metode za modeliranje procesa koje zahvaćaju poslovne procese, primjenu, informacije, fizičke komponente i ljude koji su uključeni u proces, nakon čega se izrađuju dijagrami koji tok procesa povezuju sa primjenama, podatkovnim zahtjevima i vrstama potrebnih senzora.

Drugi dio se odnosi na oblikovanje arhitekture sustava koja će omogućavati odvijanje svih prethodno definiranih procesa, što obuhvaća odabir fizičkih komponenti i tehničke podrške za uspješno prikupljanje, pohranu, dohvaćanje, pretvorbu i analizu podataka. Oblikovanje arhitekture digitalnog blizanca može se najbolje opisati kao slijed ovih šest koraka: kreiranje, komunikacija, prikupljanje, analiza, sagledavanje, djelovanje. U nastavku na prethodnu sliku Slika 5 prikazuje koji se koraci oblikovanja arhitekture digitalnog blizanca odvijaju u fizičkoj, a koji u digitalnoj okolini. Korak kreiranja odnosi se na modeliranje sustava koji će moći prikupljati podatke o radnim parametrima pomoću senzora. Podaci prikupljeni sensorima mogu se podijeliti u dvije skupine: podaci koji prikazuju ponašanje sustava te podaci koji opisuju radnu okolinu. Komunikacija predstavlja postupak integriranja, tj. Povezivanja fizičkih procesa i digitalne platforme. Mrežna komunikacija ključni je faktor koji omogućava stvaranje digitalnih blizanaca, a sastoji se od su procesnog sučelja kroz koje se prenose signali i podaci od proizvoda do digitalne platforme i kroz nju, komunikacijskog sučelja kojim se omogućava komunikacija između komponenti i platforme te sigurnosnih sustava za zaštitu podataka. Idući korak je prikupljanje koje obuhvaća prikupljanje podataka, njihovu pohranu i pripremu za analizu. Kroz analizu podaci se obrađuju te se rezultati vizualno prikazuju, što omogućava korisniku da jasnije percipira cijeli sustav i uvidi gdje se otvara prostor za poboljšanje. Prema



tome se donosi odluka za daljnje djelovanje koje se realizira u fizičkom obliku pomoću aktuatora. [6]



Slika 5 Koraci oblikovanja arhitekture digitalnog blizanca [6]

## 2.6. Primjeri primjene digitalnih blizanca

Digitalni blizanci mogu se primjenjivati u različitim granama ljudskog djelovanja, od kojih se najčešće spominju slijedeće: personalizacija proizvoda, optimizacija performansi i povećanje učinkovitosti, konstruiranje i razvoj industrijskih strojeva i robota, zdravstvo, pametni gradovi (eng. *Smart cities*) te održavanje i testiranje proizvoda [8], [9] **Error! Reference source not found.**

Da bi se što bolje prikazala primjena digitalnih blizanca u nastavku je dano nekoliko konkretnih primjera.

### 2.6.1. Digitalni blizanci zrakoplova

Digitalni blizanci pogodni su za vizualizaciju rada kompleksnih sustava čime omogućavaju lakše uočavanje bilo kakvih odstupanja parametara izvan tolerancijskih granica. Zrakoplovi su primjeri jednih od najsloženijih sustava uopće te se sastoje od velikog broja komponenti koje su međusobno povezane i utječu jedne na druge. Počne li se jedna komponenta ponašati izvan dopuštenih tolerancija, te se njen utjecaj proširi na ostale, može doći do fatalnih posljedica. Iz tog je razloga potrebno pratiti ponašanje komponenti, ali i radne uvjete u kojima se one nalaze, a posebnu je pažnju potrebno posvetiti onima koje mogu uzrokovati zakazivanje ključnih dijelova sustava kao što su npr. motori. Tvrtka General Electric služi se sensorima za mjerenje podataka na mlaznom motoru koji se mogu analizirati u kombinaciji sa različitim podacima, a jedan od primjera su i snimača leta [10]. Takvim analizama utvrđuju se uzroci zakazivanja avionskih dijelova na temelju čega se oni kasnije mogu konstruirati na drugačiji način, tako da izdrže kritičnije uvjete ili se može odrediti kada će biti potrebna njihova zamjena prije nego dođe do zakazivanja.



**Slika 6** Prikaz mjerenih podataka na motoru zrakoplova u digitalnog okolini [10]

U analizu se također mogu unijeti i podaci o vremenskim uvjetima, tako npr. za zrakoplove koji su dugo vremena letjeli u suhoj i toploj klimi digitalni blizanac može predložiti zamjenu sa zrakoplovima koji su dugo vremena proveli u hladnoj klimi. Obzirom da svaki zrakoplov ima svoju virtualnu kopiju koja se identificira prema IP adresi, digitalni blizanac može ustanoviti točno određeni zrakoplov ili više njih koji bi bili pogodni za ovakvu zamjenu.

Navedeni primjeri pokazuju kako se digitalni blizanci mogu koristiti kod održavanja, dajući pravovremena upozorenja u slučaju nekontroliranog ponašanja pojedinih komponenata ili nastanka nepovoljnih uvjeta u radnoj okolini. U tu se svrhu također mogu pratiti trendovi ponašanja komponenti te na temelju toga predvidjeti kada će biti potrebna njihova zamjena.

### 2.6.2. Digitalni blizanac brdskog bicikla

Jedan od primjera tehnologije digitalnog blizanca je kolaborativni projekt tvrtke Santa Cruz Bicycles iz Kalifornije i tvrtke PTC [11]. Radi se o brdskom biciklu opremljenom sensorima koji preko platforme PTC ThingWorx prikuplja i obrađuje podatke o korištenju uređaja. Bicikl je opremljen računalnom jedinicom preko koje se prikupljaju i šalju podaci očitavani sa senzora postavljenih na ključne dijelove bicikla. Prikupljaju se podaci o broju okretaja kotača, pedala, pomaci ovjesa, kut zakreta volana koji se kasnije preko ThingWorx platforme mogu iskoristiti za provođenje simulacija unutar PTC Creo programskog paketa [Slika 7 a)].

Podaci se također mogu iskoristiti u svrhe proširene stvarnosti koja u realnom vremenu preko prikaza bicikla podvrgnutog radnim uvjetima može prikazivati podatke prikupljene pripadajućim sensorima [Slika 7 b)].

Korištenjem takve tehnologije moguće je dobiti detaljan uvid u eksploatacijske uvjete proizvoda što se može iskoristiti u svrhe preventivnog održavanja uređaja te boljeg definiranja ulaznih parametara kako bi se ubrzao i pospješio proces optimizacije i razvoja idućih verzija istog.



a)

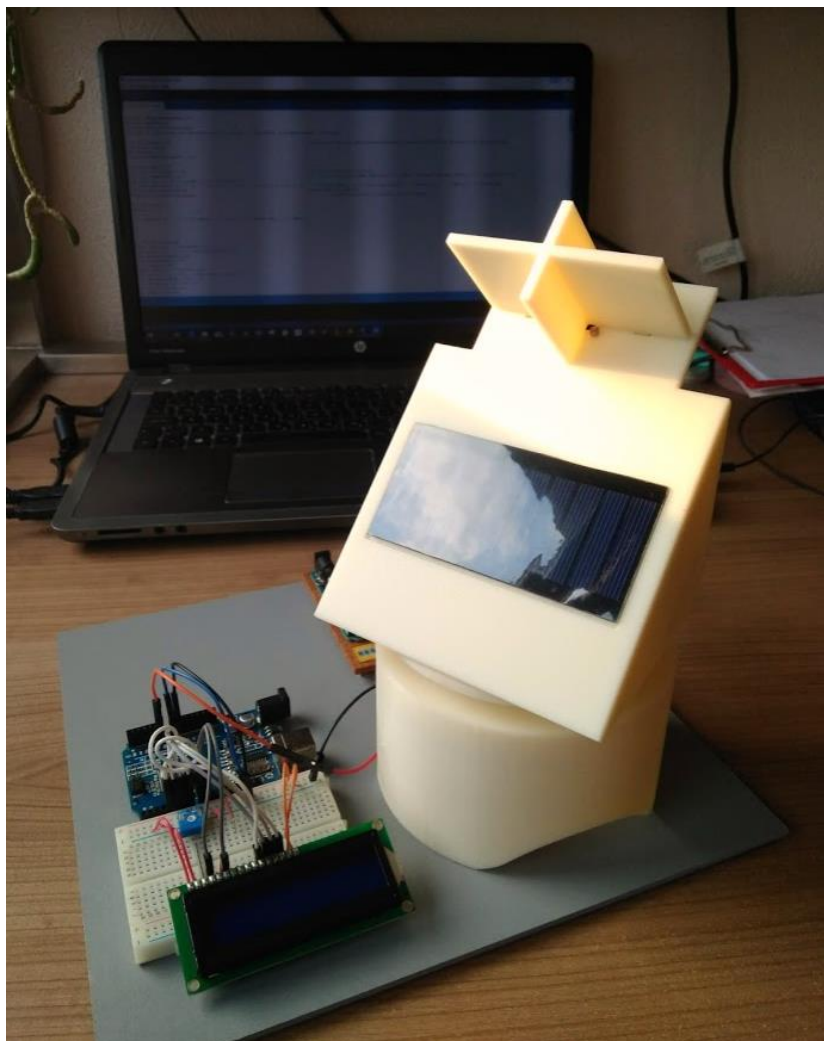


b)

Slika 7 Digitalni blizanac brdskog bicikla [11]

### 3. Primjer razvoja digitalnih blizanaca

U svrhu realizacije paradigme digitalnih blizanaca osmišljen je i izrađen uređaj [Slika 8] koji ima ugrađen solarni panel te prati izvor svjetlosti pomoću senzora kako bi se povećala učinkovitost proizvodnje električne energije. Takav uređaj bi se mogao nazvati pametnim (eng. *Smart product*) proizvodom jer je svjestan svoje okoline te sukladno tome prilagođava svoje ponašanje. Svrha njegovog digitalnog blizanca je predviđanje koliko bi energije takav uređaj mogao proizvesti u određenom periodu u promjenjivim okolišnim uvjetima na temelju mjerenja raznih parametara tokom rada. Ovakva se primjena digitalnog blizanca može promatrati kao težnja za povećanjem učinkovitosti već postojećeg proizvoda, što je u ovom slučaju solarni panel na nepomičnoj platformi, ali i kao razvoj proizvoda sa novim mogućnostima čime se dodaju nove vrijednosti za korisnika.



**Slika 8** Uređaj korišten za izradu digitalnog blizanca

### 3.1. Funkcionalnost uređaja

Solarni paneli proizvode više električne energije ako je izvor svjetlosti direktno usmjeren prema njima. Iz tog je razloga poželjno da budu okrenuti prema suncu što je više vremena moguće. Ovaj je uređaj napravljen na način da će se, neovisno o njegovom početnom položaju, uvijek okretati prema izvoru svjetla uz pomoć senzora. Ako je uređaj ugrađen na neko fiksno mjesto, može se izračunati sunčeva putanja kroz godinu te unaprijed programirati uređaj da je prati. U takvom slučaju ne bi bilo potrebe za uvođenjem svjetlosnih senzora na uređaj. No, ako je uređaj ugrađen na pomični objekt ili ga je potrebno premješati tokom vremena, senzori uklanjaju potrebu za reprogramiranjem uređaja. Takvi uređaji, koji su svjesni svoje okoline te na nju istovremeno reagiraju, mogu naći svoju primjenu na brodovima, kamp kućicama i slično. Uređaj koji je izrađen i detaljnije opisan dalje u radu je pojednostavnjena verzija pravog uređaja, ali ima prikazane sve potrebne funkcije i komponente te je za njega izrađen digitalni blizanac.

### 3.2. Prototip

Za potrebe ovog rada bilo je potrebno osmisliti fizički uređaj koji zadovoljava sve uvjete za izradu digitalnog blizanca te u digitalnom okruženju kreirati njegovu virtualnu inačicu. Obzirom da ne postoji puno pokaznih primjera za izradu digitalnog blizanca, trebalo je što prije krenuti u modeliranje sustava u digitalnoj okolini, njegovo postavljanje i testiranje da bi se u konačnici uspio dobiti funkcionalan model. Iz tog je razloga fizički uređaj prvotno bio izrađen od kartona [Slika 9] sa pričvršćenim sensorima i servomotorima koji su se putem žica povezivali sa mikrokontrolerom. Iz njega su se podaci očitani sa senzora slali na računalo preko USB veze na temelju čega se gibanje uređaja upravljalo kroz programski kod, a podaci o gibanjima motora su se istovremeno slali u 3DExperience digitalnu radnu okolinu. Slika 9 prikazuje prototipnu izvedbu uređaja.



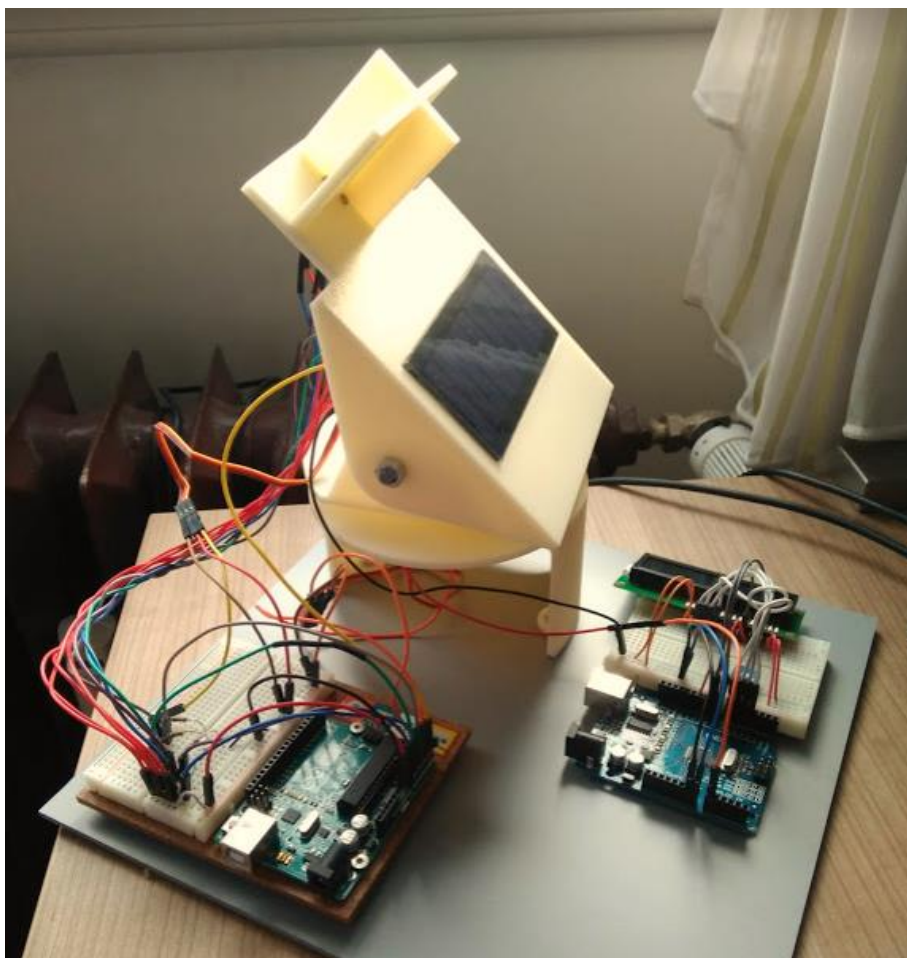


**Slika 9 Rana faza fizičkog prototipa**

### **3.3. Komponente uređaja**

Na temelju prototipa napravljena je nova inačica uređaja [Slika 10], čija je nosiva konstrukcija izrađena postupkom 3D printanja. Konačna verzija uređaja sastoji se od pet podsustava:

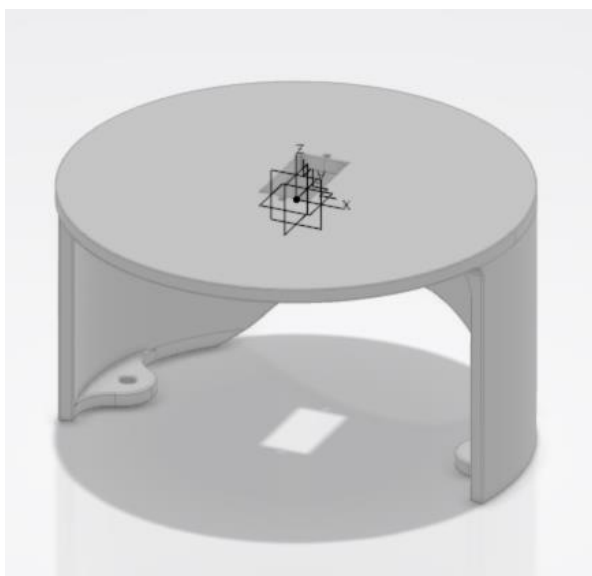
1. Nosiva konstrukcija
2. Upravljačka jedinica
3. Senzori
4. Aktuatori
5. Solarni panel



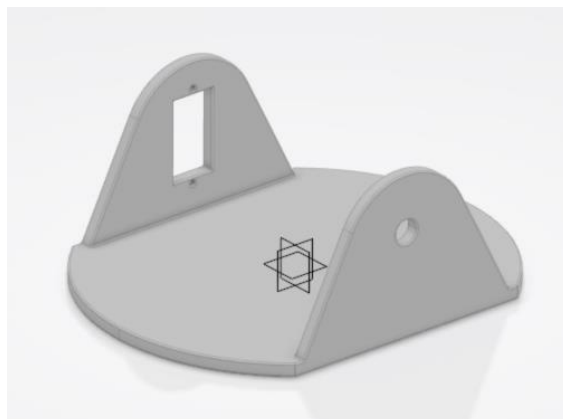
**Slika 10** 3D printana inačica uređaja

### **3.3.1. Nosiva konstrukcija**

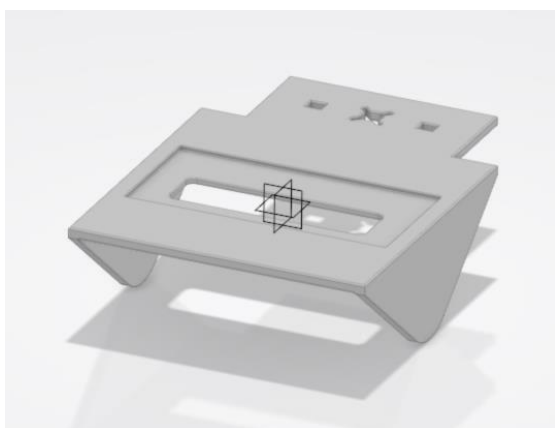
Nosiva konstrukcija napravljena je od tri glavna dijela [Slika 11]: a) postolje koje je učvršćeno za podlogu, b) središnji rotacijski dio i c) gornji rotacijski dio. Središnji dio rotira u horizontalnoj ravnini, a gornji u vertikalnoj ravnini. Pozicioniranje solarnog panela prema izvoru svjetlosti postiže se kombinacijom ta dva gibanja. Na gornjem dijelu nalazi se pregrada [Slika 11 d)] križnog oblika koja ograničava izvor svjetlosti da dopire do senzora ako nije direktno na njih usmjerena, tako da senzori imaju jasnija očitavanja.



