



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У
НОВОМ САДУ



Елена Стојковски

Развој физичке архитектуре и програмске подршке за *Click* плочице

ДИПЛОМСКИ РАД
- Основне академске студије -

Нови Сад, 2022.



КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, РБР:		
Идентификациони број, ИБР:		
Тип документације, ТД:	Монографска документација	
Тип записа, ТЗ:	Текстуални штампани материјал	
Врста рада, ВР:	Завршни (Bachelor) рад	
Аутор, АУ:	Елена Стојковски	
Ментор, МН:	др Владимир Рајс	
Наслов рада, НР:	Развој физичке архитектуре и програмске подршке за Click pločice	
Језик публикације, ЈП:	Српски	
Језик извода, ЈИ:	Српски	
Земља публиковања, ЗП:	Република Србија	
Уже географско подручје, УГП:	АП Војводина	
Година, ГО:	2022.	
Издавач, ИЗ:	Ауторски репринт	
Место и адреса, МА:	Факултет техничких наука, 21000 Нови Сад, Трг Доситеја Обрадовића 6	
Физички опис рада, ФО: (поглавља/страна/ цитата/табела/слика/графика/прилога)	7 / 53 / 0 / 3 / 41 / 0 / 0	
Научна област, НО:	Електротехника и рачунарство	
Научна дисциплина, НД:	Примењена електроника	
Предметна одредница/Клучне речи, ПО:	Физичка архитектура, софтверска подршка, mikroBUS стандард, mikroSDK, Click плочице	
УДК		
Чува се, ЧУ:	Библиотека ФТН, Трг Доситеја Обрадовића 6, Нови Сад	
Важна напомена, ВН:		
Извод, ИЗ:	У завршном раду описаны су mikroBUS стандард, Click плочице и mikroSDK библиотеке. Објашњен је детаљно начин креирања PCB и шематик дизајна плочице на два различита примера и описане су могуће примене плочице. Такође је и детаљно описан програм креiran за једну Click плочицу, чиме је комплетирана израда и хардвера и софтвера.	
Датум прихватања теме, ДП:		
Датум одбране, ДО:		
Чланови комисије, КО:	Председник:	др Мирјана Дамњановић, проф, ФТН Нови Сад
	Члан:	др Јован Бајић, проф, ФТН Нови Сад
	Члан, ментор:	др Владимир Рајс, проф, ФТН Нови Сад
		Потпис ментора



KEY WORDS DOCUMENTATION

Accession number, ANO:		
Identification number, INO:		
Document type, DT:	Monographic publication	
Type of record, TR:	Textual printed material	
Contents code, CC:	Bachelor Thesis	
Author, AU:	Elena Stojkovski	
Mentor, MN:	PhD Vladimir Rajs	
Title, TI:	Hardware and Software development for <i>Click</i> boards	
Language of text, LT:	Serbian	
Language of abstract, LA:	Serbian	
Country of publication, CP:	Republic of Serbia	
Locality of publication, LP:	AP of Vojvodina	
Publication year, PY:	2022.	
Publisher, PB:	Author's reprint	
Publication place, PP:	Faculty of technical sciences, 21000 Novi Sad, squ.Dositeja Obradovica 6	
Physical description, PD: (chapters/pages/ref./tables/pictures/graphs/appendices)	7 / 53 / 0 / 3 / 41 / 0 / 0	
Scientific field, SF:	Electrical Engineering	
Scientific discipline, SD:	Applied Electronics	
Subject/Key words, S/KW:	Hardware design, Software, mikroBUS standard, mikroSDK, Click boards	
UC		
Holding data, HD:	The Library of Faculty of Technical Sciences, square Dositeja Obradovica 6, Novi Sad	
Note, N:		
Abstract, AB:	This bachelor thesis describes mikroBUS standard, Click boards and mikroSDK libraries. Creating PCB and schematic designs was explained on two different examples, and possible applications of boards is described. Also, the program developed for one of the Click boards was explained, with which the creation of hardware and software is complete.	
Accepted by the Scientific Board on, ASB:		
Defended on, DE:		
Defended Board, DB:	President:	dr Mirjana Damnjanović, prof., FTN Novi Sad
	Member:	dr Jovan Bajić, prof., FTN Novi Sad
	Member, Mentor:	dr Vladimir Rajs, prof., FTN Novi Sad
		Menthor's sign



УНИВЕРЗИТЕТ У НОВОМ САДУ • ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА
21000 НОВИ САД, Трг Доситеја Обрадовића 6

Број:

ЗАДАТAK ЗА ИЗРАДУ ДИПЛОМСКОГ РАДА

Датум:

(Податке уноси предметни наставник - ментор)

Врста студија:	Основне академске студије		
Студијски програм:	Енергетика, електроника и телекомуникације		
Руководилац студијског програма:	Prof. dr Milan Sečujski		

Студент:	Елена Стојковски	Број индекса:	EE71/2018			
Област:	Примењена електроника					
Ментор:	др Владимир Рајс					
НА ОСНОВУ ПОДНЕТЕ ПРИЈАВЕ, ПРИЛОЖЕНЕ ДОКУМЕНТАЦИЈЕ И ОДРЕДБИ СТАТУТА ФАКУЛТЕТА ИЗДАЈЕ СЕ ЗАДАТАК ЗА ДИПЛОМСКИ РАД, СА СЛЕДЕЋИМ ЕЛЕМЕНТИМА:						
<ul style="list-style-type: none">- проблем – тема рада;- начин решавања проблема и начин практичне провере резултата рада, ако је таква провера неопходна;						

НАСЛОВ ДИПЛОМСКОГ РАДА:

Развој физичке архитектуре и програмске подршке за Click плочице

ТЕКСТ ЗАДАТКА:

Описати шта су *mikroBUS* стандард и *Click* плочице. Након тога описати SPI и I²C комуникације које се користе у стандарду. Потом објаснити принцип креирања плочица до најситнијих детаља на два различита примера. Навести адекватне примене креираних плочица. Детаљно описати начин писања софтвера за плочице уз помоћ *mikroSDK* библиотека.

Руководилац студијског програма:	Ментор рада:
Prof. dr Milan Sečujski	др Владимир Рајс

Примерак за: - Студента; - Ментора

SADRŽAJ

1.	Uvod.....	1
2.	<i>MikroBUS</i> standard.....	3
2.1	Informacije o mikroBUS standardu	3
2.2	Opis nožica na konektorima mikroBUS standarda	4
2.3	Click pločice	5
3.	Mašinski postupak proizvodnje pločica.....	8
3.1	Postupak štampanja lemne paste na pločice	9
3.2	Postupak postavljanja komponenti na pločice	10
3.3	Postupak zagrevanja pločica	10
3.4	Druge korisne mašine u procesu proizvodnje pločica.....	12
4.	Komunikacije	13
4.1	SPI	13
4.2	I ² C.....	16
5.	Kreiranje Click pločica.....	19
5.1	CapTouch 6 Click.....	19
5.1.1	Opis senzora AS8579	20
5.1.2	Dizajn šematska pločice <i>CapTouch 6 Click</i>	22
5.1.3	PCB dizajn pločice <i>CapTouch 6 Click</i>	24
5.1.4	Primena <i>CapTouch 6 Click</i> pločice.....	28
5.2	LightRanger 9 Click	32
5.2.1	Opis senzora TMF8828	32
5.2.2	Dizajn šematska pločice <i>LightRanger 9 Click</i>	34
5.2.3	PCB dizajn pločice <i>LightRanger 9 Click</i>	35

5.2.4	Primena LightRanger 9 Click pločice	37
6.	Razvoj softvera za <i>Click</i> pločice	39
6.1	mikroSDK.....	39
6.2	Program za Temp&Hum 21 Click pločicu	41
6.2.1	Kreiranje <i>header</i> fajla.....	42
6.2.2	Kreiranje <i>source</i> fajla	45
6.2.3	Glavna funkcija.....	47
7.	Zaključak.....	49
8.	Literatura.....	52

SKRAĆENICE

AFE	- <i>Analog Front-End</i> – arhitektura koju čini skup kola za kondicioniranje analognog signala
API	- <i>Application Programming Interface</i>
FM	- <i>Frequency Modulation</i> – Frekventna modulacija
GND	- <i>Ground</i> – masa, uzemljenje
I²C	- <i>Inter-Integrated Circuit</i>
LED	- <i>Light Emitting Diode</i> – dioda koja emituje svetlost
MOSFET	- <i>Metal – Oxide – Semiconductor Field-Effect Transistor</i>
PCB	- <i>Printed Circuit Board</i> – dizajn hardvera koji se kreira za štampanje pločice
RAM	- <i>Random Access Memory</i>
SDK	- <i>Software Development Kit</i> – komplet za razvijanje softvera
SMD	- <i>Surface Mount Device</i> – elektronska komponenta koja se montira površinski
SMT	- <i>Surface Mount Technology</i> – tehnologija izrade sa površinskom montažom
SPAD	- <i>Single-Photon Avalanche Diode</i> – jednofotonska lavinska dioda
SPI	- <i>Serial Peripheral Interface</i>
UART	- <i>Universal asynchronous receiver-transmitter</i>
VCESL	- <i>Vertical-cavity surface-emitting laser</i> – vrsta lasera

1. Uvod

Izbor teme za izradu ovog diplomskog rada je inspirisan upravo razvojem elektronike. Razvoj kao takav je neophodni element funkcionisanja ljudskog društva. Svako od nas se razvija, raste, stiče nova znanja, unapređuje sebe kao osobu. Isto tako, potrebno je razvijati nove tehnologije, koje će pratiti naše potrebe. Da je u prošlom veku došlo do stagnacije u razvoju tehnologija, danas ne bismo imali pametne telefone, satove, kućne uređaje, razvijene medicinske aparate, i slično, što bi ljudski život učinilo mnogo težim.