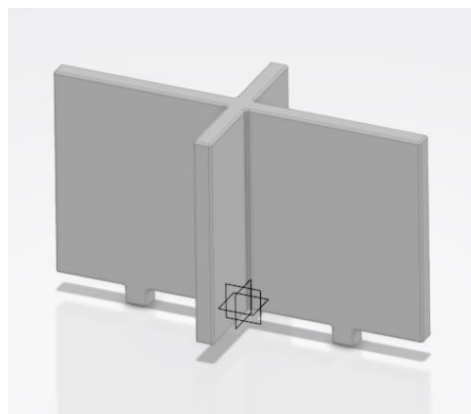
a) Postolja



b) Središnji rotacijski dio



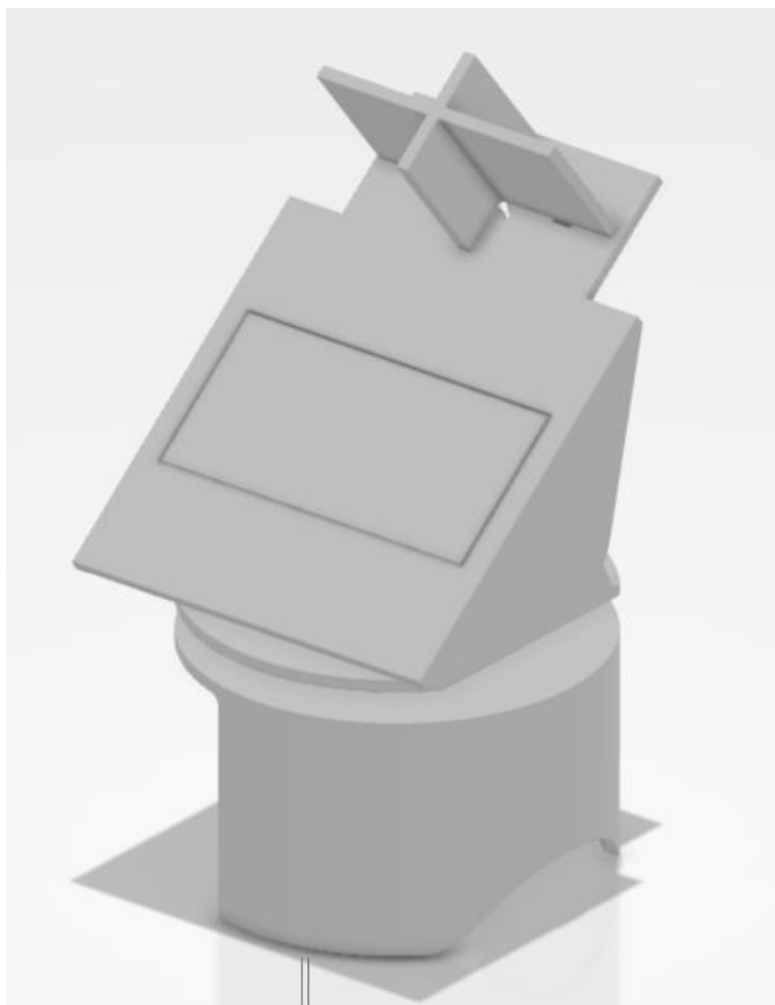
c) Gornji rotacijski dio



d) Križna pregrada

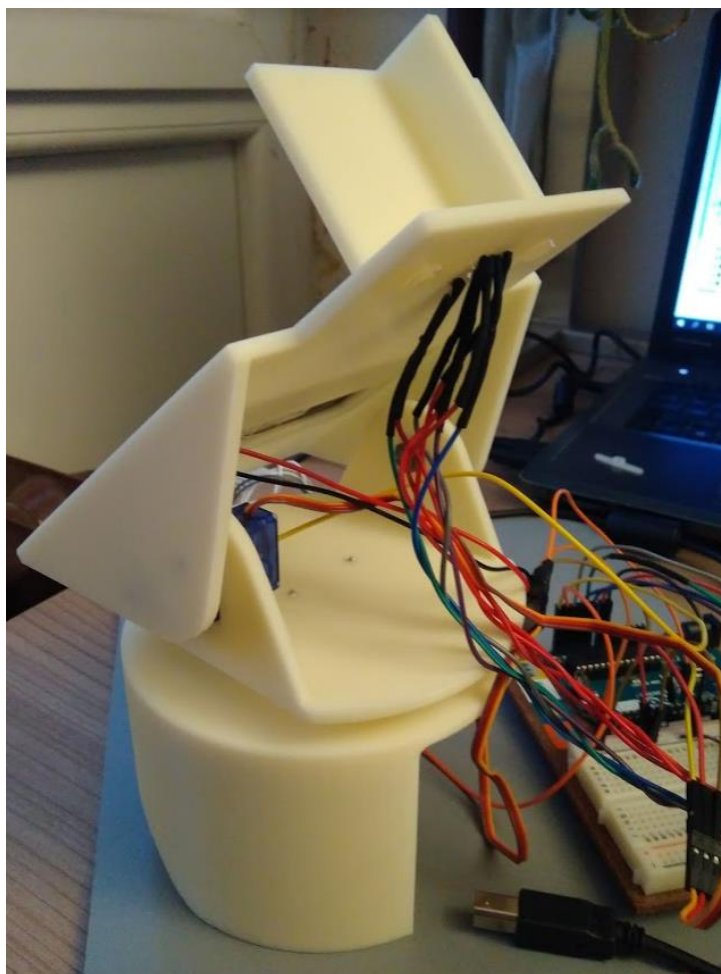
**Slika 11 Komponente nosive konstrukcije**





**Slika 12 3D model uređaja**

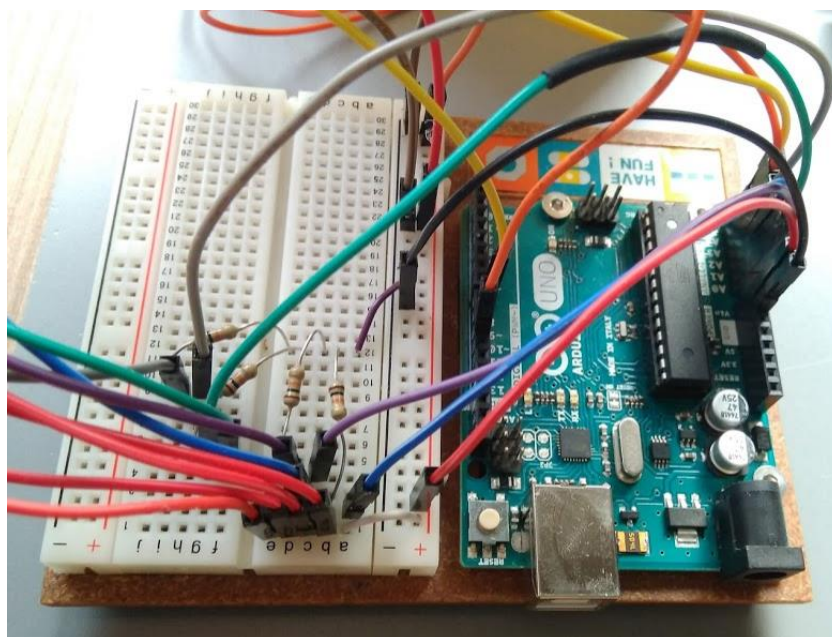
Svi su dijelovi nosive konstrukcije [Slika 12] izrađeni 3D printanjem te je kod konstruiranja trebalo uvažiti smjernice za pravilno oblikovanje dijelova, npr. izbjegavati prave kutove i provrte malih dimenzija te pravilno odrediti debljinu stjenke [12]. Svi su dijelovi povezani rastavljivim vezama – vijcima osim križne pregrade koja je ljepilom učvršćena na gornji dio konstrukcije. Nakon toga su na uređaj ugrađene sve kupljene elektroničke komponente sa pripadajućim ožičenjem koje je za potrebe ovog rada trebalo modificirati [Slika 13]



**Slika 13** Pogled na stražnji dio uređaja – nosiva konstrukcija i ožičenje

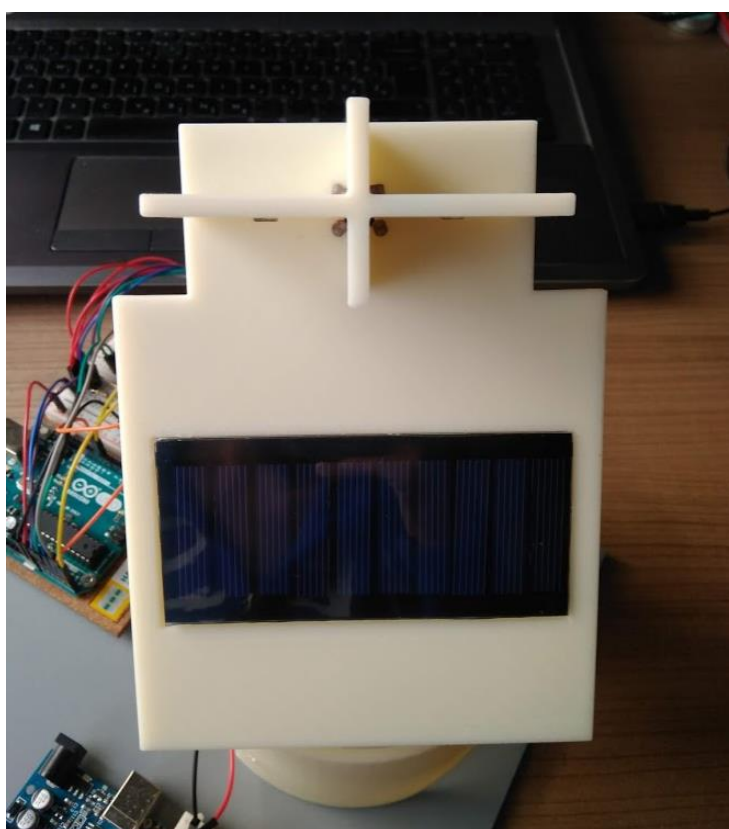
### **3.3.2. Upravljačka jedinica**

Cijeli je uređaj upravljan preko Arduino elektroničke platforme koja je povezana s računalom USB kablom. Željeni program napisan je u Arduino IDE softveru, te učitana na Arduino mikrokontroler. Aktuatori, tj. servomotori, senzori i solarni panel su povezani s mikrokontrolerom preko žica, a radi preglednosti i lakšeg spajanja žica i otpornika koristi se eksperimentalna ploča (eng. *breadboard*) [Slika 14].



**Slika 14** Arduino mikrokontroler (desno) i eksperimentalna pločica (lijevo)

### 3.3.3. Senzori

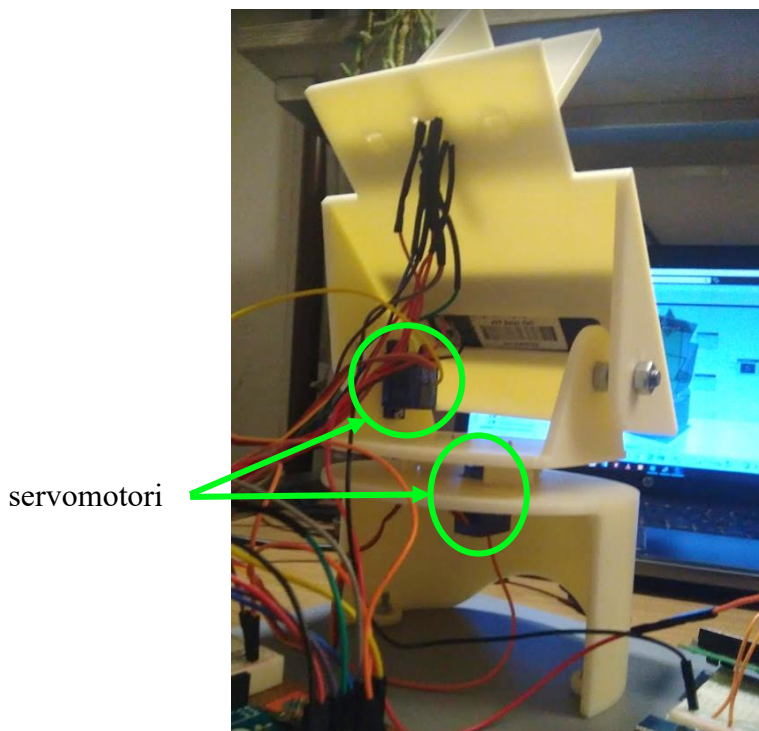


**Slika 15** Fotootpornici postavljeni na uređaj

Za određivanje položaja izvora svjetlosti koriste se četiri fotootpornika međusobno odijeljenih križnom pregradom [Slika 15] koji funkcioniraju na način da im se električni otpor smanjuje s povećanjem intenziteta ulazne svjetlosti [15]. Kada je gornji rotacijski dio okrenut direktno prema izvoru svjetlosti otpor svakog fotootpornika je jednak te uređaj miruje. Kada se izvor svjetlosti pomakne dolazi do smanjivanja otpora fotootpornika te se na ulazni konektor uređaja šalje veći napon. Upravljački kod uređaja napisan je tako da se uređaj zakreće u onom smjeru gdje je iznos otpora za svaki fotootpornik najmanji.

### 3.3.4. Aktuatori

Kod napisan u Arduino IDE softveru služi za upravljanje servomotrima [Slika 16] pomoću podataka očitanih na fotootpornicima. Jedan servomotor služi za rotaciju središnjeg dijela u horizontalnoj ravnini, a drugi vrši rotaciju gornjeg dijela u vertikalnoj ravnini. Svaki servomotor povezan je na Arduino mikrokontroler koji im daje napajanje, uzemljenje i preko posebne žice dobivaju naredbe za upravljanje. U svrhu izrade digitalnog blizanca uređaja bilo je potrebno dodatno zalemiti još jednu žicu preko koje mikrokontroler može očitati kut zakreta propelera motora čime se pogoni 3D model mehanizma.