Elektronika, kao i primena elektronskih komponenti danas ima sve veći značaj. Nalazi se svuda oko nas i znatno olakšava život ljudi. Upravo zbog toga, svakodnevno rastu ljudske potrebe za novim proizvodima, pa je samim time traženiji razvoj novih fizičkih arhitektura (hardvera), komponenti, kao i programske podrške (softvera). Sve je veća težnja da se nadogradi stari, ili napravi novi, inovativni elektronski proizvod koji će doprineti ljudskim potrebama. Ljudska težnja za razvojem elektronike se najbolje može dočarati kroz kratak istorijat iste.

Iako je prvi elektronski uređaj – relej napravljen 1835. godine od strane naučnika Džozefa Henrija (*Joseph Henry*), smatra se da je sam početak ere elektronike bio pre samo nešto više od 100 godina. Po nekim izvorima, bilo je to 1904. godine, otkrićem vakuumskе cevi (diode) od strane naučnika Džona Ambroza Fleminga (*John Ambrose Fleming*). U drugim izvorima se navodi da se početak elektronike desio ipak 1906. godine, kada je naučnik Li de Forest (*Lee de Forest*) izumeo triodu. Nakon ovoga su usledili novi uređaji i komponente, poput elektročnog oscilatora (1902, *Edwin Howard Armstrong*), televizora (1927, *Philo Farnsworth*), FM radia (1933, *Edwin Armstrong*), prvog programabilnog računara (završen 1941, *Konrad Zuse*), bipolarnog tranzistora (1947, *Bell Telephone Laboratories*)...

S mogućnošću primene novih komponenti i tehnologija, počela su usavršavanja već postojećih uređaja, tako je, na primer, 1950. godine prvi put predstavljen televizor u boji. Težnja da se naprave komponente koje su manje, jeftinije i ekonomičnije je dovela do nastanka prvih

MOSFET-a, 1962. godine. Par godina kasnije nastale su prve poluprovodničke *RAM* memorije, od samo 64 bita, kao i prvi mikroprocesori koji su bili četvorobitni.

Danas, elektronika čini sastavni deo svakodnevnog života. Prisutna je u pomagalima za zdravstvene svrhe, poput slušnih aparata, pokretnih kolica, raznih uređaja u bolničkim laboratorijama. Daje nam osećaj lične sigurnosti, kao i sigurnosti materijalne svojine, primenom raznih senzora i aktuatora u kombinacijama koje čine alarmne sisteme. Omogućava edukaciju, zabavu, ili jednostavno „ubijanje dosade” pomoću televizora, kompjutera i interneta, kao i kontakt sa svetom pomoću mobilnih uređaja, nezavisno od vremenskih okvira i geografske pozicije osobe. Doprinosi uštedi prirodnih dobara i očuvanju sredine primenom električnih automobila. Omogućava nam da se zabavimo u virtuelnoj realnosti pomoću VR „slušalica”, koje se koriste za sasvim drugačije doživljaje u igranju igrica, opuštanje, razgledanje... Ekrani su mnogo kvalitetniji, te se smanjuje rizik od oštećenja vida primenom OLED (*Organic Light - Emitting Diode*) ekrana, čija je slika mnogo prijatnija, prirodnija i bezbednija za vid.

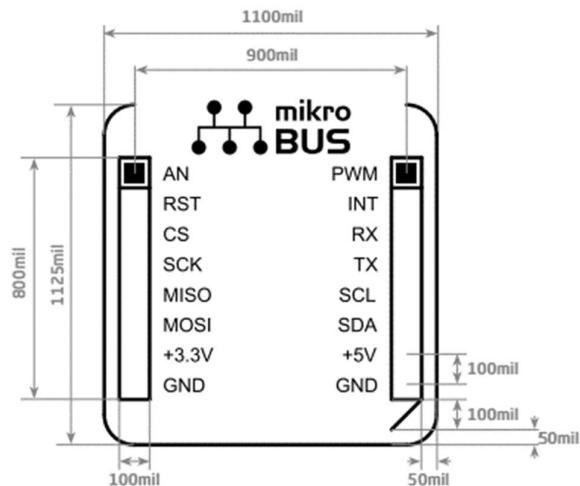
Cilj elektronike nije samo osmišljavanje novih uređaja, nego i unapređivanje starih. Zbog potreba industrije, a naročito vojne, dolazilo je do porasta potražnje za boljim elementima i naučnicima koji bi doprineli istraživanjima. Komponente koje su nekada bile ogromnih dimenzija, danas mogu da se naprave u nanotehnologiji, čime se povećava gustina pakovanja i smanjuju dimenzije krajnjeg uređaja. Usavršavanje različitih tehnologija dovelo je do toga da su današnji uređaji, po kvalitetu, drastično ispred uređaja iz perioda njihovog nastanka, pri čemu im je cena mnogo manja. Primer ovoga može biti upravo programibilni računar. Prvi računari su nekada bili veličine ormara, sa jako malo funkcija. Danas imamo računare veličine sveske, sa mnogo većom brzinom rada, više funkcija, memorije, i po veoma pristupačnim cenama.

Elektronika unosi još mnoštvo dobrih stvari u naše živote, sa ciljem da ih olakša ili učini zabavnijim. Iz tog razloga je izuzetno bitno razvijati svakim danom nove uređaje i komponente koje će imati svoju ulogu u ljudskim životima i svoje mesto primene. Tema ovog rada je upravo kreiranje novih uređaja po *mikroBUSTM* standardu, to jest, *Click* pločica, o čemu će biti reč u narednim poglavljima. Osim samog kreiranja uređaja, biće reči i o razvoju programske podrške za iste, koja je od jednakе važnosti. Programska podrška je rađena pomoću *mikroSDK* biblioteka.

2. MikroBUS standard

2.1 Informacije o mikroBUS standardu

Kao što je već pomenuto, za izradu uređaja (hardvera) korišćen je *mikroBUSTM* standard. Ovaj standard je razvijen od strane kompanije *Mikroelektronika* iz Beograda. On definiše *socket* (priključak) na razvojnim sistemima, i predstavlja *add-on board* (dodatak za ploču ili razvojni sistem) standard sa najbržim razvojem na svetu. Svrha *mikroBUS*-a je da se napravi standardizovan priključak za različite vrste *add-on* hardvera (dalje u tekstu – *Click* pločica), i da omogući rad sa velikim opsegom različitih uređaja pomoću samo jedne vrste priključka. Rad sa ovakvim pločicama je nezavisan od toga kakvu svrhu ima uređaj na pločici, i kojeg je od tipa (senzor, aktuator).



Slika 2.1. Raspored i udaljenost nožica na *mikroBUSTM* priključcima

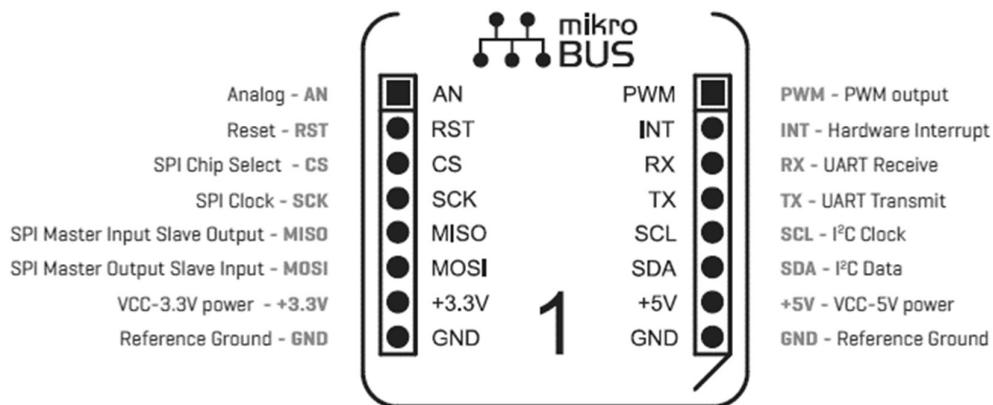
Izvor: Zvaničan sajt firme *Mikroelektronika*, www.mikroe.com

Ovaj standard se koristi za povezivanje različitih pločica sa mikrokontrolerima i mikroprocesorima. Veliku primenu je našao u razvojnim sistemima koji sadrže *mikroBUS* konektor, kontroler (ili konektor za kontroler), programator, i druge pomoćne elemente. Na korisniku je samo da izabere željene elemente (*Click* pločicu, kontroler, dodatne uređaje), nakon čega se može posvetiti razvijanju softvera.