Slika 16 Aktuatori postavljeni na uređaj

### 3.3.5. Solarni panel

Solarni panel sačinjen je od skupa solarnih ćelija te mu je uloga pretvaranje energije sunčeva zračenja u električnu energiju uslijed odvijanja fotoelektričnog efekta. Na taj je način omogućeno iskorištavanje do četrdesetak posto prikupljenog zračenja što se u komercijalno dostupnim sustavim namijenjenima upotrebi od strane široke populacije svodi na svega petnaest do dvadeset posto 0. Korisnost fotonaponske solarne ćelije definira se kao omjer električne snage koju daje fotonaponska ćelija i snage sunčevog zračenja prema izrazu (1):

$$\eta = \frac{P_{el}}{P_{sol}} = \frac{U \cdot I}{E \cdot A} \quad (1)$$

gdje je:

$P_{el}$  - Izlazna električna snaga

$P_{sol}$  - Snaga zračenja (najčešće Sunčevog)

$U$  - Efektivna vrijednost izlaznog napona

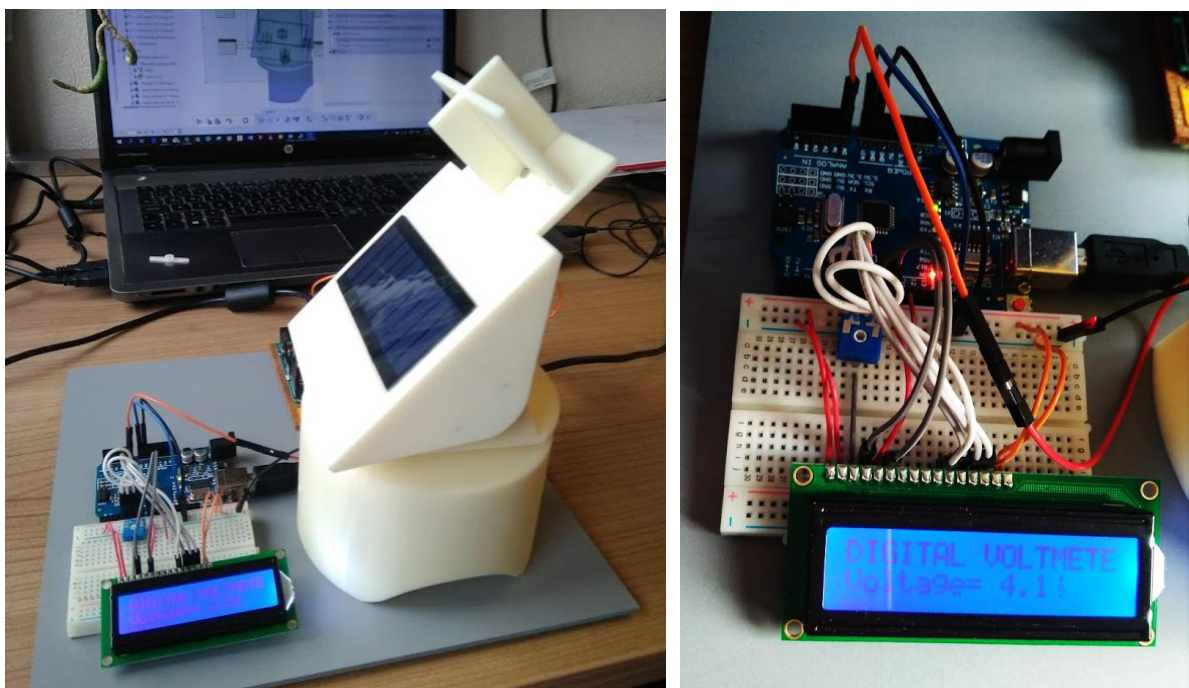
$I$  - Efektivna vrijednost izlazne struje

$E$  - Specifična snaga zračenja (npr. u  $W/m^2$ )

$A$  - Površina

Poznato je kako na izlaznu snagu solarnog panela utječu nazivna snaga solarnog panela (produkt jakosti struje i napona), intenzitet sunčeve svjetlosti, broj sati izloženosti suncu, kut gledanja prema suncu te radna temperatura. Isto je tako poznato da fotonaponske ćelije proizvode najviše električne energije kada fotoni na njih dolaze iz smjera normale na njihovu površinu. Većina navedenih parametra izvan je utjecajne domene čovjeka. Jedna od mogućnosti za poboljšanje učinkovitosti solarnih panela je da, u što je moguće većoj mjeri, zadržavaju pravi kut naspram sunca što se na ovom uređaju postiglo korištenjem svjetlosnih senzora i sukladno navođenom pokretnom platformom na kojoj se nalazi solarni panel. Solarni panel je pričvršćen na gornji dio nosive konstrukcije, a napon koji se na njemu postiže prikazuje se u realnom vremenu na LCD ekranu spojenom s mikrokontrolerom [Slika 17]. Prema specifikacijama odabrani solarni panel može postizati do maksimalno 5 V, što može postizati jedino u slučaju kada je izvor svjetlosti direktno usmjeren prema panelu.





**Slika 17** Uređaj sa postavljenim solarnim panelom i digitalnim voltmetrom

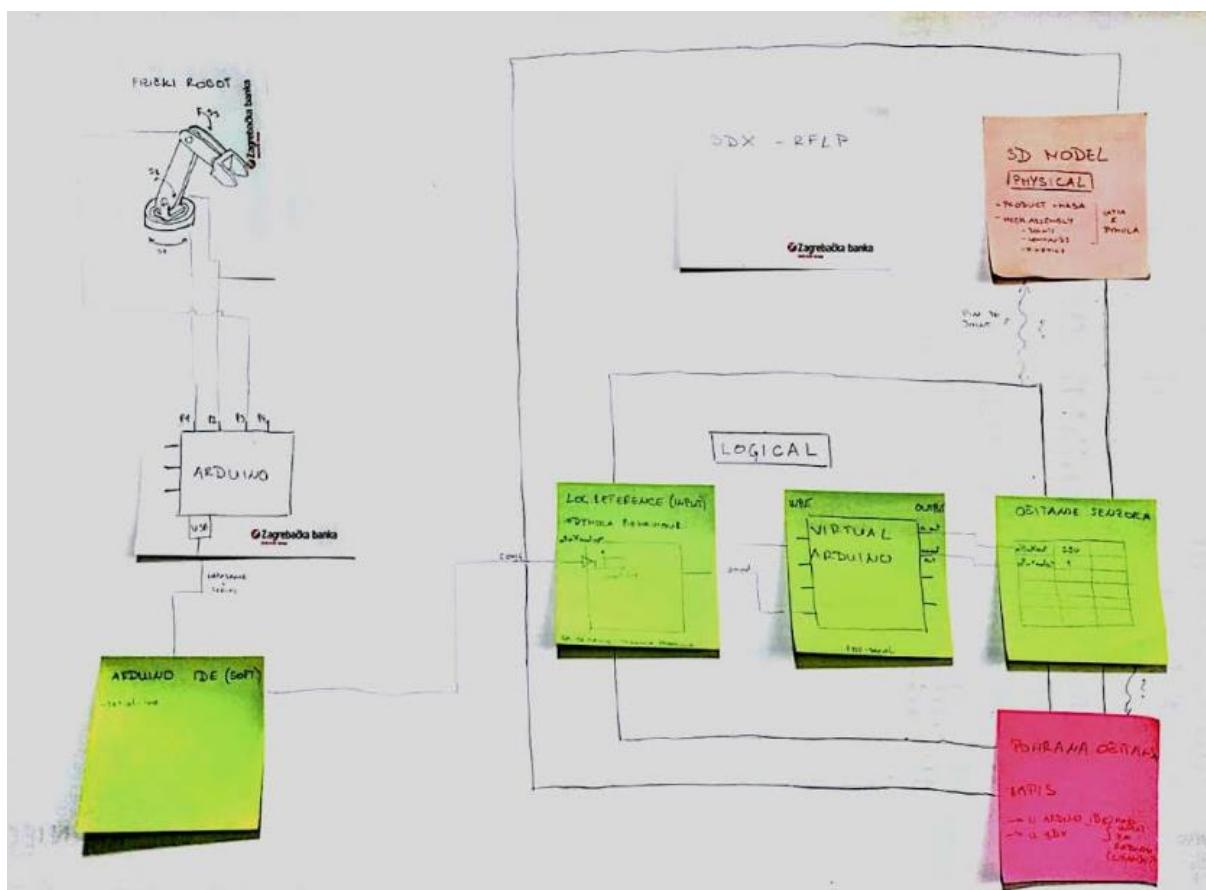
### 3.4. Upravljanje

Ponašanje uređaja definirano je programskim kodom napisanim u Arduino IDE programskom alatu. Na temelju očitavanja otpora sa svjetlosnih senzora, tj. fotootpornika određeno je gibanje servomotora na način da se solarni panel okreće prema mjestu najmanjeg otpora, tj. prema izvoru svjetlosti. Programski kod za upravljanje djelomično je preuzet s [14] i prilagođen izrađenom uređaju, te je u konačnici integriran u programski kod kojim se ostvaruje serijska komunikacija između fizičkog Arduina i njegove virtualne inačice na računalu. Konektori preko kojih uređaj dobiva naredbe o gibanju su digitalni konektor broj 9 za gornji servomotor i digitalni konektor broj 10 za donji servomotor, što se može vidjeti i u programskom kodu.

## 4. Modeliranje digitalnog blizanca

### 4.1. Modeliranje digitalno-fizičkog sustava

Slijedeći postupak kreiranja digitalnog blizanca kako je opisano u poglavlju 2.5. izrađen je dijagram koji prikazuje glavne fizičke i digitalne komponente sustava te tok podataka među njima [Slika 18]. Na lijevoj se strani nalaze fizičke komponente (odabrani primjer uređaja i Arduino mikrokontroler) te Arduino IDE softver u kojem je napisan kod za upravljanje uređajem, a koji ujedno omogućuje prijenos i pretvorbu signala iz fizičke u digitalnu okolinu. Na desnoj je strani prikazana digitalna okolina na 3DExperience platformi koja se sastoji od logičke radne okoline, 3D modela i prozora u kojem se prikazuju rezultati. Tim je dijagramom ugrubo opisana arhitektura sustava koja je detaljno razrađena u nastavku ovog rada.



Slika 18 Apstraktni model digitalno-fizičkog sustava

Nakon što su definirane glavne komponente sustava trebalo je osmisliti koji će se podaci prikupljati, kakvi će se senzori koristiti, u kojem će obliku biti dolazni podaci te kako ih pretvoriti u oblik pogodan za korištenje u digitalnoj radnoj okolini, kako će se vizualizirati sustav u sklopu 3DExperience platforme te kako će se podaci koristiti za daljnju analizu.

## 4.2. 3DExperience

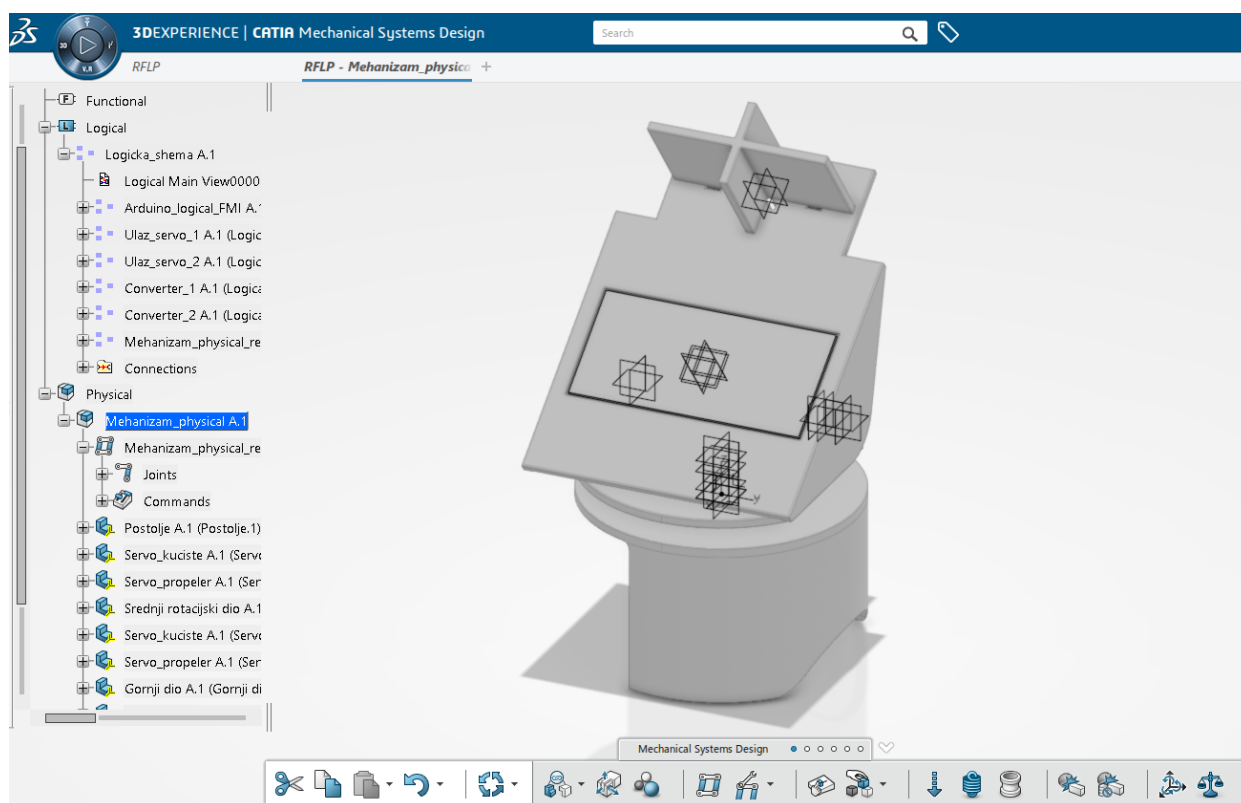
3DExperience je PLM (eng. *Product Lifecycle Management*) radna platforma koja je instalirana u računalnom oblaku (eng. *Cloud based*). Ideja na kojoj počiva je ta da se korisniku, u realnom slučaju se radi o korporacijama, omogući neometan, timski te dislocirani rad. U tu je svrhu osmišljena kao mrežna platforma na kojoj se korisnicima u skladu s dodijeljenim ulogama omogućava pristup pojedinim aplikacijama [17]. Tako korisnici kojima su dodijeljene uloge npr. Konstruktora imaju pristup aplikaciji Catia zajedno sa pripadajućim joj modulima kao što su Part Design, Assembly Design te Systems Mechanism Design, što je detaljnije opisano u nastavku rada.

U sklopu 3DExperience platforme kreirani su portali (eng. *Dashboards*) za učenje na kojima se mogu naći razni materijali u obliku prezentacija, CAD dokumenata, programskih kodova, popratnih biblioteka i slično. Za potrebe ovog rada preuzeti su neki materijali sa portala za projekte vezane uz robote i Arduino, te sa portala koji objašnjava izradu digitalnog blizanca na primjeru humanoidnog robota. Za upoznavanje sa programskim paketom Dymola za modeliranje dinamičkih sustava te radom u RFLP radnoj okolini korišteni su materijali dostupni na web stranicama 3DExperience-a, koja pruža mogućnosti personalizacije korisničkog sučelja te komunikaciju putem raznih foruma i grupa sa drugim korisnicima sličnih interesa [18].

### 4.2.1. CATIA

3D model uređaja i mehanizma napravljeni su u programskom paketu za modeliranje CATIA V6, koji se nalazi unutar 3DExperience platforme [19]. Aplikacije korištena za modeliranje dijelova uređaja je Part design, a za izradu mehanizma korištena je aplikacija Mechanical Systems Design. Sve veze među komponentama [Slika 19] navedene su u stablu pod značajkom Joints, a od pomičnih veza (u ovom slučaju skup veza pod nazivom Revolute) napravljene su kinematsko-logičke značajke sklopa navedene pod Commands. One omogućavaju rotacijsko gibanje propelera servomotora i dijelova koji su na njih vezani.

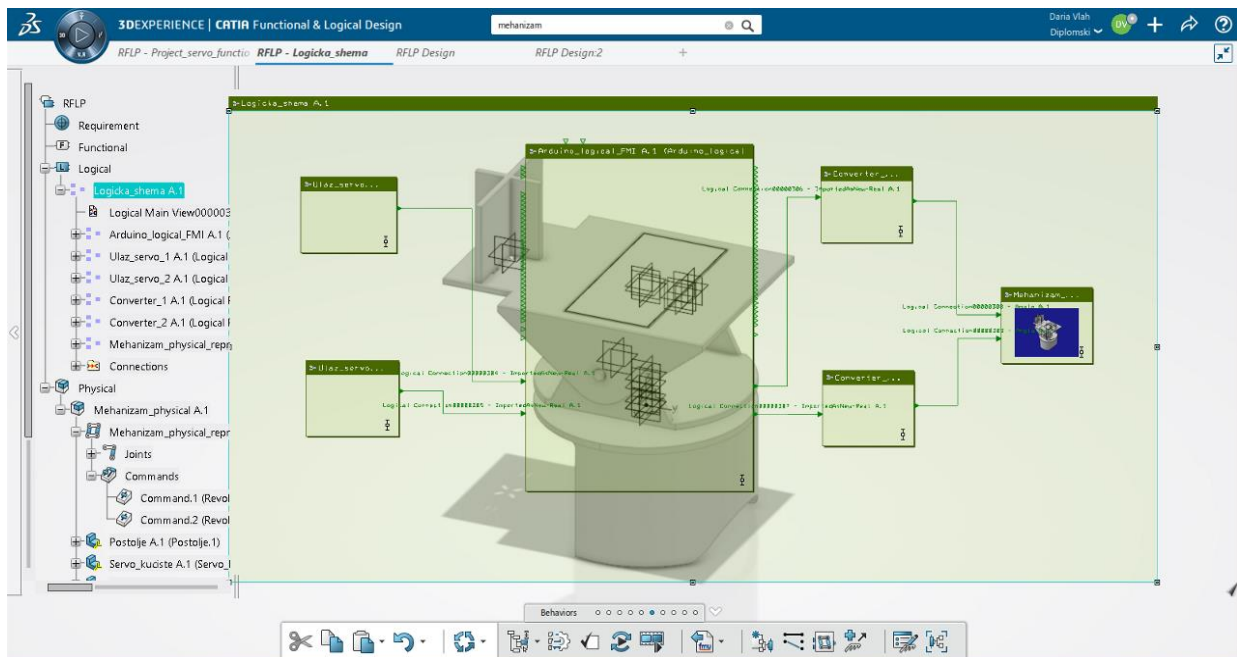




Slika 19 Catia V6 sučelje unutar 3DExperience platforme sa ekspanzivnim stablom značajki