Priklučak se sastoji od 16 nožica, organizovanih u paru. One su uvek poređane istim redosledom, na istim udaljenostima. Raspored nožica, kao i udaljenosti su prikazani na slici 2.1. Ukoliko neko ne poseduje razvojni sistem ili elektronsku tablu sa ovim priključkom, postoji i opcija povezivanja pločica dizajniranih za *mikroBUS* pomoću protoborda. Sa slike 2.1. se može videti da je između dve susedne nožice razmak 100 mila¹, dok je između dva reda nožica razmak 900 mila. Ovaj razmak odgovara razmacima između šupljina na protobordu, koji iznosi 100 mila.

Druga opcija za povezivanje *mikroBUS* uređaja je implementacija odgovarajućeg konektora pri kreiranju PCB dizajna. Moguće je napraviti sopstveni uređaj, ili čak razvojni sistem, koji sadrži konektor u koji se može dodati *Click* pločica.

2.2 *Opis nožica na konektorima mikroBUS standarda*



Slika 2.2. Nožice na priključku sa kratkim opisom

Izvor: Zvaničan sajt firme *Mikroelektronika*, www.mikroe.com

Kratak opis svih nožica se može videti na slici 2.2. Nožice (pinovi) se mogu podeliti u tri grupe:

1. Pinovi za komunikaciju (SPI, UART, I²C)
2. Pinovi za napajanje (+3.3V, +5V, GND)
3. Dodatni pinovi (Analog Input, Reset, Chip Select, PWM, Interrupt).

¹ Merna jedinica, 100 mil = 2,54 mm

Omogućene su tri različite vrste komunikacija između pločice i kontrolera. Za korišćenje SPI komunikacije, potrebne su četiri fizičke linije, a time i četiri pina: CS (*Chip Select*), SCK (*SPI clock*), MISO (*Master Input Slave Output*) i MOSI (*Master Output Slave Input*). UART komunikacija koristi pin TX (*Transmit*) za slanje i pin RX (*Receive*) za primanje podataka. Treća komunikacija je I²C koja koristi SCL (*Clock line*) i SDA (*Data line*) pinove za davanje signala takta i prenos podataka.

Za napajanje su ponuđene dve različite vrednosti napona. Priključena pločica može koristiti ili napajanje od +3.3V, ili od +5V. Pojedine pločice imaju implementirana oba napajanja, gde se pomoću kratkospojnika vrši izbor koje će se koristiti. Opcija sa kratkospojnikom je efikasna ukoliko se na pločici nalazi komponenta kojoj odgovaraju obe vrste napajanja, ili za koju je moguće otpornicima regulisati napon od izvora napajanja do komponente tako da se ne prekoračuje ni jedno ograničenje koje ona poseduje.

Od dodatnih pinova imamo analogni pin (AN), pin za ponovno pokretanje ili restart (RST) pin za impulsno – širinsku modulaciju (PWM), pin za hardverski prekid (INT), i pin za izbor čipa (CS). Ukoliko je potrebno, dodatne pinove je moguće koristiti i u druge svrhe. Na primer, pin CS se može koristiti kao pin za dozvolu (*enable*, EN).

Ovakav skup nožica omogućava maksimalni opseg uređaja za povezivanje sa minimalnom količinom nožica. Većina komponenti koje se pojavljuju danas na tržištu radi na +3.3V ili na +5V, koristi neku od tri ponuđene komunikacije i ne zahteva više dodatnih pinova nego što imamo na raspolaganju.

2.3 Click pločice

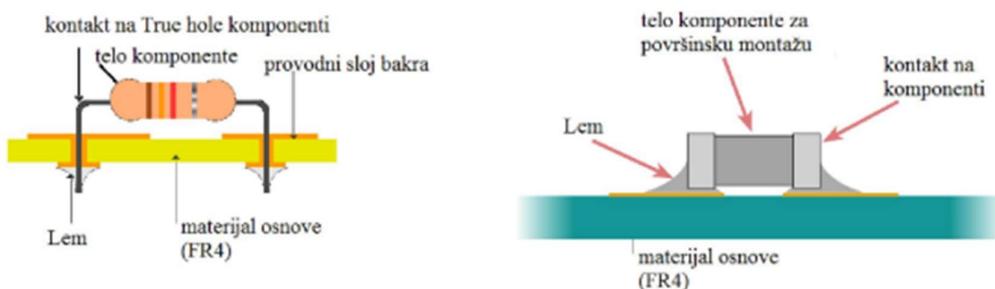
U odeljku 2.2. se pominje pojam „Click pločica”. Upravo za svrhe *mikroBUS* standarda, kreirane su Click pločice od strane kompanije *Mikroelektronika*. One predstavljaju hardverske uređaje kreirane tako da sadrže nožice po *mikroBUS* standardu, te se mogu sa lakoćom ubaciti u postojeće konektore. Do sada je kreirano više od 1.300 različitih pločica. Svaka pločica sadrži jednu ili više glavnih komponenti, od kojih zavisi uloga i primena pločice. Primenjene komponente mogu biti senzori za topotu, vlažnost vazduha, osvetljenost, gasove, merenje udaljenosti, pritisak, mikrofon... Osim senzora, pločice mogu sadržati i aktuator, poput ekrana, različitih matrica sa svetlećim diodama, zvučnika... Pored ovih, na prvi pogled, jednostavnijih pločica, postoje i pločice koje služe za upravljanje uređajima (drajver za motor), pločice sa *GPS* ili *GSM* modulima, pojačavači snage ili signala, ili bilo kakav uređaj koji može da nađe primenu.

Click pločice olakšavaju rad korisnicima, jer predstavljaju gotove uređaje koji su spremni za razvoj softvera na njima. Ovo je najlakše objasniti na primeru. Pretpostavimo da neka osoba želi da napravi barometar. Izborom glavne komponente za merenje vazdušnog pritiska, i povezivanjem iste na kontroler kolo nije završeno. Postoji niz drugih komponenti koje je potrebno međusobno povezati. Prevashodno treba detaljno izučiti dokumentaciju glavne komponente radi provere različitih ograničenja (napona, struje, snage). Uz to je neophodno poznavanje, kako analognih, tako i digitalnih elektronskih kola, zbog postojanja osnovnih tehničkih stvari koje se ne spominju u dokumentacijama (poput dodavanja *decoupling* kondenzatora između mase i napajanja, *pull-up* otpornika kod vodova komunikacije za I²C...). Osim kreiranja šematika, potrebno je što bolje povezati (izrutirati) komponente na *PCB* dizajnu, ili ih ispravno spojiti žicama na protobordu. Do problema dolazi ukoliko osoba ne ume da se snađe oko kreiranja hardvera, ali se razume u softver

i želi da isprogramira uređaj. Upravo zato su ove pločice izuzetno korisne i smanjuju posao korisnicima koji bi hteli da programiraju određene čipove.

Dimenzije predstavljaju još jednu prednost *Click* pločica. Sve komponente na pločici je moguće gusto upakovati jer su jako malih dimenzija. Click pločice dolaze u tri različite veličine. Širina pločice je uvek ista, i ona iznosi 25,4mm. Najmanja veličina je S (*small*, mala), i njena visina iznosi 28,6mm. Pločica veličine M (*medium*, srednja) je visine 42,9mm. Najveća pločica je veličine L (*large*, velika), visine 57,15mm. Pločice su veoma kompaktne, i maksimalno je iskorišćen prostor na njima, te dobijamo uređaje minimalnih dimenzija.

Pri izradi pločica se koristi tehnologija pod nazivom *SMT*, koja koristi *SMD* komponente čije dimenzijs mogu biti i deset puta manje od komponenti korišćenih u *Through Hole* tehnologiji koje se uobičajeno koriste pri izradi pločica koja nije mašinska. Upravo zato što komponente mogu biti izuzetno malih dimenzija, nije preporučljivo koristiti ih ukoliko se proces lemljenja (ostvarivanja elektročnog kontakta između komponente i lemne površine) vrši ručno.



Slika 2.3.1.. Prikaz montiranja *Through Hole* komponenti (levo), i montiranja *SMD* komponenti na pločice (desno).

Izvor: *Praktikum za softverski paket Altium Designer 17 na predmetu Praktična elektronika*

Na slici 2.3.1. je prikazana glavna razlika u montiranju ova dva tipa komponenti. Kao što se može videti na slici 2.3.1. levo, za *Through Hole* tehniku je potrebno izbušiti rupe na pločici kroz koje će proći nožice komponente, i sa donje strane je potrebno zalemiti komponentu za pločicu. Sa druge strane, *SMD* komponente ne zahtevaju dodatan proces bušenja rupa u pločici, i njihovi kontakti se leme sa iste strane na kojoj se nalazi komponenta, vidljivo na slici 2.3.1. desno.