#### 4.2.2. RFLP

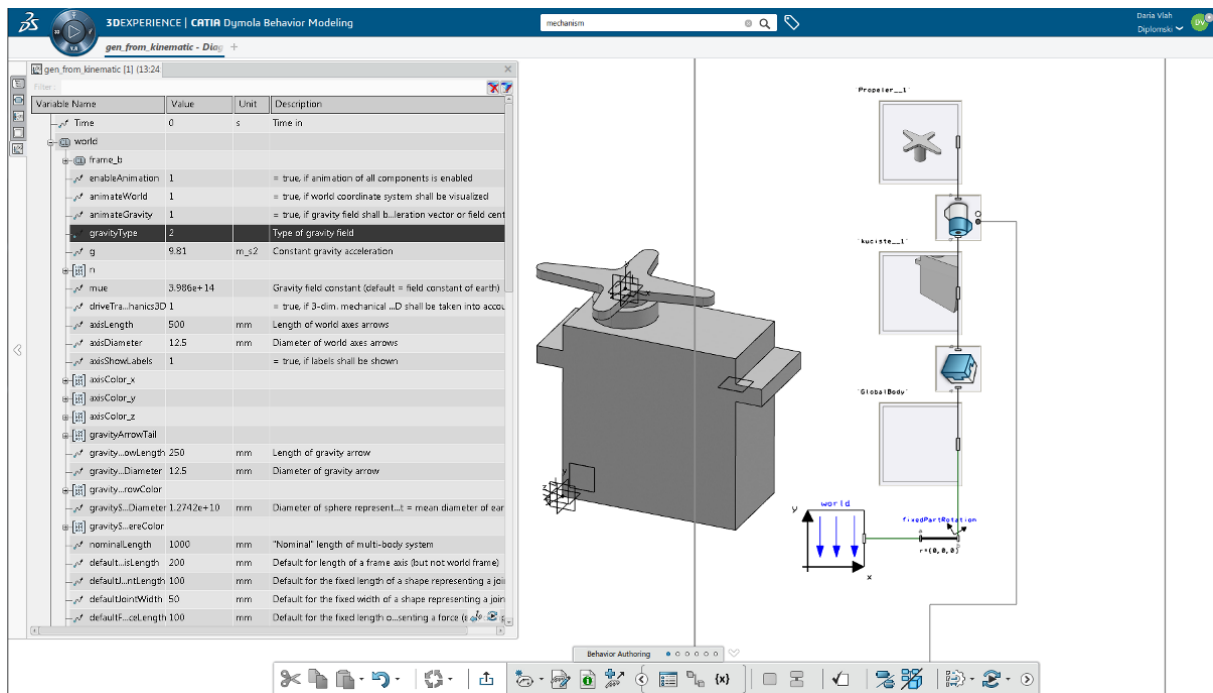
RFLP je radna okolina unutar 3DExperience platforme [Slika 20] koja služi za oblikovanje sustava koje se zasniva na modelu (eng. *Model Based Systems Engineering*) [paper11]. Slova RFLP čine kraticu koja označava 4 glavne značajke radne okoline: R predstavlja zahtjeve (eng. *Requirements*): opisuje sve zahtjeve koje sustav mora ispuniti, od zahtjeva različitih sudionika do konstrukcijskih zahtjeva; F označava funkcionalnost (eng. *Functional*): opisuje mogućnosti sustava, predstavlja arhitekturu sustava sa svim funkcijama i tokovima informacija među komponentama; L označava logičke veze (eng. *Logical*): opisuje arhitekturu komponenata sustava, njihova sučelja te alocirane funkcije i tokove; P predstavlja fizičke komponente sustava (eng. *Physical*). RFLP okolina omogućava sljedivost inženjerskih informacija od postavljanja zahtjeva do fizičkih dijelova sustava i obratno. Za izradu digitalnog blizanca uređaja korištene su logički i fizički dijelovi radne okoline.



Slika 20 RFLP radna okolina

#### 4.2.3. DYMOLA

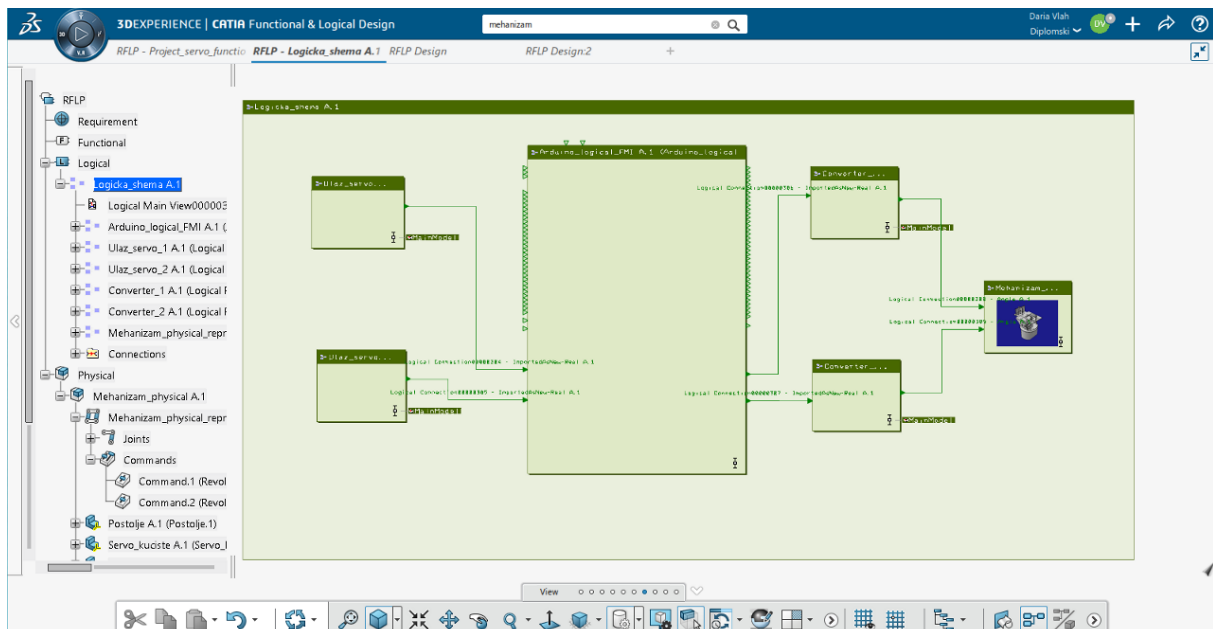
Dymola (eng. *Dynamic Modeling Laboratory*) [Slika 21] je alat za modeliranje i simulaciju integriranih i složenih sustava kakvi se javljaju u raznim granama industrije kao što su automobilska i zrakoplovna industrija, ali i robotika. Dymola se temelji na programskom jeziku za modeliranje zvanom Modelica te koristi njenu standardnu biblioteku koja ima preko 1000 generičkih modela komponenti i funkcija za različita područja [20]. U sklopu 3DExperience platforme korištena je aplikacija Dymola Behaviour Modeling pomoću koje je definirano i modelirano ponašanje pojedinih dijelova sustava i veza među njima.



Slika 21 DYMOLA radna okolina

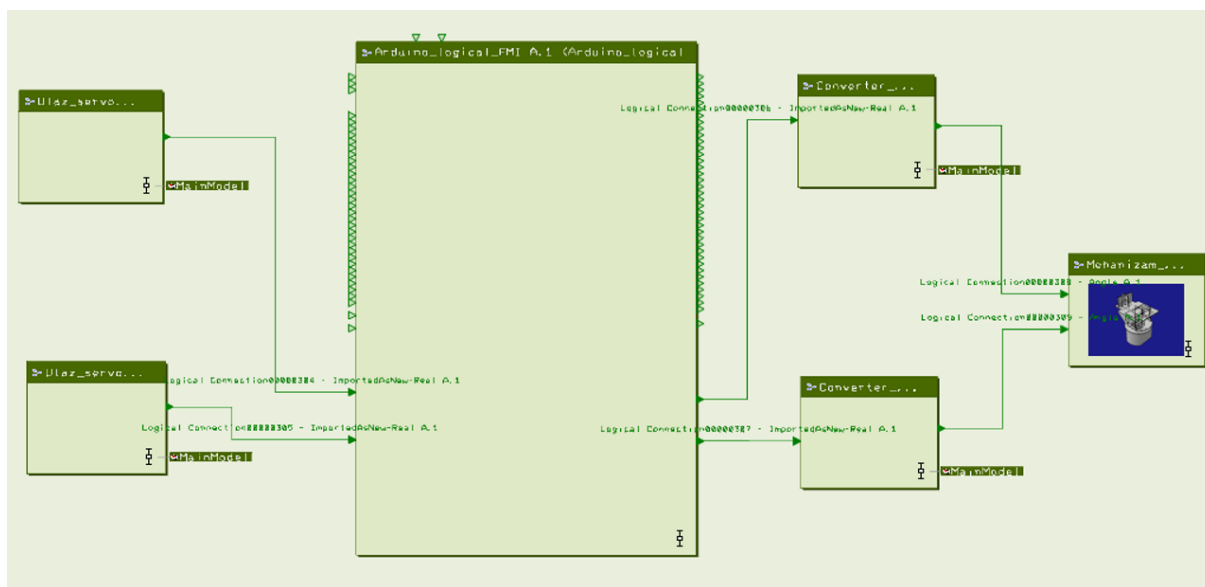
### 4.3. Povezivanje uređaja i virtualnog modela na 3DEXperience platformi

#### 4.3.1. Logička shema uređaja



Slika 22 RFLP logička radna okolina

Povezivanje uređaja i 3D modela mehanizma izvedeno je u 3DExperience logičkoj radnoj okolini (RFLP). [Slika 22] U njoj se nalaze zasebne logičke jedinice (eng. *Logical reference*) koje definiraju ponašanje pojedinih dijelova sustava i logičke veze koje su uspostavljene među njima. Slika 23 prikazuje uvećanu logičku shemu realiziranog uređaja.



Slika 23 Logička shema realiziranog uređaja

#### 4.3.2. Logička jedinica Arduino mikrokontrolera

Fizički uređaj povezan je sa svojim digitalnim blizancem preko Arduino mikrokontrolera i njegove odgovarajuće logičke jedinice u virtualnoj okolini 3DExperience platforme. Logička jedinica koja predstavlja Arduino u RFLP okolini je uvezena komponenta u obliku FMI (eng. *Functional mock-up interface*), zasebne funkcijske jedinice. FMI definira standardizirano sučelje za različite virtualne modele proizvoda koji se tako definirani mogu koristiti u računalnim simulacijama čime se olakšava modeliranje složenih digitalno-fizičkih sustava. FMI za Arduino UNO preuzet je sa stranica 3DExperience platforme, a na logičkoj je shemi ta jedinica nazvana *Arduino\_logical\_FMI*. Na sebi ima označene ulazne i izlazne konektore oblika trokuta usmjerenog prema ili u suprotnu stranu od logičke jedinice, ovisno o kakvom se konektoru radi. Svaki konektor koji je na fizičkom Arduino označen posebnim brojem za digitalne i analogne signale, na logičkoj jedinici ima 4 komponente: 2 ulazne (eng. *input*) i dvije izlazne (eng. *output*), od kojih jedna predstavlja vrstu signala koju definira mod (npr. *aPin4modinp In*), a druga brojčanu vrijednost tog signala (npr. *aPin4inp In*). Svi signali koji

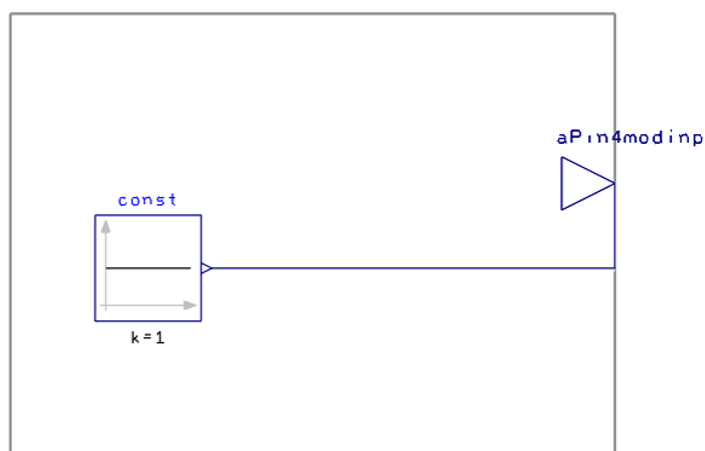
ulaze i izlaze na konektore na fizičkom Arduinou očitavaju se i na njegovoj virtualnoj kopiji, no da bi se omogućila komunikacija između fizičkog i virtualnog prvo je potrebno u RFLP radnoj okolini definirati kakvi će signali (digitalni ili analogni) ulaziti i izlaziti na kojim konektorima [Slika 24].

aPin4modinp	In	aPin4modinp aPin4r
aPin4inp	In	aPin4inp aPin4inp In
aPin4modout	Out	aPin4modout aPin4r
aPin4out	Out	aPin4out aPin4out C

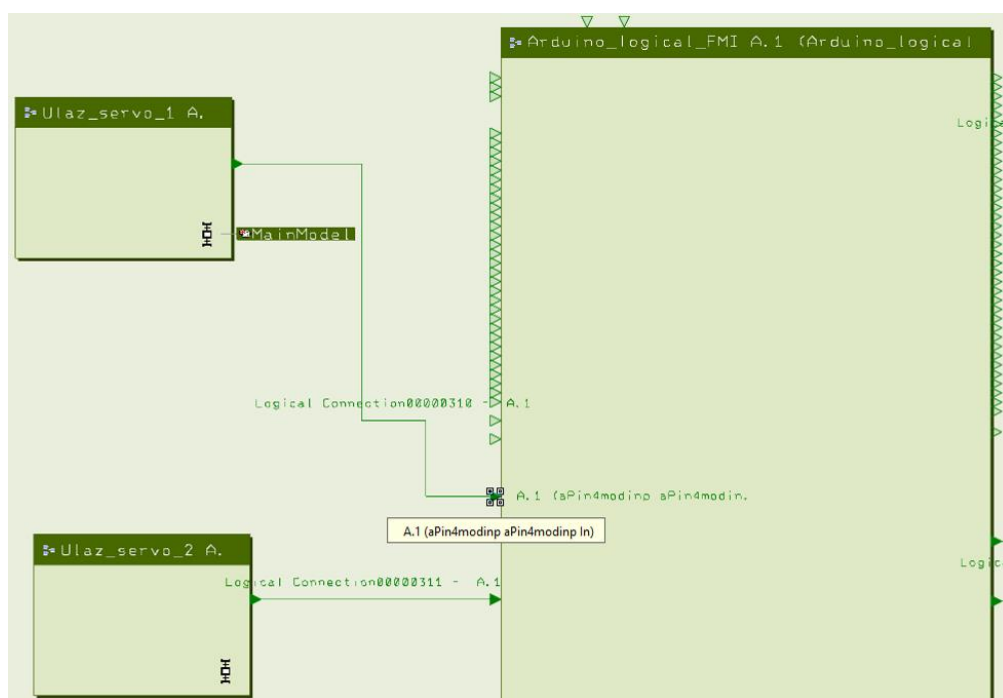
Slika 24 Četiri logičke komponente vezane uz jedan fizički konektor

#### 4.3.3. Ulazne logičke jedinice

Dvije logičke jedinice koje se nalaze lijevo na logičkoj shemi definiraju ulazne podatke koji se šalju sa servomotora na Arduino mikrokontroler. Sukladno tome njihovi nazivi su Ulaz\_servo\_1 za dolazne signale sa donjeg servomotora te Ulaz\_servo\_2 za one sa gornjeg servomotora. Slika 25 prikazuje kako izgleda definicija jedne od ulaznih logičkih jedinica, čije je ponašanje modelirano u Dymola Behaviour Modeling aplikaciji. Druga ulazna jedinica modelirana je na isti način, a razlikuje se jedino u broju konektora na koji se šalje signal. Želi li se konektor na koji se šalje signal definirati tako da prima analogni signal, potrebno je konstantu  $k$  postaviti na 1 te povezati sa konektorom aPin4Mod Inp. Ulazna jedinica Ulaz\_servo\_1 ima logičku vezu sa analognim konektorom aPin4 Inp, a jedinica Ulaz\_servo\_2 sa analognim konektorom aPin5 Inp.



Slika 25 Ulazna logička jedinica



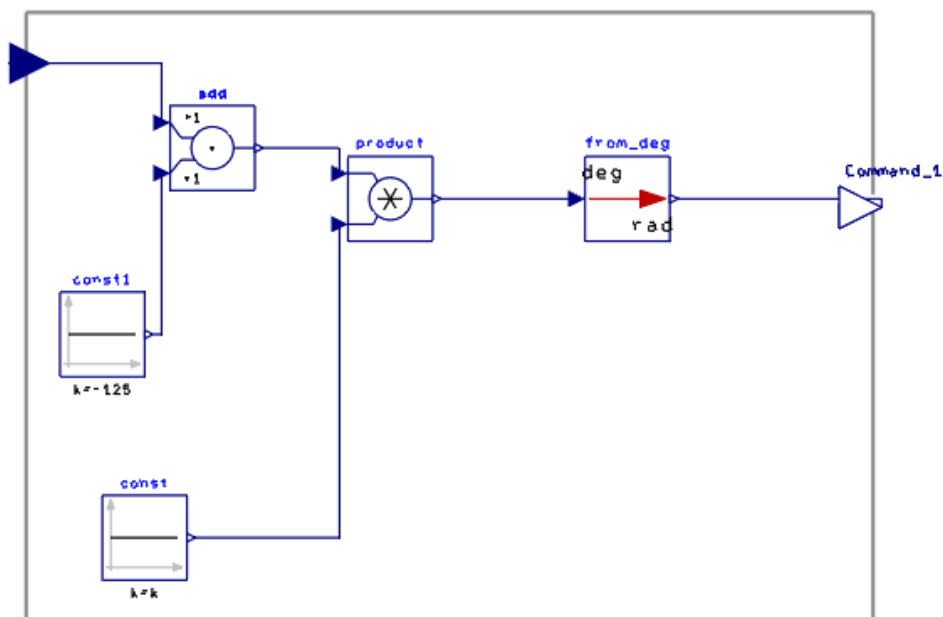
Slika 26 Povezivanje ulaznih logičkih jedinica i jedinice logičkog Arduina

Povezivanjem dviju ulaznih jedinica sa logičkom jedinicom Arduino\_logicki\_FMI [Slika 26] signali se učitavaju na ulazne konektore i prenose na izlazne konektore virtualnog Arduina aPin4 Out i aPin5 Out, što se može vidjeti prilikom pokretanja simulacije u novom prozoru gdje se prikazuju trenutne vrijednosti parametara koji se mijenjaju tokom izvođenja simulacije [Slika 27].

Name	Value
A.1 (aPin3out aPin3out Out)	0
A.1 (aPin4modout aPin4modout Out)	0
A.1 (aPin4out aPin4out Out)	0
A.1 (aPin5modout aPin5modout Out)	0
A.1 (aPin5out aPin5out Out)	0
Ulaz_servo_1 A.1 (Logical Reference00000396.1)	
MainModel	
A.1 (aPin4modinp aPin4modinp Out)	1
Ulaz_servo_2 A.1 (Logical Reference00000397.1)	
MainModel	
A.1 (aPin5modinp aPin5modinp Out)	1
Converter_1 A.1 (Logical Reference00000398.1)	
Converter_2 A.1 (Logical Reference00000399.1)	
Mehanizam_physical_representation ..._representation A.1_Logical11023.1)	
Physical	

Slika 27 Prikaz trenutnih vrijednosti očitavanih parametara

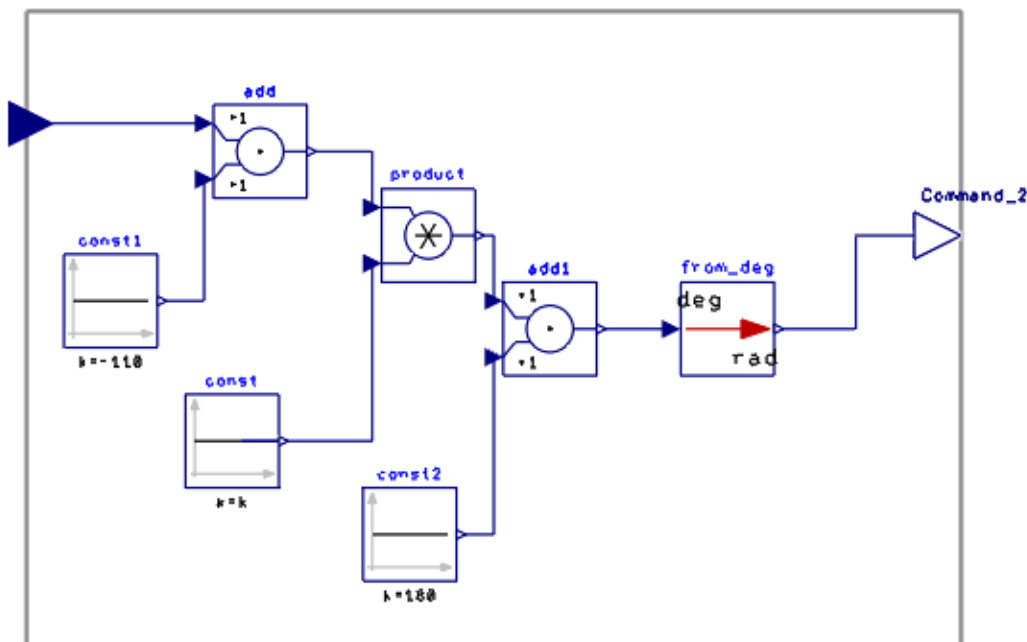
#### 4.3.4. Logičke jedinice za pretvorbu varijabli



Slika 28 Shematski prikaz logičke jedinice za pretvorbu i normizaciju signala Converter 1

Signali na izlazu iz Arduinove logičke jedinice imaju realni tip varijable, no nemaju definiranu pripadajuću mjernu jedinicu zbog čega ih nije moguće direktno povezati kao ulazne signale (vrijednost kuta zakreta, u radijanima) na 3D model. Iz tog su razloga napravljene dvije nove logičke jedinice za pretvorbu tipa varijabli nazvane Converter\_1 [Slika 28] i Converter\_2 [Slika 29], svaka za odgovarajući servomotor. Njihovo je ponašanje također modelirano u Dymola Behaviour Modeling aplikaciji. Komponente su im slične, a razlika je u kalibraciji servomotora i radnom području koji je definirano za svaki. Ulazna komponenta (plava strelica) označava ulazni signal koji dolazi sa logičke jedinice Arduina. To je brojčana vrijednost analognog očitano signala, napona, koji u ovoj primjeni služi kako bi se izmjerio kut zakreta servomotora. Nema mjernu jedinicu i nije kalibriran zbog čega je dodana komponenta „const1“. Ona predstavlja operaciju zbrajanja ulazne veličine i nove konstante  $k$ , čime se dobiva željeno mjerno područje (kut od 0 do 180 stupnjeva za donji servomotor ili nešto manje područje za gornji). Iduća komponenta je „product“, što označava operaciju množenja. Ona ulaznom broju pridružuje mjernu jedinicu stupnjeve („deg“) čime je ulazni kut potpuno definiran. Za ulazni signal u mehanizam 3D modela potrebno je imati kut u radijanima, stoga je dodana komponenta „from\_deg“ koja pretvara kut u stupnjevima u radijane. Time je ulazni signal u mehanizam

definiran (Command 1) i može se napraviti logička veza između Convertera i logičke jedinice 3D modela.

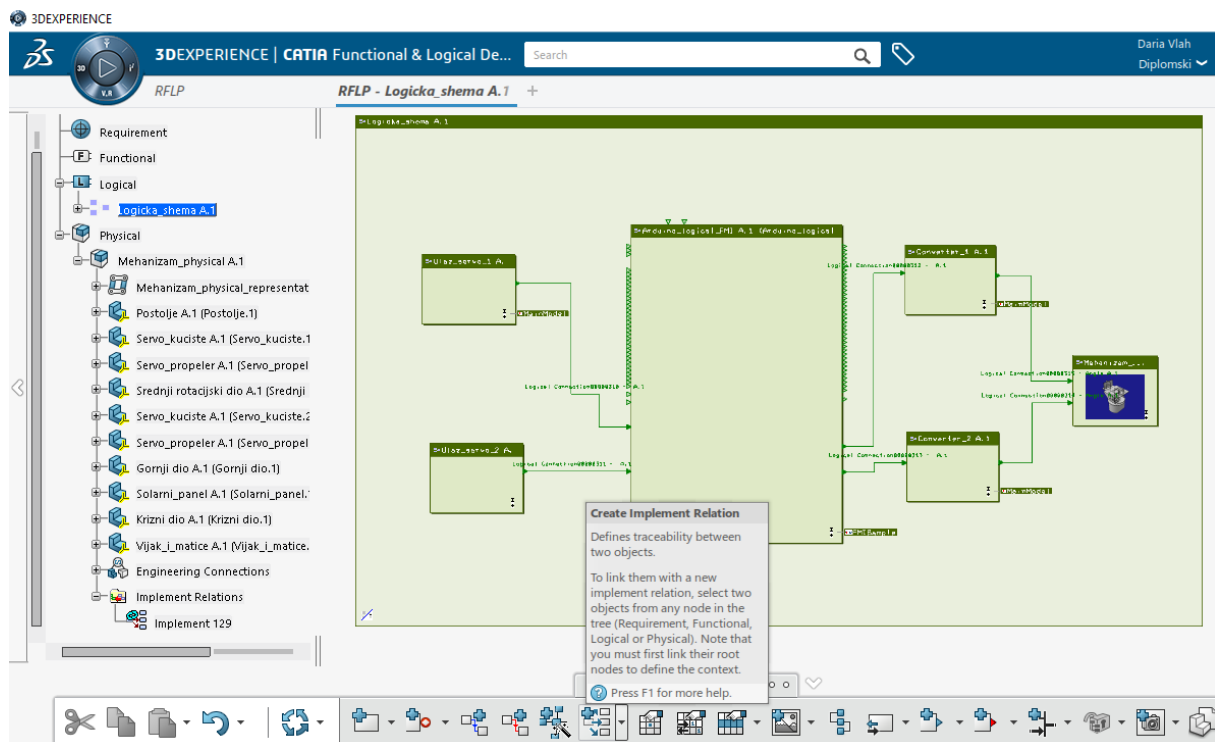


Slika 29 Shematski prikaz logičke jedinice za pretvorbu, normizaciju i modifikaciju područja očitavanja signala Converter 2

#### 4.3.5. Logička jedinica 3D modela mehanizma

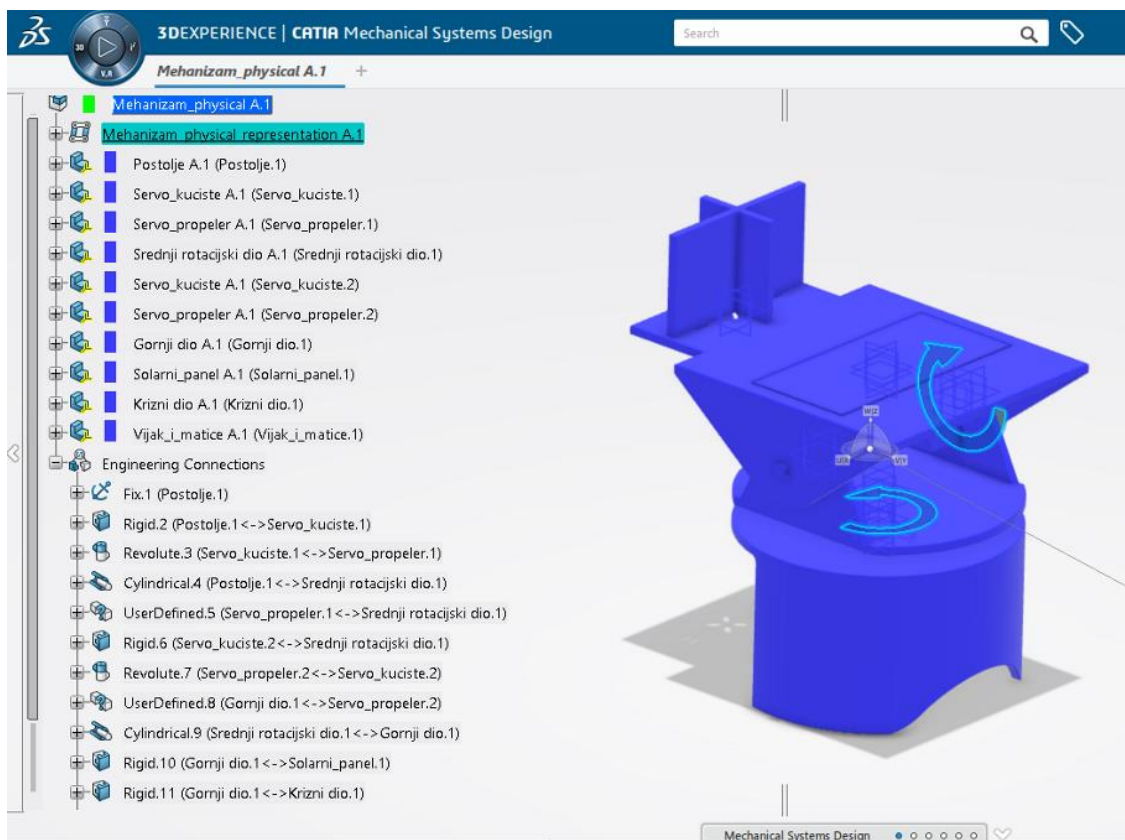
Za povezivanje 3D modela uređaja sa logičkom okolinom potrebno je napraviti odgovarajuću logičku jedinicu u koju se mogu unositi ulazni signali za kut zakreta servomotora. Logička jedinica koja predstavlja mehanizam može se napraviti na dva načina, jedan je dodavanje ponašanja kreiranog iz kinematike u Dymola-i, a drugi način, koji je primijenjen u ovom radu je mapiranje logičkog i fizičkog modela. Da bi se provelo mapiranje prvo je potrebno u radnu okolinu uvesti fizičku komponentu, tj. 3D model uređaja i povezati ga sa glavnom logičkom jedinicom (Logicka\_shema) tako da se kod pokretanja simulacije uvijek učitavaju zajedno. Učitavanje mehanizma provodi se tako da se u fizički dio radne okoline (eng. *Physical*) učita već postojeći model koji je izrađen u Catia-i. Povezivanje fizičkog i logičkog modela ostvaruje se naredbom *Implement relations*, gdje se označava željena logička jedinica i mehanizam. [Slika 30]



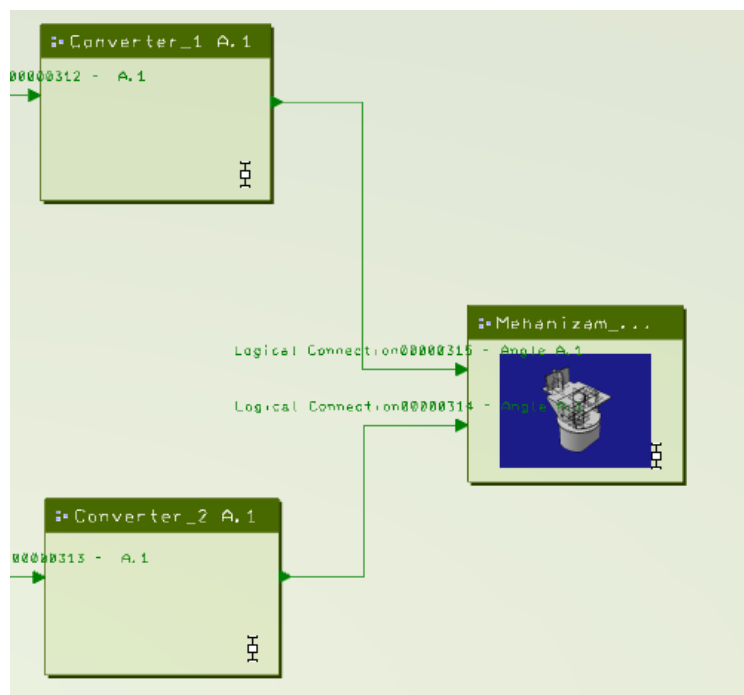


**Slika 30** Povezivanje fizičkog i logičkog modela naredbom **Implement relations**

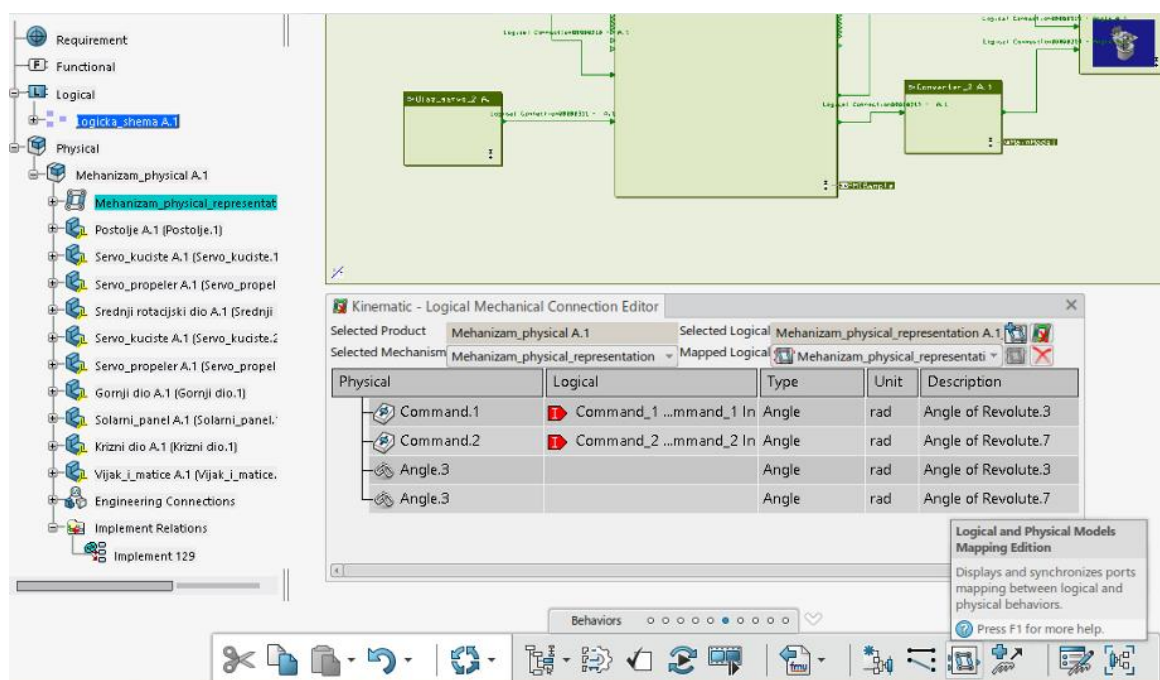
3D model mora biti sastavljen u funkcionalni mehanizam te sadržavati značajke koje omogućuju gibanje komponenti [Slika 31]. Na temelju definiranih kretanja unutar mehanizma mapiranjem se kreira nova logička komponenta za odgovarajućim ulaznim logičkim konektorima.[Slika 32] Na odabranom primjeru kretanje mehanizma definiraju rotacijska gibanja servomotora, a logički konektor vezan na naredbu Revolute kao ulazni signal zahtjeva kut zakreta u radijanima [Slika 33]