U konkretnom primeru montiranja *SMD* komponente na slici 2.3.1. desno, postoji mogućnost i ručnog zaledljivanja jer se kontakti nalaze na pristupačnom mestu. Tipovi *SMD* komponenti su prikazani na slici 2.3.2. Za kućišta poput *SOT* (*Small Outline Transistor*), ne bi bio veliki problem ručno povezati komponentu. *SOIC* (*Small Outline Integrated Circuit*) kućišta imaju kontakte na pristupačnom mestu, ali su oni dosta blizu jedni drugima, te je proces lemljenja za ovakva kućišta otežan jer zahteva veliku preciznost i mirnu ruku. *QFP* (*Quad Flat Package*) kućišta imaju još manje pinove i razmak između njih. Postoje i kućišta poput *BGA* (*Ball Grid Area*) i *LCCC* (*Leadless Ceramic Chip Carrier*), koje nemaju izvedene nožice, nego im se kontakti nalaze na donjoj strani. Kod ovakvih kućišta je pomoću lemilice praktično neizvodljivo povezati komponentu za pločicu, već su potrebne tehnologije sa lemnom pastom i zagrevanjem.

Komponente sa kontaktima sa donje strane se danas sve više proizvode zbog svojih malih dimenija i uštedi na prostoru pri dizajniranju pločica. Najoptimalnije rešenje je korišćenje mašinske proizvodnje za izradu pločica. Pri kreiranju *Click* pločica, kompanija *Mikroelektronika* koristi mašine za proizvodnju, što ove pločice čini veoma pouzdanim. Greške koje nastaju pri izradi pločica su minimalne i svaka pločica prolazi kroz proces testiranja nakon završene faze proizvodnje.



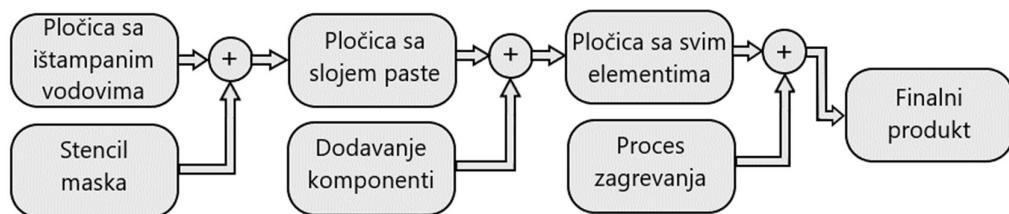
Slika 2.3.2. Tipovi kućišta za *SMT* komponente

Izvor: www.electronicsandyou.com

3. Mašinski postupak proizvodnje pločica

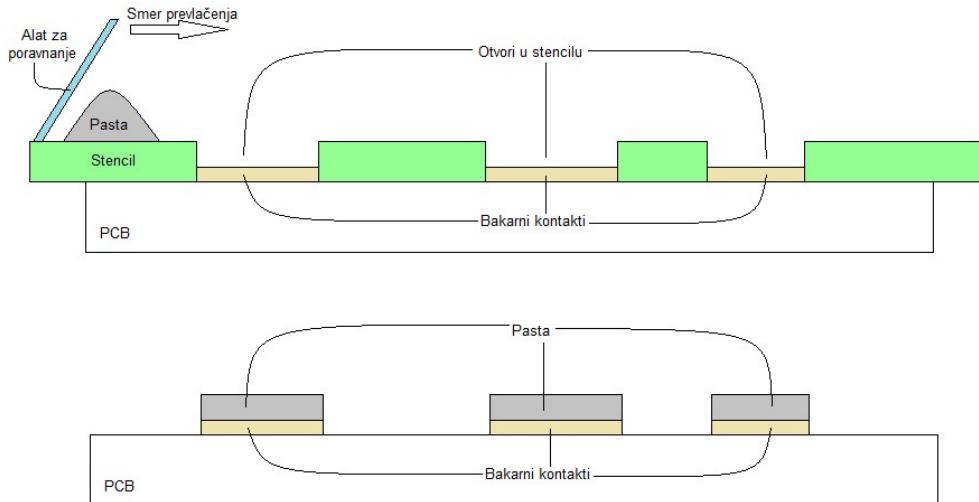
Ukoliko su *SMD* komponente izuzetno male, gusto napakovane ili sa nepristupačnim kontaktima, postoji veliki rizik od greške u lemljenju (kratak spoj, hladan lem, pregorevanje komponenti...) ako se ne koriste stručne mašine. Postoji niz mašina koje se mogu koristiti u ovom postupku, te se sam postupak izrade pločica automatizuje.

Na slici 3.1. je prikazan algoritam procesa potrebnih za proizvodnju pločice. Na početku procesa je potrebna štampana pločica, koja na sebi mora da sadrži odštampane vodove, koji su ucrtani u PCB dizajnu. Pored vodova, na pločici mogu biti odštampane i oznake komponenti, pinova, logo, ili drugi natpisi i crteži koji utiču samo na estetiku ploče. Dalje se na pločicu, pomoću maske, to jest, stencila, nanosi sloj paste. Na pločicu koja sada sadrži i pastu se dodaju komponente. Nakon dodavanja komponenti, pločicu je potrebno zagrevati tako da komponente stvore čvrstu vezu sa podlogom. Detaljniji opis celog postupka, zajedno sa načinom funkcionisanja mašina, dat je u nastavku.



Slika 3.1. Algoritam procesa proizvodnje pločica

3.1 Postupak štampanja lemne paste na pločice



Slika 3.1.1. Postupak štampanja paste na pločicu

Postupak štampanja paste na pločicu je vidljiv na slici 3.1.1. Na početku procesa mašinske proizvodnje pločica, treba kreirati stencil (tanka metalna ploča sa otvorima) za korišćeni PCB. Maska se stavља preko štampane pločice, i zajedno sa njom se ubacuje u mašinu za štampanje lemne paste. Stencil sadrži šupljine na svim kontaktnim mestima između komponenti i vodova na pločici. Ova mašina nanosi tanak sloj paste na stencil, koji ravnomerno rasporedi i izravna u svim otvorima pomoću ravног alata. Alat je potrebno postaviti pod uglom u odnosu na podlogu za najbolje rezultate raspoređivanja paste. Nakon toga se stencil uklanja i ostaje lemljena pasta samo na potrebnim kontaktnim mestima. Debljina sloja paste zavisi od debljine stencila. Ovako je moguće praviti sloj paste odgovarajuće debljine koja neće dovesti do prelivanja ukoliko ima previše paste, ili do lošeg lema ukoliko je nema dovoljno.

Za ovaj postupak mašina nije neophodna, moguće je napraviti sloj paste ručno. Ručni metod zahteva da se maska postavi preko ploče, zatim se preko maske dodaje pasta. Pomoću obične plastične kartice, ili nekog drugog ravnog predmeta, treba izravnati pastu i ravnomerno je rasporediti u svim šupljinama, u nivou visine stencila. Kada su sve šupljine popunjene, i otklonjen je višak paste sa njih, stencil se može ukloniti. Potencijalni problemi se javljaju ukoliko maska nije dovoljno pričvršćena za podlogu. Usled pomeranja maske, može doći do razmazivanja paste, i njenog nanošenja u većoj količini nego što je potrebna.

3.2 Postupak postavljanja komponenti na pločice

Pick And Place je jedna od najvažnijih mašina u procesu proizvodnje, ali i najskuplja. Kao što joj naziv kaže, to je mašina koja će da izabere komponentu i postavi je na pločicu. Komponente se dodaju tako da njihovi kontakti naležu na prethodno nanetu pastu. Obzirom da pasta ima svojstvo lepljivosti, kada se na nju postave komponente, ona sprečava pomeranja komponenti. Komponente, koje ova mašina dodaje, pakuju se u jednom redu na uskoj traci, namotanoj u rolnu. Namotaje sa komponentama postavljamo u mašinu, i softverski joj naglašavamo gde se nalazi kolut sa kojim komponentama, tako što joj nagovestimo lokaciju namotaja zajedno sa specifičnim brojem za komponente koju sadrži.

U softver maštine je potrebno ubaciti PCB dizajn po kojem se kreiraju pločice, nakon čega ona očitava podatke iz dizajna da bi znala tačne pozicije svake od komponenti koje treba postaviti, i specifičan broj za svaku vrstu komponente. Specifičan broj za komponentu zavisi od proizvođača, i pomaže mašini da odabere odakle će uzeti komponentu koju postavlja, to jest, sa kojeg koluta će uzeti komponente. Sada, kada mašina zna odakle uzima svaku od komponenti, kao i raspored komponenti na pločici, može početi sa radom. Mašina je izuzetno brza i precizna, greške koje pravi su minimalne. Brzina postavljanja komponenti može biti čak i više od 100.000 komponenti po satu, i varira zavisno od tipa maštine i njenih performansi.