Slika 31 Mehanizam sa pripadajućim stablom značajki

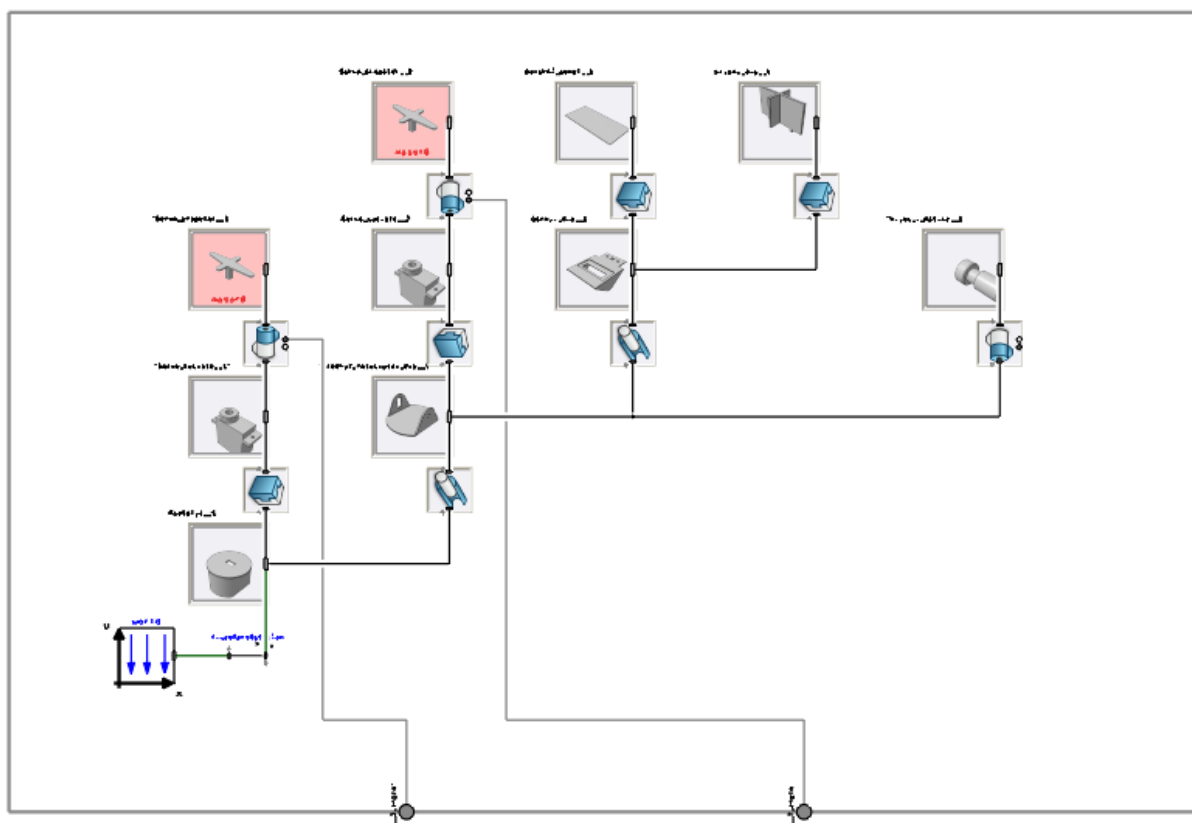


Slika 32 Povezivanje pretvorbenih logičkih jedinica i logičke jedinice mehanizma



**Slika 33 Kreiranje logičke jedinice mehanizma postupkom mapiranja**

Mapiranjem se dobiva direktna veza na željenu naredbu, no tako nije moguće vidjeti ponašanje prikazano unutar logičke komponente. Slika 34 prikazuje kako izgleda ponašanje kreirano iz mehanizma u Dymola-i, gdje se može jasno vidjeti kako su komponente 3D modela međusobno logički povezane te na koje značajke se vežu ulazni signali. Ovakav prikaz dan je radi pojašnjenja onoga što se događa unutar logičke jedinice koja predstavlja mehanizam, iako samim provođenjem mapiranja ponašanje te jedinice nije moguće vidjeti. Dvije točke na dnu predstavljaju mjesta za primanje signala, tj. mjesta na kojem se u mehanizam unose kutovi zakreta servomotora. Linije koje njih povezuju s mehanizmom vode do grafičkog prikaza naredbi Revolute koja predstavlja odnos između dvije komponente koje su prikazane ispred i iznad nje. Na sličan su način prikazane i ostale značajke koje u modelu povezuju različite komponente, ali te značajke predstavljaju krute veze stoga ne zahtijevaju nikakve ulazne parametre.



**Slika 34** Logička reprezentacija mehanizma

Nakon što su kreirane sve logičke jedinice : ulazne jedinice, Arduino\_logicki\_FMI, Converteri i logička reprezentacija mehanizma, potrebno je povezati sve odgovarajuće konektore logičkim vezama. Veze se kreiraju dvoklikom miša prvo na ulazni konektor, zatim na izlazni. Ako su ulazne i izlazne varijable ispravno definirane, logička veza će se moći kreirati. Time je završena izrada logičkog modela uređaja u 3DExperience radnoj okolini.

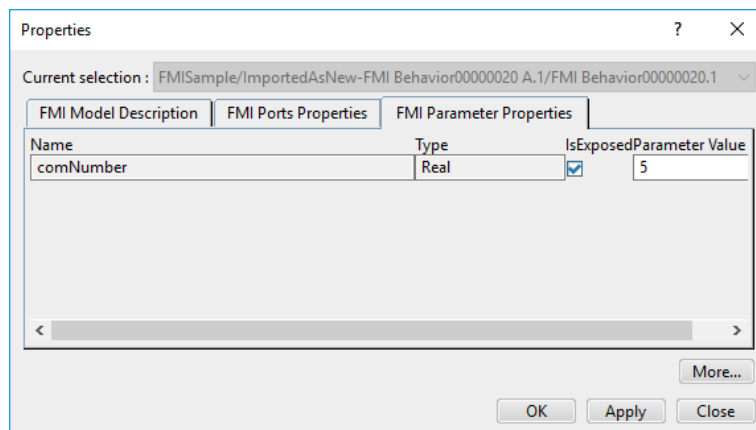
#### 4.4. Serijska komunikacija

Sada je još preostalo uspostavljanje serijske komunikacije između Arduino mikrokontrolera, signala koje on prima i šalje na svojim konektorima i računala, tj. logičke okoline na 3DExperience platformi. Za ostvarivanje serijske komunikacije potrebno je pokrenuti programski kod, koji je preuzet sa 3DExperience platforme za učenje, u Arduino IDE programskom alatu. Njegovim pokretanjem omogućuje se prijenos i čitanje signala sa fizičkog na virtualni Arduino. Time se postiže prijenos informacija iz fizičke okoline u virtualnu u stvarnom vremenu bez kašnjenja, čime je postignuto da je digitalni blizanac doživljava okolinu i reagira na vanjske podražaje na isti način kao i fizički uređaj, tj. kopira reakcije stvarnog

uređaja. Da bi uređaj s jedne strane bio upravlján prema očitánjima svjetlosnih senzora i očitavao podatke sa servomotora, a istovremeno preko serijske komunikacije slao signale o gibanju uređaja u virtualnu okolinu, bilo je potrebno integrirati zasebne programske kodove za upravljanje i za komunikaciju u jedan kod [Prilog II]. Signali za očitavanje vrijednosti svjetlosnih senzora šalju se na analogne konektore a0 do a3, a potrebni ulazni signali za upravljanje servomotorima šalju se na digitalne konektore d9 i d10. Za praćenje gibanja servomotora u stvarnom vremenu, signali se iz njih šalju na analogne konektore a4 i a5, te se učitavaju u virtualnoj okolini što pokreće 3D model uređaja koji tada vrši jednako gibanje kao i fizički uređaj. Time je ostvareno pravovremeno očitavanje podražaja iz okoline kako na fizički uređaj, tako i na virtualni model, ali i reakcija fizičkog uređaja na te podražaje koja se istovremeno prikazuje na njegovom digitalnom blizancu.

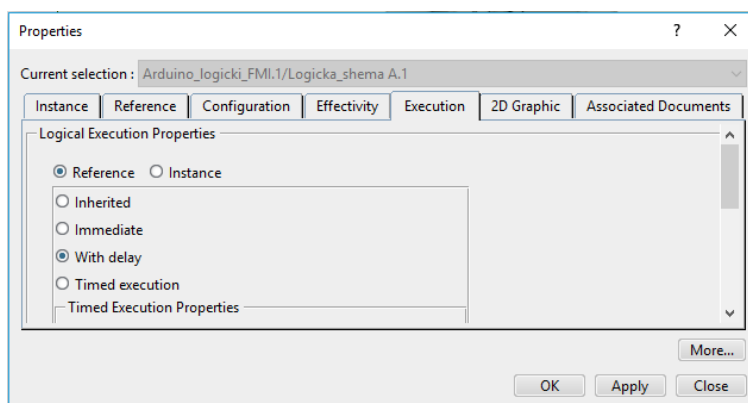
#### 4.5. Pokretanje uređaja i digitalnog blizanca

Kreiranjem komponenti i definiranjem logičke okoline na način kako je opisano u prethodnom poglavlju, te uspostavljanjem serijske komunikacije između uređaja i računala ispunjeni su svi uvjeti za realizaciju digitalnog blizanca uređaja u okvirima 3DExperience platforme. Za pokretanje uređaja i digitalnog blizanca prvo je potrebno u 3DExperience platformi učitati prethodno napravljenu logičku shemu zajedn sa 3D modelom koji se veže uz nju. Zatim je potrebno otvoriti programski kod za upravljanje i komunikaciju u Arduino IDE programskom alatu te učitati program na Arduino mikrokontroler. Prije pokretanja simulacije unutar logičke okoline potrebno je provesti inicijalizaciju Arduino\_logicki\_FMI logičke jedinice tako da se postavi odgovarajući COM port na koji je fizički Arduino povezan s računalom preko USB kabela [Slika 35].



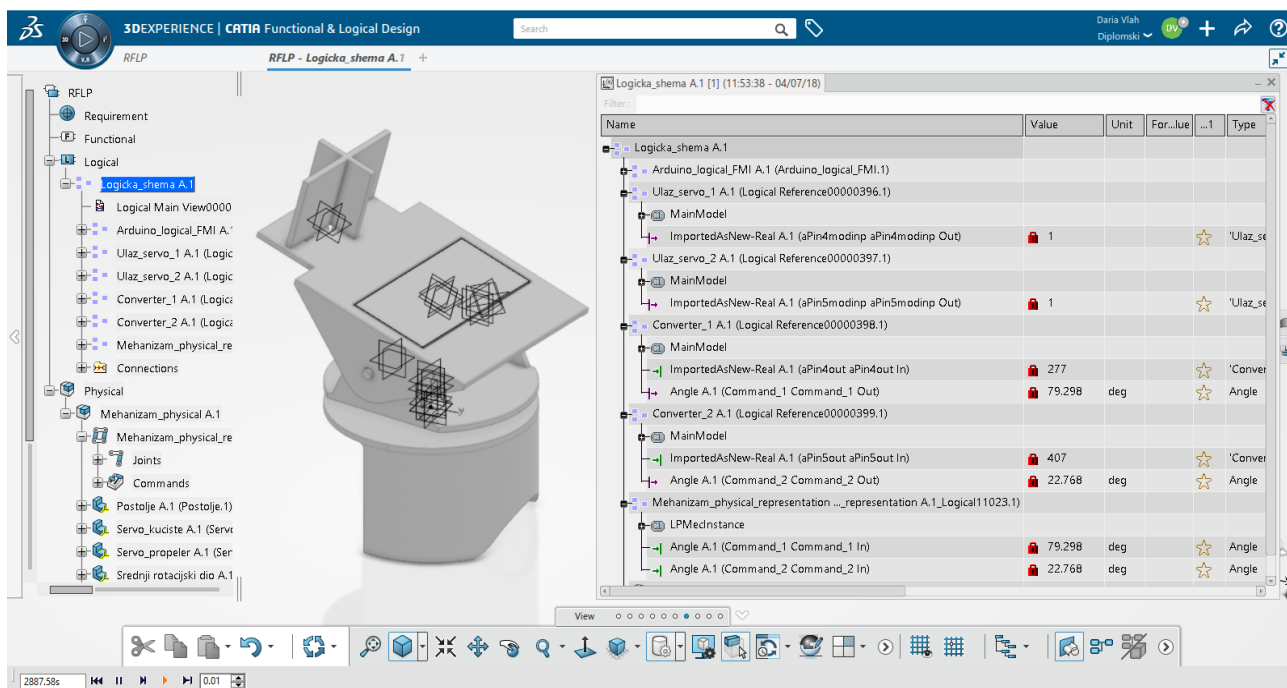
Slika 35 Definiranje ulaznog USB porta

U postavkama koje se tiču izvođenja simulacije za Arduino\_board logičku jedinicu treba označiti opciju da se reakcija bloka izvodi sa malom vremenskom zadržkom (eng. with delay) [Slika 36], jer inače nije moguće provesti inicijalizaciju i pokrenuti simulaciju. Kad su sve postavke namještene, potrebno je pritiskom na gumb osvježiti logičku shemu čime se spremaju sve unesene promjene.



**Slika 36** Definiranje načina izvođenja simulacije uz uključenu opciju vremenske zadržke

Tada se u 3DEXperience radnoj okolini može pokrenuti simulacija čime se pokreće digitalni bliznac uređaja. Ako je sve pravilno postavljeno, pokretanjem simulacije otvara se prozor u kojem se može pratiti trenutno stanje digitalnog blizanca i vrijednosti signala koji ulaze i izlaze na konektorima logičkih jedinica, te se u pozadini vidi kretanje 3D modela koje reflektira stvarno kretanje uređaja. [Slika 37]



**Slika 37** Simulacija rada uređaja sa prikazom mjerenih podataka

## 5. Prikupljanje i analiza podataka

Industrija 4.0 je, kao što je ranije opisano, rezultira masovnim prikupljanjem podataka iz različitih izvora te njihovoj obradi u svrhu boljeg shvaćanja promatranih sustava, njihovog boljeg razumijevanja i optimizacije te kreiranja na tome zasnovanih novih vrijednosti. Kako je u sklopu izrade ovog rada naglasak stavljen na postavljanje veze između 3DExperience platforme i Arduino elektroničke platforme u ovom će poglavlju biti izložen kratak pregled i struktura podataka prikupljenih u industrijskim okruženjima te će u kratkim crtama biti opisana jedna od najraširenijih metoda obrade velikih setova podataka – neuronska mreža.

### 5.1. Veliki skupovi podataka u Industriji 4.0

Veliki skupovi podataka nastaju digitalnim prikupljanjem velike količine podataka iz svih dostupnih izvora te svoju aktualizaciju duguje sve pristupačnijem i brojnijem prostoru za pohranu [21]. Neizostavan paralelni razvoj računalne tehnologije omogućio je učinkovitiju obradu takvih skupova podataka.

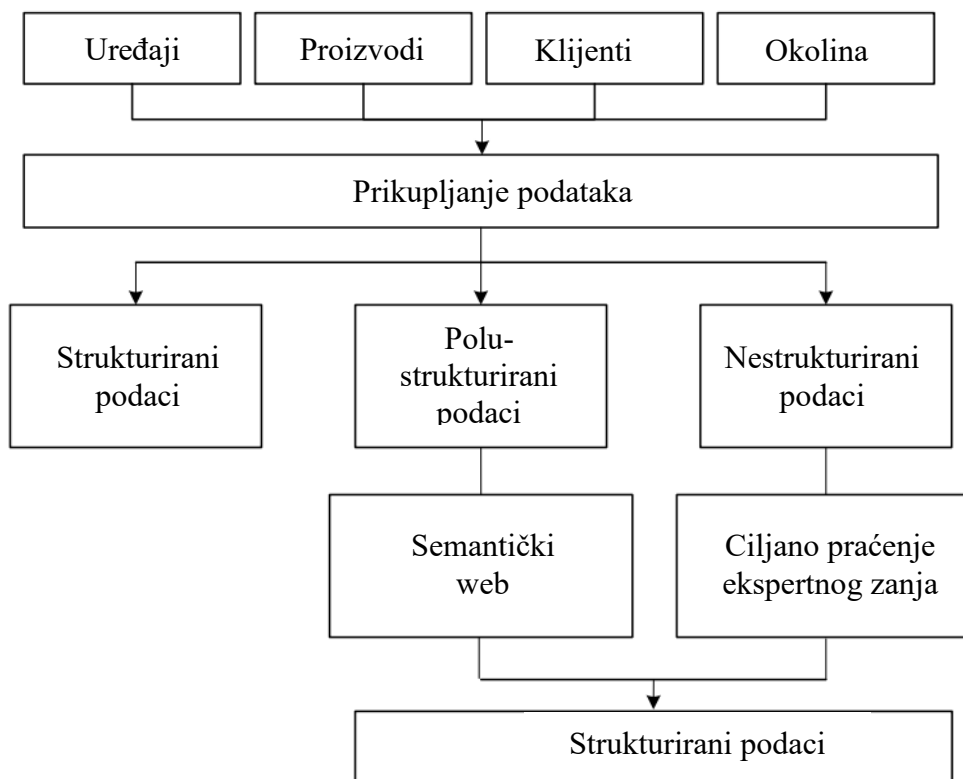
Prema [21] prikupljeni podaci u kontekstu Industrije 4.0 potječu iz velikog broja izvora kao što su: podaci vezani uz konstruiranje i projektiranje proizvoda; podaci vezani za upravljanje strojevima iz kontrolnih sustava i upravljačke opreme; podaci vezani za ponašanje i performanse radnika u obliku izvješća ili videozapisa; podaci vezani za cijenu koštanja npr. proizvodnih procesa i operacija; podaci iz domene logistike; podaci o okolišu kao što su vremenski uvjeti, temperatura, vlažnost i buka u radnom okruženju; podaci vezani uz praćenje procesa i nepravilnost u radu; podaci o kvaliteti u vidu udjela nesukladnosti unutar procesa; podaci vezani za eksploataciju kao što su učestalost popravaka; podaci od strane direktnih korisnika u vidu povratnih informacija, sugestija i tako dalje.

Razmatrajući tako podatke prikupljene kroz cjelokupni životni ciklus proizvoda zaključuje se da se prikupljeni podaci mogu podijeliti u tri glavne skupine:

1. Strukturirani podaci – signali sa senzora, podaci s upravljačkih jedinica i tako dalje
2. Polu-strukturirani podaci – podaci vezani uz korisničku podršku preuzeti s internetskih servisa u XML formatu
3. Nestrukturirani podaci – zvukovni, slikovni i video zapisi

Kako bi se omogućila obrada takvog skupa podataka potrebno je najprije izvršiti karakterizaciju sirovih podataka (eng. *raw data*) [Slika 38]. Karakterizacija podataka je pojam koji označava

generiranje deskriptivnih parametara koji se koriste kako bi pobliže opisali karakteristike i ponašanje određenog podatka [21].



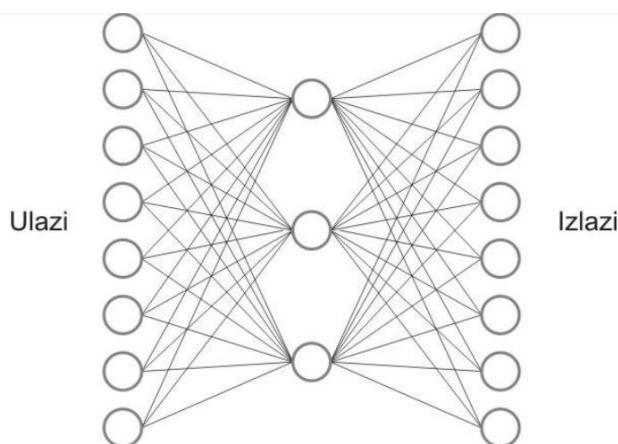
**Slika 38** Shematski prikaz prikupljanja i strukturiranja podataka [21]

U kontekstu ovog rada zanimljivo je razmotriti mogućnost korištenja prikupljenih podataka u prediktivne svrhe, na primjer predviđanja količine prikupljene električne energije u definiranom periodu pod prognoziranim vremenskim uvjetima. Poželjna karakteristika opisanog modela je kontinuirano unapređivanje modela temeljeno na podacima prikupljenima kroz cjelokupan radni vijek.

Kao logičan odabir prediktivne metode nameće se model neuronskih mreža. Neuronske mreže su skup umjetnih neurona koji su u međusobnoj interakciji kroz operacije obrade signala te posjeduju sposobnost korištenja iskustvenog znanja [23].

Model rada neuronskih mreža postavljen je po uzoru na ljudski mozak te su uređene na način da mogu imati samo jedan ili niz ulaza za samo jedan izlaz između kojih se nalazi jedan ili više sakrivenih slojeva [Slika 39].





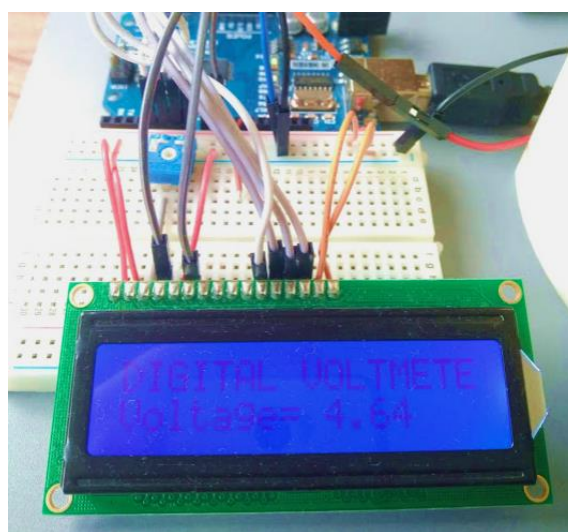
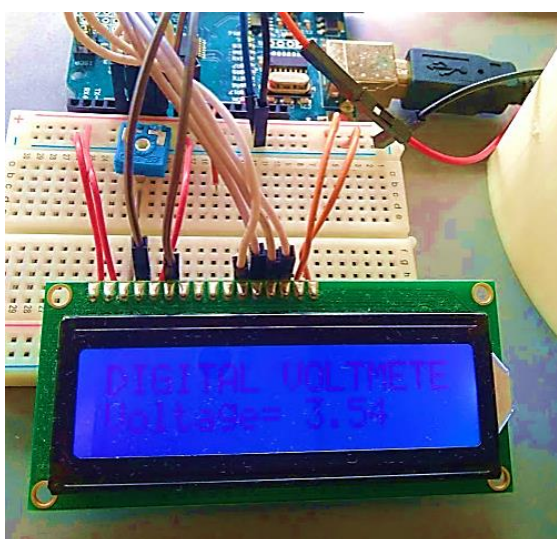
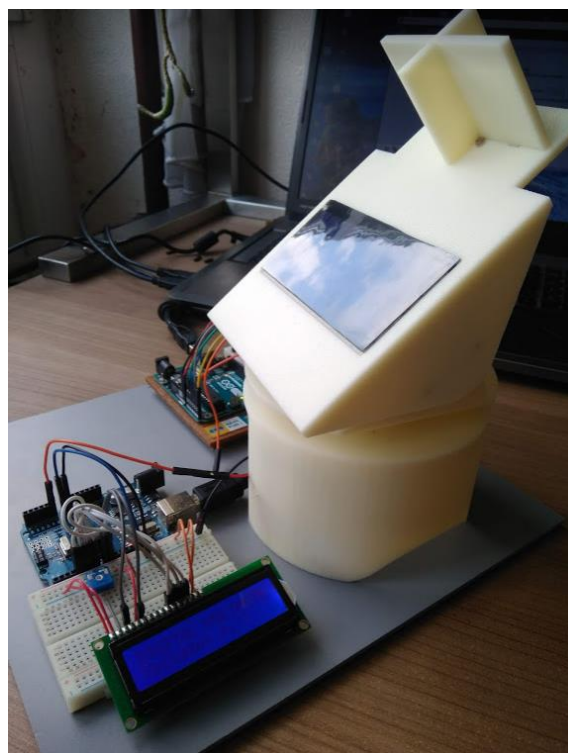
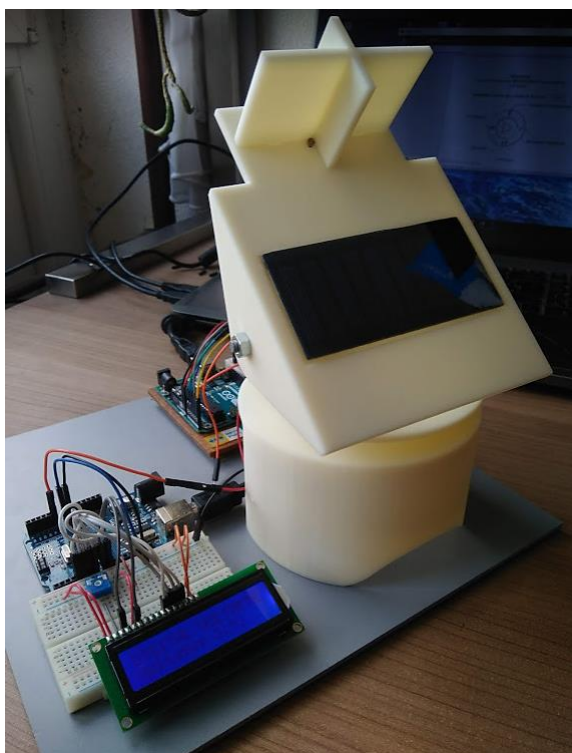
**Slika 39 Prikaz pojednostavljenog modela neuronske mreže [23]**

Glavna primjena neuronskih mreža je u određivanju zavisnosti među podacima koji nisu nužno u linearnoj vezi. Kao što je i ranije spomenuto glavno svojstvo neuronskih mreža je sposobnost učenja u obliku uvježbavanja kroz niz ponavljajućih postupaka analize. Od skupa podataka koje neuronska mreža obradi većina služi za proces učenja, odnosno uvježbavanja, a manji dio za samo previđanje.

## 5.2. Analiza podataka na izvedenom primjeru

Zbog uskih vremenskih okvira predviđenih za izradu ovog rada detaljnija obrada analize nije provedena te ovo poglavlje predstavlja temelje za daljnji razvoj razrađenog sustava.

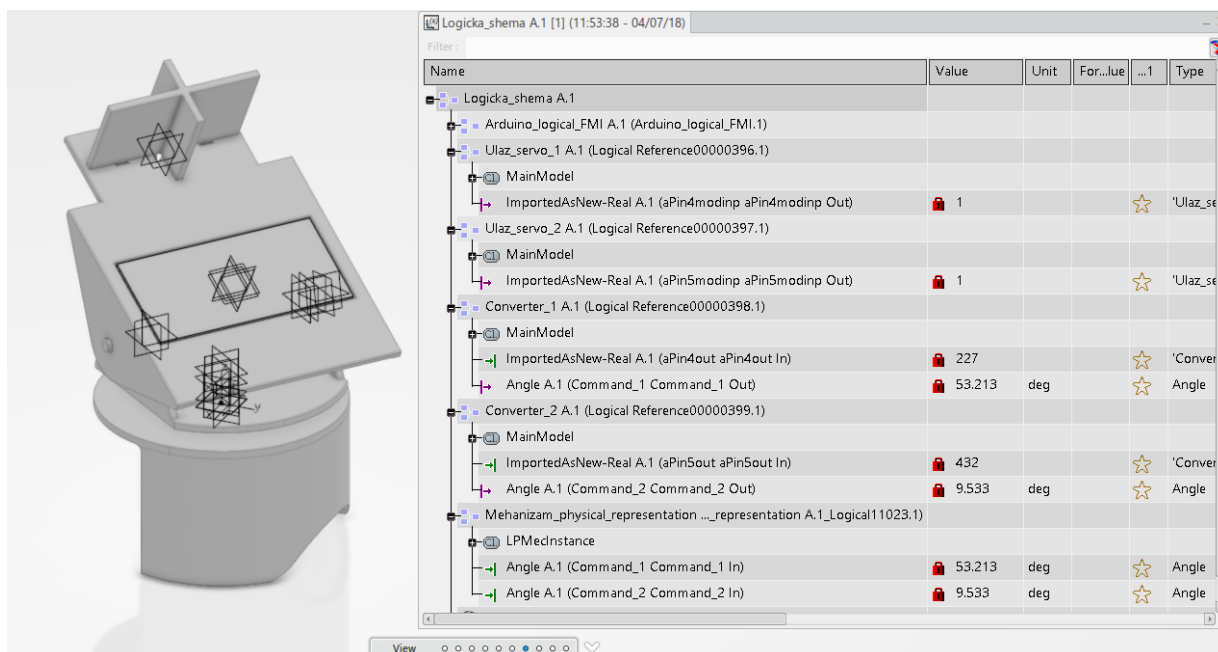
U radu razmatrani sustav sastoji se od solarnog panela koji, kako je ranije opisano, služi za pretvorbu energije zračenja sunca u električnu energiju. Dobivena se energija može pohranjivati pomoću baterijskog sustava za kasniju upotrebu. Ranije je opisano kako energija dobivena korištenjem solarnih panela ovisi i o upadnom kutu sunčevih zraka na površinu solarnih ćelija te je najviše sunčeve energije moguće pretvoriti u električnu upravo kada taj kut iznosi  $90^\circ$ . Spomenuti je način jedan od mogućih putem kojih se može utjecati na iskoristivost solarnog panela što potvrđuje i eksperimentalno mjerenje napona pomoću uređaja realiziranog za potrebe izrade ovog rada. Slika 40 a) prikazuje uređaj u položaju kad solarni panel nije usmjeren prema izvoru svjetlosti. Na LCD ekranu može se vidjeti očitavanje iznosa napona u tom slučaju. Slika 41 prikazuje digitalnu inačicu uređaja u tom trenutku. U posebnom prozoru mogu se vidjeti očitavanja kutova servomotora koja definiraju položaj 3D modela.



a) Uređaj usmjeren od sunca

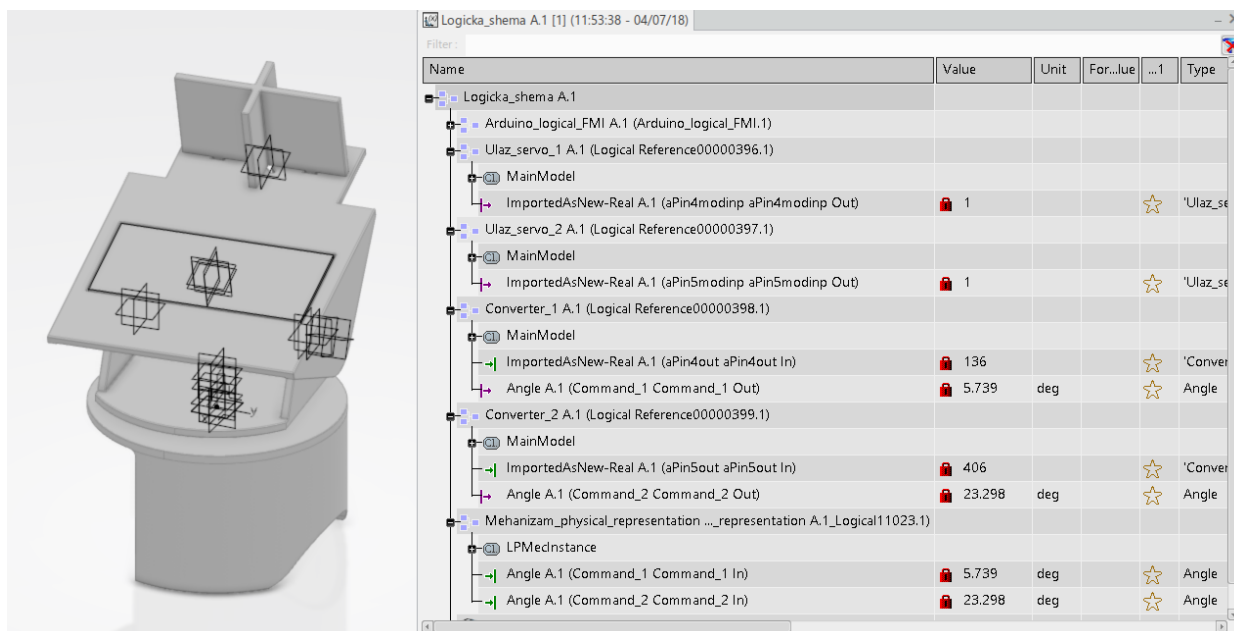
b) Uređaj usmjeren prema suncu

**Slika 40** Različiti položaji uređaja naspram izvora svjetlosti



**Slika 41 Digitalni blizanac uređaja u trenutku kad je usmjeren od izvora svjetlosti**

Slika 40 b) i Slika 42 prikazuju uređaj i njegov digitalni prikaz kad je u njega učitani programski kod za upravljanje koji se automatski okrenuo prema izvoru svjetlosti. Ova mjerenja jasno pokazuju da solarni panel može proizvesti više energije ako je usmjeren direktno prema suncu.



**Slika 42 Digitalni blizanac uređaja u trenutku kad je usmjeren prema izvoru svjetlosti**

Svrha digitalnih blizanaca je prikupljanje podataka o radu uređaja i uvjetima koji vladaju u radnoj okolini te da se ti podaci mogu koristiti za razne simulacije, analize i predviđanja s ciljem optimizacije trenutno aktualnih proizvoda i procesa, kao i daljnjeg razvoja novih generacija

proizvoda i usluga. Ovakav se uređaj može koristiti za predviđanje količine energije koja se može generirati i pohraniti u nekom periodu u određenim vremenskim uvjetima.

Primjena takvih uređaja moguća je na prijevoznim sredstvima te pokretnim objektima kao što su brodovi, kamperi, sateliti i sl. koji prilikom životnog vijeka prolaze kroz razne vremenske uvjete o kojima ovisi i količina dobivene energije. Na takvom je sustavima potrebno, uz intenzitet sunčeve svjetlosti mjeriti podatke o trajanju izloženosti sunčevom zračenju i temperaturi okoliša kako bi se računski mogao odrediti podatak o trenutno dobivenoj korisnosti solarnog panela.

Razvojna faza prediktivnog sustava tako bi se sastojala od tri glavna dijela. U prvoj fazi provodilo bi se mjerenje i prikupljanje podataka o radnim uvjetima kao što su geografska lokacija, temperatura, intenzitet sunčevog zračenja te broj sati izloženosti suncu te izlazni parametri sustava kao što su napon i jakost struje. Na temelju prikupljenih podataka moguće je odrediti model međuovisnosti promatranih varijabli. U drugoj se fazi pomoću podataka preuzetih sa meteoroloških servisa predviđa iskoristivost sustava u željenom vremenskom okviru te je moguće predvidjeti količinu prikupljene energije. U trećoj, iterativnoj fazi se predviđeni podaci uspoređuju sa stvarno izmjerenim podacima te služe za daljnje razvijanje modela principima umjetne inteligencije i strojnog učenja.

## 6. ZAKLJUČAK

Cilj ovog rada bila je realizacija paradigme digitalnih blizanaca u kontekstu razvoja proizvoda korištenjem 3DExperience PLM platforme.