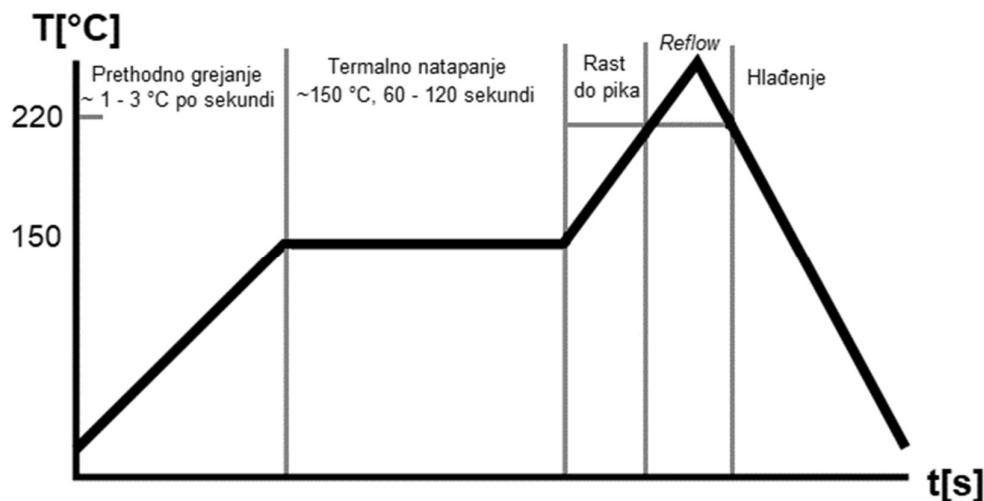
Postoje četiri različita tipa *Pick And Place* maština.

- *In-line placement* – kod ovog tipa maštine se pločica postavlja na pokretnu traku. Duž pokretne trake se nalaze robotske glave. Kada pločica na traci dođe ispred robotske glave, glava postavlja individualnu komponentu koju drži, i pločica nastavlja dalje da se kreće do sledeće glave. Kada pločica prođe celu pokretnu traku, proces je završen.
- *Sequential Placement* – komponente se postavljaju jedna po jedna na pločicu. Redosled postavljanja se može zadati programski pre početka procesa, zajedno sa izborom dela maštine koji će biti pokretan. Moguće je zadati da bude pokretka podloga po X – Y osi, ili da se kreće glava robota.
- *Sequential/simultaneous Placement* – sistem sličan prethodnom. Podloga je pokretna u dve ose. Mašina sadrži više robotskih glava za postavljanje komponenti, te ova mašina postavlja individualne komponente sukcesivo.
- *Simultaneous Placement* – najbrža mašina za postavljanje komponenti. Sve komponente se pozicioniraju i postavljaju na pločicu tokom samo jedne operacije.

3.3 Postupak zagrevanja pločica

Nakon prolaska kroz *Pick And Place* mašinu, na pločicama se nalaze sve *SMD* komponente, ispod kojih postoji sloj lemne paste. Pastu je potrebno zagrejati, tako da se kalaj koji ona sadrži dovede do tačke topljenja. Kalaj je element koji se topi na nižoj temperaturi u odnosu na druge elemente ($231,93\text{ }^{\circ}\text{C}$) i stvara čvrstu vezu nakon hlađenja. Pri grejanju pločice može doći do oštećenja nekih komponenti zbog njihove temperaturne otpornosti. Ne trpe sve komponente visoke temperature. U samom procesu grejanja temperatura treba da raste gradijalno, i ceo proces ne bi

treba da traje predugo, zbog smanjenja rizika od oštećenja usled naglih promena temperature ili usled predugoj izloženosti visokoj temperaturi. Za kontrolu grejanja pločice radi ostvarivanja električnog kontakta, nakon *Pick And Place* maštine se pločice vode u, takozvanu, rernu (*Reflow Oven*), koja je programibilna i umanjuje mogućnost nastanka greške usled prethodno spomenutih problema. Vremenski dijagram celog procesa grejanja pločice prikazan je na slici 3.3.1.



Slika 3.3.1. Vremenski dijagram procesa zagrevanja pločice pri SMT izradi

Kao što se može primetiti sa slike vremenskog dijagrama, ciklus je podeljen u nekoliko delova. U prvoj fazi se temperatura polako zagreva, u proseku 1 do 3 °C po sekundi, radi sprečavanja termalnog šoka ploče ili neke od komponenti na njoj. Komponente mogu pretrpeti velika oštećenja usled termalnog šoka, poput pucanja komponente. Ukoliko bismo naglo podigli temperaturu, postojala bi šansa da neki delovi pločice nisu uspeli da se zagreju do željene temperature zbog kratkog vremenskog intervala koji im je pružen, što je još jedan od razloga za prethodno grejanje.

Kada pločica dostigne temperaturu od približno 150 °C (zavisno od tipa paste, preporučeni opseg je od 150 °C do 175 °C) sledi faza termalnog natapanja, koja traje između 60 i 120 sekundi. Tokom ove faze temperatura slabo raste. Dolazi do isparavanja rastvarača u pasti i aktiviranja fluksa. Svrha ove faze je i da osigura da svi delovi pločice dostignu ujednačenu temperaturu. Ukoliko postoji neravnomernost temperature u pojedinim delovima pločice može doći do problema u *Reflow* fazi, poput hladnog lema (prekida u povezivanju). Do hladnog lema dolazi ukoliko se lem ne zagreje na optimalnu temperaturu i ukoliko se ne omogući dovoljni vremenski interval potreban da lem stvori kontakt. Kalaj koji nije kvalitetan i sadrži manje količine fluksa češće rezultuje pojavom hladnog lema. Natapanje toplotom ne sme trajati predugo, jer postoji rizik od preterane oksidacije i isparavanja fluksa, koje opet dovodi do stvaranja loših kontakta.

Nakon ove faze temperatura raste ponovo brzinom 1 do 3 °C po sekundi do temperature topljenja kalaja (horizontalna linija - oko 220°C na slici 3.3.1.), nakon čega se ulazi u *Reflow* zonu. U *Reflow* zoni se temperatura dodatno podiže do temperature pika. Temperatura pika se održava kratko vreme, nakon čega počinje da opada. Ova faza traje između 45 i 90 sekundi. Istopljeni kalaj

stvara čvrste veze komponenti i leme površine. Pomoću uticaja fluksa istopljeni kalaj stvara čvrstu vezu između komponenti i leme podloge.

Krajnja faza je hlađenje. Hlađenje se, isto kao i zagrevanje, mora odvijati postepeno radi izbegavanja termalnog šoka. Temperatura se spušta za 2 do 4°C po sekundi.

U zavisnosti od korišćene leme paste temperature zagrevanja mogu varirati. Dodavanjem primesa kalaju spušta se temperatura topljenja. Samim tim potrebne su niže temperature pri procesu grejanja pločice. Na parametre u procesu utiče i čistoća fluksa. Paste sa nižom temperaturom topljenja svoju primenu nalaze pri kreiranju pločica koje sadrže komponente sa slabijom tolerancijom na toplotu. Ukoliko je otpornost komponente na toplotu niža od temperature pika za određenu pastu, neophodno je koristiti drugu radi izbegavanja potencijalnih kvarova. Preporučljivo je, pre upotrebe paste, proučiti njen *datasheet* (fajl sa svim potrebnim podacima), ukoliko postoji. *Datasheet* za leme paste obično sadrži sve potrebne podatke (vremena izvršavanja faza, brzine porasta temperature, temperature natapanja i pika) za ovaj proces.

U tabeli 3.3.1, su prikazane razlike u parametrima tokom procesa za dve vrste legura. Parametri su upoređeni za Sn63 Pb37 (kalaj sa primesom olova), i Sn96.5 Ag3 Cu0.5 (kalaj sa primesama srebra i bakra), koji se drugačije označava kao SAC305. Temperatura topljenja za Sn63 Pb37 iznosi 183 °C, a za SAC305 iznosi 217 - 219 °C. Iako je za leguru kalaja i olova niža temperatura topljenja, a samim tim i temperature u procesu, olovo ispušta štetne gasove pri isparavanju i nije preporučljiva za korišćenje, pogotovo ukoliko postoji mogućnost udisanja gasova.

Zona \ Legura	Sn63 Pb37	SAC305
Početno grejanje	Do 150 °C za 60 s	Do 150 °C za 60 s
Natapanje	Od 150 °C do 165 °C za 120 s	Od 150 °C do 180 °C za 120 s
<i>Reflow</i>	Pik od 225 °C do 235 °C, održava se 20 s	Pik od 245 °C do 255 °C, održava se 15 s
Hlađenje	-4 °C/s	-4 °C/s

Tabela 3.3.1. Razlika parametara procesa grejanja za dve vrste legura

3.4 Druge korisne mašine u procesu proizvodnje pločica

Osim ove tri mašine, u proizvodnji se koriste i druge mašine. Štampa vodova na pločici se može uraditi samostalno (metoda sa peglom, metoda hladnog transfera pomoću acetona, foto-postupak...), ili pomoću neke od mašina za štampu. Ukoliko pločica sadrži i *Through Hole* komponente, one se mogu dodavati samostalno ili uz pomoć mašina, pri čemu je potrebno dodatno izbušiti rupe u pločici za montažu *Through Hole* komponenti. Mašine za ispitivanja kvaliteta pločica znatno olakšavaju posao radnicima koji bi morali da proveravaju svaku pločicu ponaosob, što je izuzetno obiman posao ukoliko se pločice štampaju u velikim serijama. Obzirom da maska za stencil može sadržati veliki broj sitnih i gusto napakovanih (bliskih) otvora, poželjno je da se i postupak izrade maske uradi mašinski.

4. Komunikacije

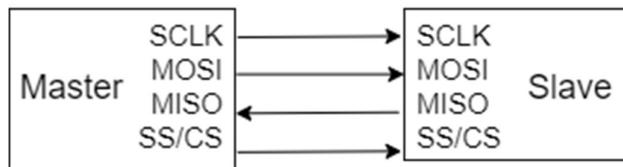
Kao što je već naglašeno u drugom poglavlju, *mikroBUS* standard od komunikacija koristi *SPI*, *I²C*, ili *UART*. U narednim poglavljima će biti objašnjen način izrade dve pločice od kojih prva pločica koristi *SPI* a druga *I²C* komunikaciju. Pored toga, biće obrađen i način pisanja programa za pločicu. Radi boljeg razumevanja kako dizajna hardvera tako i razvoja softvera, ove dve vrste komunikacije će biti detaljnije objašnjene.

4.1 SPI

SPI (*Serial Peripheral Interface*) je sinhrona serijska komunikacija. Razvijena je od strane kompanije *Motorola* sredinom 1980-ih godina. *SPI* je jedan od najrasprostranjenijih interfejsa između mikrokontrolera i periferijskih uređaja. Komunikacija je *full-duplex* tipa, što znači da se podaci mogu razmenjivati u oba smera istovremeno, poput razgovora telefonom gde dva sagovornika u isto vreme mogu i pričati i slušati drugug. Komunikacija se vrši između *master* i *slave* uređaja. Podaci se sinhronizuju na rastuću ili opadajuću ivicu taktnog signala.

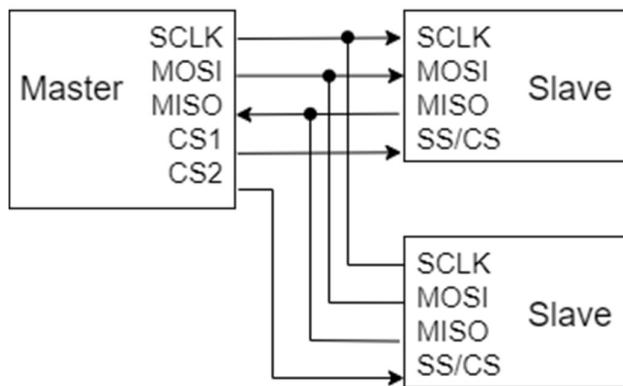
Za ostvarivanje komunikacije kod *SPI* protokola potrebne su četiri fizičke linije (slika 4.1.1):

1. **SCLK** – *Serial Clock*, sinhronizuje izlazne podatke (bitove) sa mastera sa uzorkovanjem bitova na *slave*-u. U svakom ciklusu signala takta se prenosi jedan bit podatka, te sama frekvencija taktnog signala diktira brzinu prenosa podataka.
2. **MOSI** – *Master Out Slave In*, pomoću ove linije se serijski šalju podaci bit po bit od *master* ka *slave* uređaju.
3. **MISO** – *Master In Slave Out*, linija za slanje podataka serijski, bit po bit, od *slave* ka *master* uređaju.
4. **SS/CS** – *Slave Select / Chip Select*, linija ili linije za izbor *slave* uređaja sa kojim želi da se komunicira. Moguće je povezati više *slave* uređaja na jedan *master*, i u tom slučaju je neophodno pomoću SS/CS linije odabrati sa kojim *slave* uređajem se komunicira. Odabir se vrši spuštanjem logičkog nivoa na SS/CS linije na logičku nulu. Kada je ova linija na logičkoj jedinici smatra se da je tada *slave* u *idle* stanju i nema prenosa podataka.



Slika 4.1.1. Povezanost *master* i *slave* uređaja

Na slici 4.1.2. je prikazan primer povezivanja više *slave* uređaja paralelno. Paralelno povezivanje je moguće samo ukoliko *master* ima više SS/CS linija. Ukoliko *master* ima samo jednu liniju onda je moguće *slave* uređaje povezati redno. Da bi više *slave* uređaja bilo redno vezano, MOSI linija *master*-a se povezuje na MOSI liniju prvog *slave* uređaja. Dalje se MISO linija *slave* uređaja vezuje na MOSI liniju sledećeg *slave* uređaja. Ovaj postupak se ponavlja do poslednjeg *slave* uređaja, čija se MISO linija povezuje na MISO liniju *master* uređaja. Ovaj način povezivanja se naziva *daisy-chain*. Samo jedan *slave* uređaj može biti aktivan u bilo kom trenutku i njega određuje isključivo *master* uređaj generisanjem signala na SS/CS liniji.

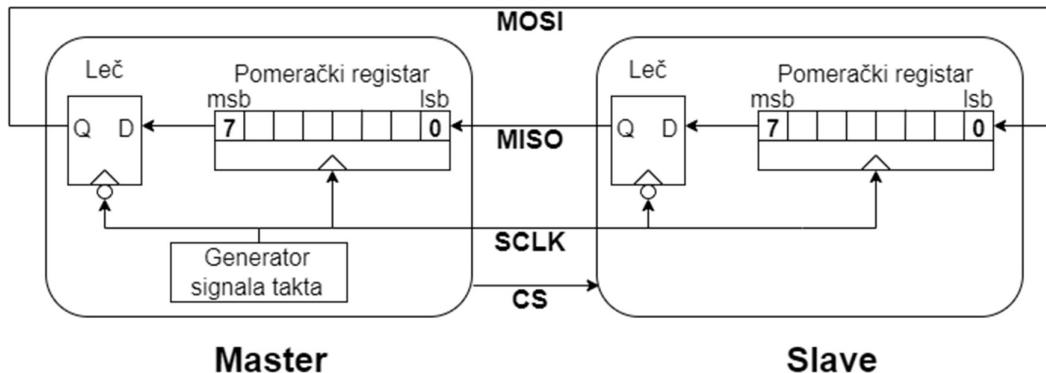


Slika 4.1.2. Povezivanje više *slave* uređaja na jedan *master*

Da bi komunikacija između uređaja započela, *master* uređaj inicira početak spuštanjem SS/CS signala na logičku nulu. Nakon toga sledi prenos podataka preko MOSI linije. Odgovor od *slave* uređaja se šalje preko MISO linije. Ukoliko se ne očekuje odgovor od *slave* uređaja, moguće je odmah nakon poslatog podatka sa MOSI linije podići vrednost SS/CS linije na logičku jedinicu. Ukoliko se ipak očekuje odgovor, vrednost SS/CS linije se podiže na logičku jedinicu tek kada se primi poruka sa MISO linije. Podaci na MISO i MOSI linijama su sinhronizovani sa signalom takta.

Kod pojedinih uređaja se komunikacija započinje podizanjem SS/CS linije na logičku jedinicu a ne njenim spuštanjem na logičku nulu. Spuštanjem na logičku nulu se završava komunikacija kod takvih uređaja. Zato je izuzetno bitno izučiti *datasheet* komponenti i proveriti za koju vrednost SS/CS pina se pokreće komunikacija.

Na slici 4.1.3. su prikazane minimalne strukture *master* i *slave* uređaja, kao i njihova međusobna povezanost. *Master* uređaj sadrži generator signala takta, pomerački registar i leč kolo, dok *slave* uređaj sadrži samo leč i registar jer mu takt diktira *master* zbog njihove međusobne sinhronizovanosti. Dok je taktni signal na logičkoj nuli, leč kolo na svoj izlaz prosleđuje najznačajniji bit (msb – *Most Significant Bit*) iz registra. Na taj način *master* i *slave* šalju jedan drugome svoj msb preko MOSI i MISO linija. Kada vrednost taktnog signala poraste na logičku jedinicu biti u registru se pomere za po jedno mesto udesno i na poziciju lsb-a (*Least Significant Bit*) se upisuje bit koji je upravo primljen od drugog uređaja. Moguće je koristiti i registre koji mogu da skladište 16 bita čime bi bilo omogućeno slanje i primanje dužih poruka između dva uređaja.

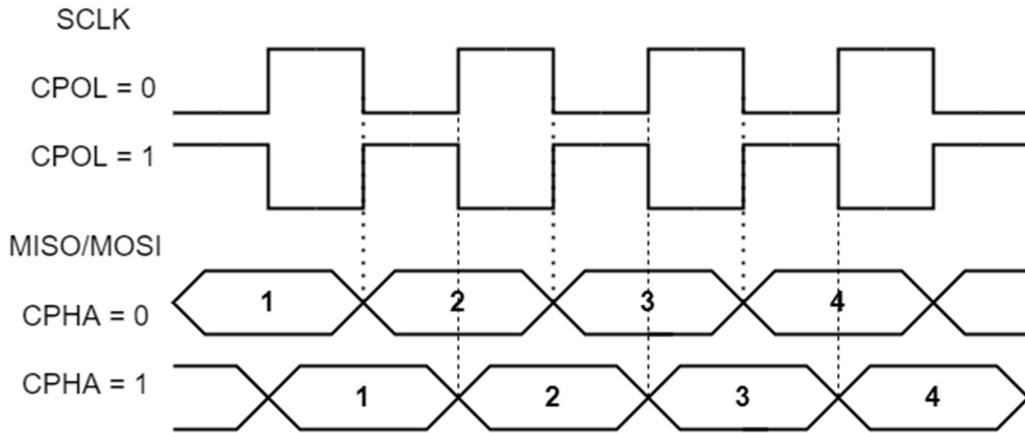


Slika 4.1.3. Minimalna unutrašnja struktura *master* i *slave* uređaja

SPI protokol nudi više različitih modova (režima rada) komunikacije. Režim rada se podešava pomoću dva parametra: polaritet taktnog signala CPOL (*Clock Polarity*) i faza taktnog signala CPHA (*Clock Phase*). Stanje signala takta zavisi od njegovog polariteta. Ukoliko CPOL ima vrednost niskog naponskog nivoa, taktni signal SCLK će imati vrednost niskog naponskog nivoa u normalnom stanju i vrednost visokog naponskog nivoa tokom prenosa podataka. Obrnuta situacija se dešava kada je vrednost CPOL na visokom naponskom nivou. Parametar CPHA govori kada će se podaci očitavati. Podatke očitavamo na prvoj ivici taktnog signala kada je vrednost CHPA = 0 ili na drugoj ivici kada je CPHA = 1. U tabeli 4.1.1. su prikazana sva četiri moda u zavisnosti od CPOL i CPHA. Slika 4.1.4. dočarava već opisan uticaj polarizacije takta na signal SCLK, kao i uticaj faze na očitavanje podataka.