Prvi dio rada sastojao se od osmišljavanja i izrade uređaja sa karakteristikama pametnog proizvoda koji uz pomoć senzora postaje svjestan svoje okoline te na nju može pravodobno reagirati. Pametni uređaj ostvaren je u obliku solarne platforme koja se dinamički prilagođava položaju sunca. Opremljena je sensorima osjetljivima na svjetlo čiji se izlazni podaci preko Arduino mikrokontrolera koriste za upravljanje uređajem, aktuatorima te solarnim panelom koji služi za prikupljanje energije sunčeva zračenja.

Drugi dio rada obuhvaćao je izradu digitalnog blizanca uređaja koja se sastojala od kreiranja modela sustava u logičkoj radnoj okolini 3DExperience platforme, konstruiranja 3D modela uređaja te njihovog povezivanja sa fizičkom uređajem. Uspostavljanje komunikacije između fizičkog i virtualnog proizvoda zahtijevalo je podosta truda i vremena obzirom da su ulazni podaci potrebni za pokretanje modela bili drugačijeg tipa od onih dobivenih na izlazu iz servomotora. Cijela 3DExperience platforma nalazi se na računalnom oblaku (eng. Cloud), što se nije pokazalo kao idealno rješenje zato što su se ponekad zbog slabe internetske mreže operacije odvijale prilično sporo te je katkad bilo potrebno ugaziti i ponovno pokrenuti program. Drugi nedostatak ovakvog sustava je vrlo česta potreba za ponovnom inicijalizacijom komponenti koja je nužna za uspostavu serijske komunikacije uređaja preko USB portova. Svaki puta kada bi se napravile neke promjene na logičkom modelu te se sustav osvježio, bilo je potrebno ponovo provesti inicijalizaciju. Obzirom da je 3DExperience platforma tek nedavno napravljena, sadrži zaista velik broj aplikacija i nudi razne mogućnosti, povremena nestabilnost sustava nije ništa neobično te će se zasigurno u skorije vrijeme ovakvi problemi ukloniti. U konačnici rad je rezultirao funkcionalnim modelom virtualnog blizanca odabranog uređaja koji pokretanjem simulacije reflektira ponašanje fizičkog uređaja.

Da bi koncept digitalnog blizanca bio u potpunosti iskorišten potrebno je prikupljene podatke analizirati te iz rezultata analize izvući korisne informacije za daljnju primjenu pri optimizaciji, povećanju učinkovitosti, prediktivnom održavanju, razvoju proizvoda i drugom. U ovom je radu opisano koje još parametre treba mjeriti sensorima te koje podatke preuzete sa meteoroloških servisa iskoristiti u analizi da bi se prediktivnim metodama moglo iz velikih skupova podataka predvidjeti koliko bi energije u određenom vremenskom periodu ovakav

sustav mogao proizvesti. Provedba navedene analize potrebna je da bi postupak izrade digitalnog blizanca bio proveden u cijelosti.

Ovim radom nastoji se na konkretnom primjeru prikazati proces realizacije koncepta digitalnih blizanca, dajući uvid u metode i tehnologije koje su u tu svrhu potrebne te kako bi se digitalni blizanci mogli iskoristiti za razvoj novih, pametnih proizvoda. Ovaj rad može poslužiti kao dobra osnova za izradu kompleksnijih digitalno-fizičkih sustava u sklopu 3DExperience platforme za daljnja istraživanja na području Industrije 4.0.

## LITERATURA

- [1] i-SCOOP: Industry 4.0: the fourth industrial revolution – guide to Industrie 4.0  
<https://www.i-scoop.eu/industry-4-0/>
- [2] Stepinac, I.: Što je to zapravo Big Data i gdje se primjenjuje?  
<https://www.ictbusiness.info/poslovna-rjesenja/sto-je-to-zapravo-big-data-i-gdje-se-primjenjuje>
- [3] Laney, D.: 3D data management: Controlling data volume, velocity and variety, 2001.
- [4] Model-Based – The Digital Twin  
<https://virtualdutchman.com/2018/07/02/model-based-the-digital-twin/>
- [5] Grieves, M.: Digital Twin: Manufacturing Excellence through Virtual Factory Replication, LLC, 2014.
- [6] Parrott, A., Warshaw, L.: Industry 4.0 and the digital twin - Manufacturing meets its match, Deloitte university press
- [7] Detzner, A., Eigner, M.: A digital twin for root cause analysis and product quality monitoring, International design conference Design, 2018.
- [8] Joshi, N.: Applications of digital twin  
<https://www.allerin.com/blog/applications-of-digital-twin>
- [9] Spacey, J.: 11 Examples of a Digital Twin  
<https://simplicable.com/new/digital-twin>
- [10] Gutierrez, C., Khizhniak, A.: Digital Twins for Aerospace: Better Fleet Reliability and Performance  
<https://www.altoros.com/blog/digital-twins-for-aerospace-better-fleet-reliability-and-performance/>
- [11] NIWeek 2015, Austin, Texas  
<https://www.youtube.com/watch?v=8jwMVRgcVls>
- [12] Brockotter, R.: Key design considerations for 3D Printing  
<https://www.3dhubs.com/knowledge-base/key-design-considerations-3d-printing>
- [13] Arduino solar tracker  
<http://www.instructables.com/id/Arduino-Solar-Tracker/>
- [14] Simple dual axis solar tracker  
<http://www.instructables.com/id/Simple-Dual-Axis-Solar-Tracker/#comment-list>
- [15] How to Use an LDR Sensor With Arduino

- <https://maker.pro/arduino/tutorial/how-to-use-an-ldr-sensor-with-arduino>
- [16] Wikipedia: Solarna fotonaponska energija  
[https://hr.wikipedia.org/wiki/Solarna\\_fotonaponska\\_energija](https://hr.wikipedia.org/wiki/Solarna_fotonaponska_energija)
- [17] 3DExperience  
<https://www.3ds.com/products-services/3dexperience/>
- [18] 3DS Academy  
<https://academy.3ds.com/plexp>
- [19] Discover Catia  
<https://www.3ds.com/products-services/catia/>
- [20] DYMOLA System Engineering  
<https://www.3ds.com/products-services/catia/products/dymola/>
- [21] Yan, J., Meng, Y., Lu, L., Li, L.: Industrial Big Data in an Industry 4.0 Environment: Challenges, Schemes, and Applications for Predictive Maintenance, 2017.
- [22] Techopedia: Characterization  
<https://www.techopedia.com/definition/30312/characterization-data-analysis>
- [23] Dalbello Bašić, B., Čupić, M., Šnajder, J.: Umjetne neuronske mreže, 2008



## **PRILOZI**

- I. CD-R disc
- II. Programski kod

**PRILOG II.: Programski kod**

```

#include <SimpleSoftwareServo.h>
#include <MsTimer2.h>
#include <Servo.h> Servo library
// Permit to control the pins and servos with the console : p (pin change)
m (pinMode change) v (pin value)

int const limite = 95;
// 95 is a good limit for the program performance (the unread char will be
uploaded in the following loop)
char buffer[limite];
int nbDonnee = 0;
boolean informationClosed = true;

int const nbPin = 20;
int const nbDigitalPin = 14;
// for 14 to 19 convert to A0 to A5
int const nbAttribute = 3;
// 0 : type ? (0=digital, 1=PWM, 2=analog) / 1 : mode ? (0=Output,
1=Input, 2=Servo) / 2 : Last output value
int const typ = 0; int const mod = 1; int const val = 2;
int defPinType[nbPin] = {-1, -1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 2, 2,
2, 2, 2, 2}; // For each UNO pin : -1 = don't use this pin / 0 =
Digital pin / 1 = PWM pin / 2 = Analog pin
int pin[nbPin][nbAttribute];
// State table for each pin
SimpleSoftwareServo servo[nbPin];
// create servo object to control a servo if needed

int currentPin = 0;
int value = 0;
int lect = 0;
int comments = 3; // 0 = No Comment ; 1 = Error ; 2 = Command done ;
3 = Comments

// 180 horizontal MAX
Servo horizontal; // horizontal servo
int servoh = 90 ; // 90; // stand horizontal servo
int servohLimitHigh = 180;
int servohLimitLow = 65;
// 65 degrees MAX
Servo vertical; // vertical servo
int servov = 120; // 90; // stand vertical servo //45
int servovLimitHigh = 180; //80
int servovLimitLow = 15; //15
// LDR pin connections
// name = analogpin;
int ldrlt = 0; //LDR top left - BOTTOM LEFT <--- BDG
int ldrrt = 1; //LDR top right - BOTTOM RIGHT
int ldrlb = 2; //LDR down left - TOP LEFT
int ldrrb = 3; //ldr down right - TOP RIGHT

void setup()
{
    Serial.begin(9600);
    // servo connections

```

```

// name.attach(pin);
horizontal.attach(9);
vertical.attach(10);
horizontal.write(180); //180
vertical.write(45);
delay(3000);
//for(int i = 0; i < nbPin; i++)
for(int i = 18; i < nbPin; i++)
{
    pin[i][typ] = defPinType[i];
    pinMode(i, OUTPUT);
}

//for(int i = 0; i < nbPin; i++)
for(int i = 18; i < nbPin; i++)
{
    for(int j = 1; j < nbAttribute; j++)
    {
        pin[i][j] = 0;
    }
}

MsTimer2::set(20, refresh); // 20 ms period
MsTimer2::start();
}

void refresh()
{
    SimpleSoftwareServo::refresh();
}

void loop()
{
    if(Serial.available() > 0)
    {
        int lt = analogRead(ldr1t); // top left
        int rt = analogRead(ldr1r); // top right
        int ld = analogRead(ldr2l); // down left
        int rd = analogRead(ldr2r); // down right
        // int dtime = analogRead(4)/20; // read potentiometers
        // int tol = analogRead(5)/4;
        int dtime = 10; int tol = 50;
        int avt = (lt + rt) / 2; // average value top
        int avd = (ld + rd) / 2; // average value down
        int avl = (lt + ld) / 2; // average value left
        int avr = (rt + rd) / 2; // average value right
        int dvert = avt - avd; // check the diffirence of up and down
        int dhoriz = avl - avr; // check the diffirence og left and rigt

        if (-1*tol > dvert || dvert > tol) // check if the diffirence is in the
tolerance else change vertical angle
        {
            if (avt > avd)
            {
                servov = ++servov;
                if (servov > servovLimitHigh)
                {
                    servov = servovLimitHigh;
                }
            }
        }
    }
}

```

```
    } }
    else if (avt < avd)
    {
        servov= --servov;
        if (servov < servovLimitLow)
        {
            servov = servovLimitLow;
        } }
        vertical.write(servov);
    }
    if (-1*tol > dhoriz || dhoriz > tol) // check if the diffirence is in
the tolerance else change horizontal angle
    {
        if (avl > avr)
        {
            servoh = --servoh;
            if (servoh < servohLimitLow)
            {
                servoh = servohLimitLow;
            } }
            else if (avl < avr)
            {
                servoh = ++servoh;
                if (servoh > servohLimitHigh)
                {
                    servoh = servohLimitHigh;
                } }
            else if (avl = avr)
            {
                // nothing
            }
            horizontal.write(servoh);
        }
        delay(dtime);
    }

{
    lecture(buffer, limite, &nbDonnee);
}

if(completeInformation(buffer))
{
    if(buffer[0] == 'p')
    {
        // modify the pin selected
        eraseFirstChar(buffer);
        if(isAValue(buffer))
        {
            value = atoi(buffer);
            modifyPinSelected(value);
        }
    }
    else if(buffer[0] == 'm')
    {
        // modify the pin mode
        eraseFirstChar(buffer);
        if(isAValue(buffer))
        {
            value = atoi(buffer);
            modifyPinMode(currentPin, value);
        }
    }
}
```

```

    }
    else if(buffer[0] == 'v')
    {
        // modify the pin value
        eraseFirstChar(buffer);
        if(isAValue(buffer))
        {
            value = atoi(buffer);
            modifyPinValue(currentPin, value);
        }
    }
    else if(buffer[0] == ' ')
    {
        // pass to the next line in the Serial monitor
        eraseFirstChar(buffer);
        Serial.println("");
    }
    else if(nbDonnee > 0)
    {
        // clean the buffer first value if not a convention
        eraseFirstChar(buffer);
    }

    informationClosed = false;
}
else
{
    if(!informationClosed)
    {
        Serial.print("c");
        informationClosed = true;
    }
}
}

void modifyPinSelected(int value)
{
    if(value >= 0 && value < nbPin)
    {
        currentPin = value;
        Serial.print("p"); Serial.print(value);
    }
}

void modifyPinMode(int currentPin, int value)
{
    if(pin[currentPin][typ] == 0 || pin[currentPin][typ] == 1 ||
pin[currentPin][typ] == 2)
    {
        if(value == 0)
        {
            // if changes to OUTPUT
            if(servo[currentPin].attached() == 1)
            {
                servo[currentPin].detach();
            }
            pinMode(currentPin, OUTPUT);
            pin[currentPin][mod] = value; // here value == 0
            Serial.print("m"); Serial.print(value);
            applyPinValue(currentPin, pin[currentPin][val]); // Apply the last
value send
        }
    }
}

```

```

else if(value == 1)
{
    // if changes to INPUT
    if(servo[currentPin].attached() == 1)
    {
        servo[currentPin].detach();
    }
    pinMode(currentPin, INPUT);
    pin[currentPin][mod] = value; // here value ==
1
    Serial.print("m"); Serial.print(value);
    Serial.print("v"); Serial.print(readPinValue(currentPin)); // Read
the pin value
}
else if(value == 2)
{
    // if changes to SERVO
    Serial.print("m"); Serial.print(value);
    if(servo[currentPin].attached() != 1)
    {
        servo[currentPin].attach(currentPin); // if
needed the min and max pulse width can be change from 544 and 2400, call :
"servo.attach(pin, min, max)"
    }
    pin[currentPin][mod] = value; // here value == 2
    applyPinValue(currentPin, servo[currentPin].read()); //
Apply the last value send
}
}
}

void modifyPinValue(int currentPin, int value)
{
    if(pin[currentPin][typ] == 0 || pin[currentPin][typ] == 1 ||
pin[currentPin][typ] == 2)
    {
        if(pin[currentPin][mod] == 1)
        {
            Serial.print("m0");
            pin[currentPin][mod] = 0;
            pinMode(currentPin, OUTPUT);
        }
        applyPinValue(currentPin, value);
    }
}

void applyPinValue(int currentPin, int value)
{
    if(pin[currentPin][mod] == 0)
    {
        // if it's an OUTPUT
        if(value == 0)
        {
            digitalWrite(currentPin, LOW);
            pin[currentPin][val] = value;
            Serial.print("v"); Serial.print(value);
        }
        else if(value == 255)
        {
            digitalWrite(currentPin, HIGH);

```

```

        pin[currentPin][val] = value;
        Serial.print("v"); Serial.print(value);
    }
    else if(pin[currentPin][typ] == 1 && value > 0 && value < 255)
    {
        analogWrite(currentPin, value);
        pin[currentPin][val] = value;
        Serial.print("v"); Serial.print(value);
    }
}
else if(pin[currentPin][mod] == 2)
{}
}
// {           // if it's a SERVO
//     if(value >= 0 && value <= 180)
//     {
//         Serial.print("v"); Serial.print(value);
//         if(servo[currentPin].attached() != 1)
//         {           // if detached
//             servo[currentPin].attach(currentPin);           // if
// needed the min and max pulse width can be change from 544 and 2400, call :
// servo.attach(pin, min, max)"
//         }
//         servo[currentPin].write(value);
//     }
// }
int readPinValue(int currentPin)
{
    if(pin[currentPin][mod] == 1)
    {
        // if it's an INPUT
        if(pin[currentPin][typ] == 0 || pin[currentPin][typ] == 1)
        { // if it's a digital or PWM
            return digitalRead(currentPin)*1023;
        }
        else if(pin[currentPin][typ] == 2)
        {           // if it's an analog
            return analogRead(currentPin);
        }
    }
    else if(pin[currentPin][mod] == 2)
    { // if it's a SERVO
        if(servo[currentPin].attached() == 1)
        {
            return servo[currentPin].read();           //
// return the last value sent in .write()
        }
    }
    return -1;
}

void eraseFirstChar(char * buffer)
{
    for(int i = 0; i < nbDonnee-1; i++)
    {
        buffer[i] = buffer[i+1];
    }
    buffer[nbDonnee-1] = '\0';
    nbDonnee = nbDonnee-1;
}

```

```
}

boolean isAValue(char * buffer)
{
    boolean isValue = false;
    if(atoi(buffer) != 0 || buffer[0] == '0')
    {
        isValue = true;
    }
    return isValue;
}

void lecture(char * buffer, int maxSize, int *lenght)
{
    /*lenght = 0;      // Clear all the buffer with the new message
    while(Serial.available() > 0 && *lenght < maxSize-1)
    {
        buffer[*lenght] = Serial.read();
        (*lenght)++;
        //delay(3);    // Needed with the serial monitor if there is no
complete information check
    }
    buffer[*lenght] = '\0';
}

boolean completeInformation(char * buffer)
{
    boolean infoComplete = false;
    int nbInfo = 0;
    for(int i = 0; i < nbDonnee && !infoComplete; i++)
    {
        if(buffer[i] == 'p' || buffer[i] == 'm' || buffer[i] == 'v' ||
buffer[i] == 'c')
        {
            nbInfo++;
            if(nbInfo > 1)
            {
                infoComplete = true;
            }
        }
    }
    return infoComplete;
}
```