Mod	CPOL	CPHA
0	0	0
1	0	1
2	1	0
3	1	1

Tabela 4.1.1. Modovi komunikacije u zavisnosti od CPOL i CPHA



Slika 4.1.4. Izgled signala u zavisnosti od CPOL i CPHA

4.2 I²C

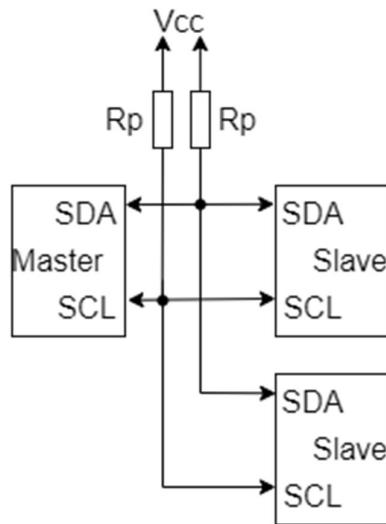
I²C (*Inter-Integrated Circuit*) komunikacija je razvijena je od strane kompanije *Phillips*. Za razliku od SPI, zahteva samo dve fizičke linije za komunikaciju: SCL (Serial Clock) i SDA (Serial Data). Ove linije su tipa otvoren drejn (*open drain*) i zahtevaju dodavanje *pull-up* otpornika tipične vrednosti između 4,7kΩ i 10kΩ (slika 4.2.1). Otpornici omogućavaju da se na izlazu *open drain* uređaja dobije visok naponski nivo i smanjuju struju curenja kroz FET (*Field Effect Transistor*) jer uređaji ovoga tipa ne mogu da dovoljno povećaju potencijal sami. Kada je na izlazu nizak naponski nivo (približno GND) tada je protok struje ograničen otpornom i računa se po formuli Omovog zakona. Iz njega je moguće dobiti formulu (1) koja određuje minimalnu vrednost otpora.

$$R_{min} = \frac{V_{CC} - V_{GND}}{I_{out_max}} \quad (1)$$

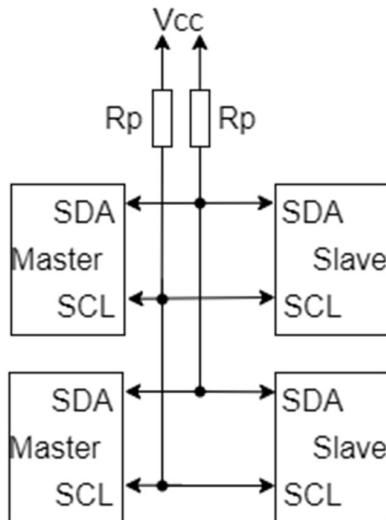
Vrednost otpornika treba biti dovoljno velika da ograniči struju kroz FET u zavisnosti od njegove maksimalne dozvoljene struje koja se navodi u *datasheet-u*. Maksimalna vrednost otpornosti je ograničena sa vrednošću kapacitivnosti magistrale – C_b i maksimalnim vremenom uspona – t_r. Obično se određuje pomoću formule (2).

$$R_{max} = \frac{t_r}{0,8473 \times C_b} \quad (2)$$

Proizvođači komponenti koje koriste I²C protokol uglavnom u dokumentaciji naglase koja je optimalna vrednost *pull-up* otpornika koji treba koristiti i nema potrebe za primenom računa.



Slika 4.2.1. Povezanost *master* i *slave* uređaja kod I²C protokola



Slika 4.2.2. *Multi-master* povezivanje

Prenos podataka kod I²C komunikacije se vrši brzinom od 100kb/s u normalnom modu, 400kb/s u brzom modu, ili do 3,4Mbit/s. Uređaji međusobno komuniciraju preko istih linija i prenos je dvosmeran. Sistem može biti *multi-master* tipa gde je omogućena komunikacija više *master* uređaja (slika 4.2.2), dok se izbor aktivnog *master*-a vrši arbitražom. Takođe je moguća i komunikacija sa više *slave* uređaja, dok za izbor aktivnog *slave* uređaja nije potrebna dodatna fizička linija. Na slici 4.2.1. je prikazano povezivanje dva *slave* uređaja sa jednim *master* uređajem. Izbor aktivnog slave uređaja se vrši slanjem njegove adrese sa master uređajem. Za razliku od SPI, komunikacija kod I²C je *half-duplex* tipa i ne mogu oba uređaja istovremeno slati podatke.

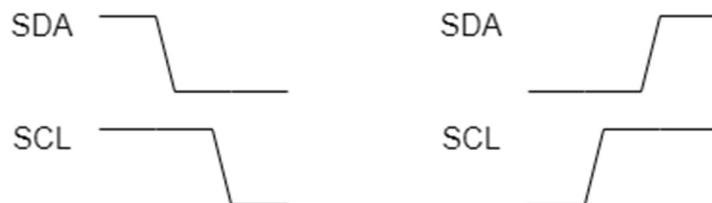
Princip slanja podataka od *master* uređaja do *slave* uređaja je prikazan na slici 4.2.3. Sivom bojom su označeni bitovi koje šalje *master*, dok su belom označeni odgovori od *slave* uređaja.

Početak prenosa inicira *master* uređaj slanjem S bita (*Start* – početak). Nakon toga *master* šalje 7-bitnu (ili 10-bitnu) adresu *slave* uređaja sa kojim želi da komunicira. Potrebno je da *master* nakon toga javi da li želi da čita podatke (R – *Read*) ili da ih šalje (W – *Write*). Adresirani *slave* uređaj šalje signal potvrde (ACK – *ACKnowledge*) čime javlja da li je adresiran i da li je primio R/W bit. *Master* uređaj sada može da šalje ili prima podatke. Ukoliko *master* šalje podatke, *slave* uređaj je dužan da mu nakon svakog primljenog bajta podataka potvrdi da li je primio podatke. Ova sekvenca slanja podataka i potvrde se može ponavljati dok se ne pošalju svi željeni podaci. Ako bi ipak *master* uređaj htEO da čita podatke, jedine razlike u postupku bi bile u tome što bi sada *slave* uređaj slao podatke dok bi mu *master* potvrđivao da ih je primio. Na kraju je potrebno da *master* pošalje P (*stopP*) kao zaustavni bit čime naglašava da je prenos završen.

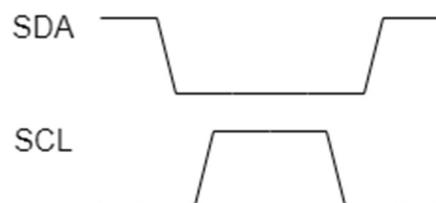


Slika 4.2.3. Slanje podataka kod I²C komunikacionog protokola

Bit početka (S) se realizuje spuštanjem naponskog nivoa SDA linije sa visokog na niski (slika 4.2.4. levo), dok se bit završetka (P) realizuje podizanjem naponskog nivoa SDA linije (slika 4.2.4. desno). Tokom ovih prelaza je potrebno da linija SCL bude na visokom naponskom nivou. R/W bit će imati vrednost 1 ukoliko *master* želi da čita podatke, odnosno 0 ukoliko želi da ih šalje. Potvrda prijema se vrši spuštanjem naponskog nivoa SDA linije sa visokog na niski tokom perioda jednog taktnog signala (slika 4.2.5). ACK signal se neće spustiti na niski naponski nivo ako poruka od *master* uređaja nije uspešno primljena.



Slike 4.2.4. Izgled signala pri slanju *start* (slika levo) i *stop* (slika desno) komande



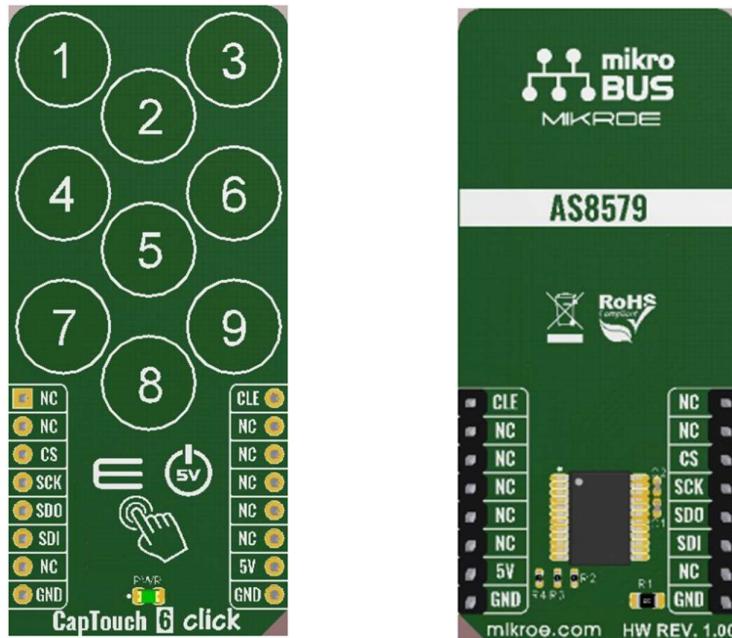
Slika 4.2.5. Slanje signala potvrde

5. Kreiranje *Click* pločica

U cilju izrade završnog rada biće objašnjen postupak kreiranja dveju različitih pločica – *CapTouch 6 Click* i *LightRanger 9 Click*. Pločice su dizajnirane pomoću softverskog alata *AltiumDesigner*.

5.1 *CapTouch 6 Click*

Pločica *CapTouch 6 Click* predstavlja kapacitivni senzor dodira. Realizovana je kao tastatura koja na sebi ima 9 tastera koji reaguju na dodir. Na slici 5.1.1. levo je prikazan 3D prikaz konačnog izgleda pločice gledano sa prednje strane, dok je na desnoj strani slike prikazan izgled pločice gledano sa zadnje strane.

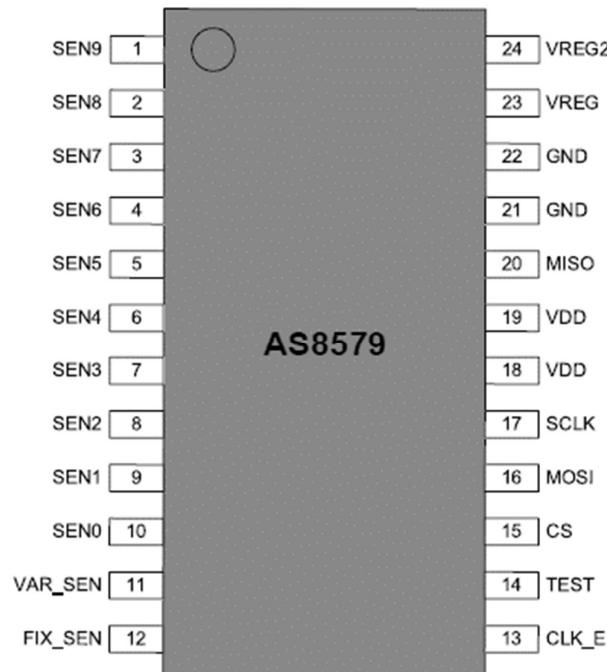


Slika 5.1.1. Izgled *CapTouch 6 Click* pločice, gledano sa prednje strane (slika levo), i gledano sa zadnje strane (slika desno)

5.1.1 Opis senzora AS8579

Glavna komponenta koja je korišćena pri izradi *CapTouch 6 Click* pločice je *AS8579*. Komponenta je kreirana od strane Austrijske kompanije *ams OSRAM*, koja se bavi proizvodnjom elektronskih komponenti visokih performansi. *AS8579* je kapacitivni senzor koji ima primenu u različitim uređajima za detekciju promene kapacitivnosti. Senzor meri relativnu promenu kapacitivnosti. Za komunikaciju koristi SPI protokol.

Princip rada kapacitivnih senzora na dodir je izuzetno jednostavan. Kapacitivnom senzoru dodira približimo prst, ili neki drugi objekat koji sadrži kapacitivne karakteristike. Prst ili objekat, zbog svoje dielektrične prirode, deluje kao još jedan kondenzator povezan sa senzorom.

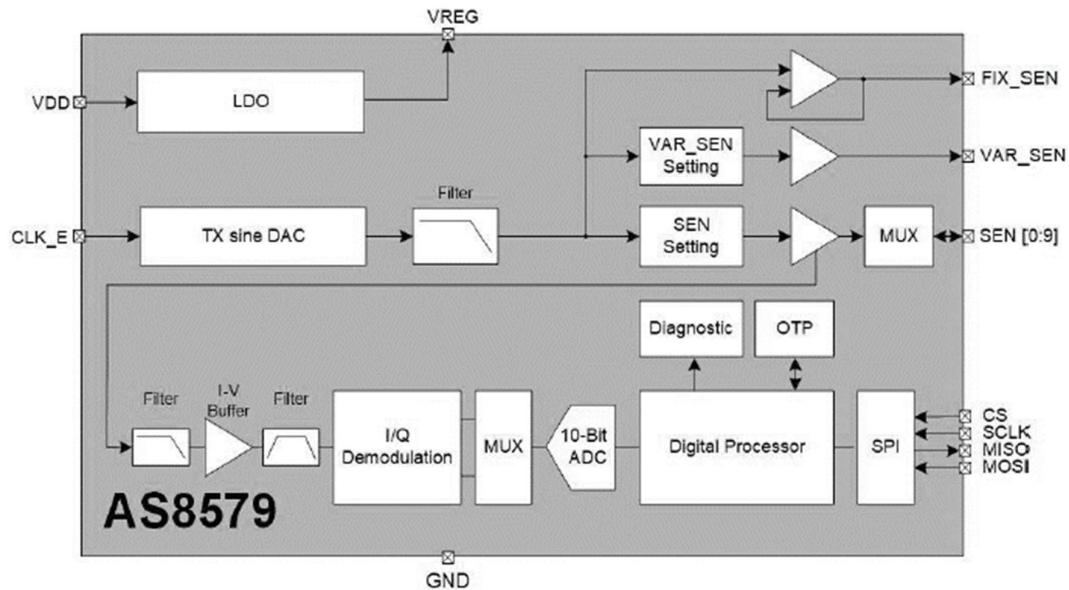


Slika 5.1.1.1. Pinovi komponente *AS8579*

Izvor: Datasheet komponente, www.ams.com

Na slici 5.1.1.1. je prikazan raspored pinova ove komponente, zajedno sa njihovim nazivima. Pinovi *SEN0* - *SEN9* predstavljaju 10 zasebnih senzorskih linija, tipa *Analog Input / Output*, skraćeno *AIO*. Ovim je omogućeno merenje kapacitivnosti u 10 različitih tačaka pomoću samo jednog senzora. Pinovi *VAR_SEN* i *FIX_SEN* su opisani kao *Cable Shielding Driver* i *PCB Shielding Driver* pinovi. Oni su takođe tipa *AIO*, i njihova uloga je takva da se na ova dva pina mogu povezati oklop kabla (spoljni sloj) i / ili zaštitni sloj PCB-a, čime je moguće otkloniti uticaj parazitnih kapacitivnosti koji potiču od njih. Pin *VREG* predstavlja interni (unutrašnji) izvor napona vrednosti 3,45V, i potrebno ga je spojiti sa pinom *VREG2*. *GND* i *VDD* su pinovi za napajanja. Na *VDD* je potrebno dovesti eksterno (spoljašnje) napajanje vrednosti 5V. *MOSI* i *MISO* su pinovi za *SPI* komunikaciju. Imamo dva pina za takt, i to su *CLK_E*, na koji se dovodi sistemski takt signal, i *SCLK* za taktni signal *SPI* komunikacije. *CS* je *SPI Chip Select* pin. I na kraju, pin *TEST* služi kao *Factory Test Pin*, i po podacima iz *datasheet-a* ga treba povezati na

masu. *MISO* je pin tipa *Digital Output*, a svi ostali pinovi kojima već nije naznačen tip (osim napajanja) su tipa *Digital Input*.



Slika 5.1.1.2. Unutrašnja struktura komponente AS8579

Izvor: Datasheet komponente, www.ams.com

Na slici 5.1.1.2. je prikazana unutrašnja struktura komponente AS8579. Arhitektura prenosa je *AFE (Analog Front End)*. Detektovanje se vrši pomoću predajnih i prijemnih blokova. Predajni blok snabdeva opterećenje sinusnim signalom. Prijemnik registruje promene struje usled promene opterećenja. Promene struje se konveruju u napon, zatim se vrši njihova demodulacija na *in-phase* (infazna, u fazi) i *quadrature* (kvadraturne), to jest, I i Q komponente². Dve komponente se zatim filtriraju i konvertuju u 10-bitnu digitalnu reč pomoću analogno / digitalnog konvertora. Ove I i Q reči se akumuliraju i čekaju SPI komunikaciju.

Stabilnu vrednost internog napona *VREG* uzimamo sa izlaza *LDO* (*Low-dropout*) regulatora. *LDO* je linearni regulator jednosmernog napona, koji na svom izlazu daje stabilnu vrednost napona, i na njegov ulaz dovodimo eksterno napajanje *VDD*.

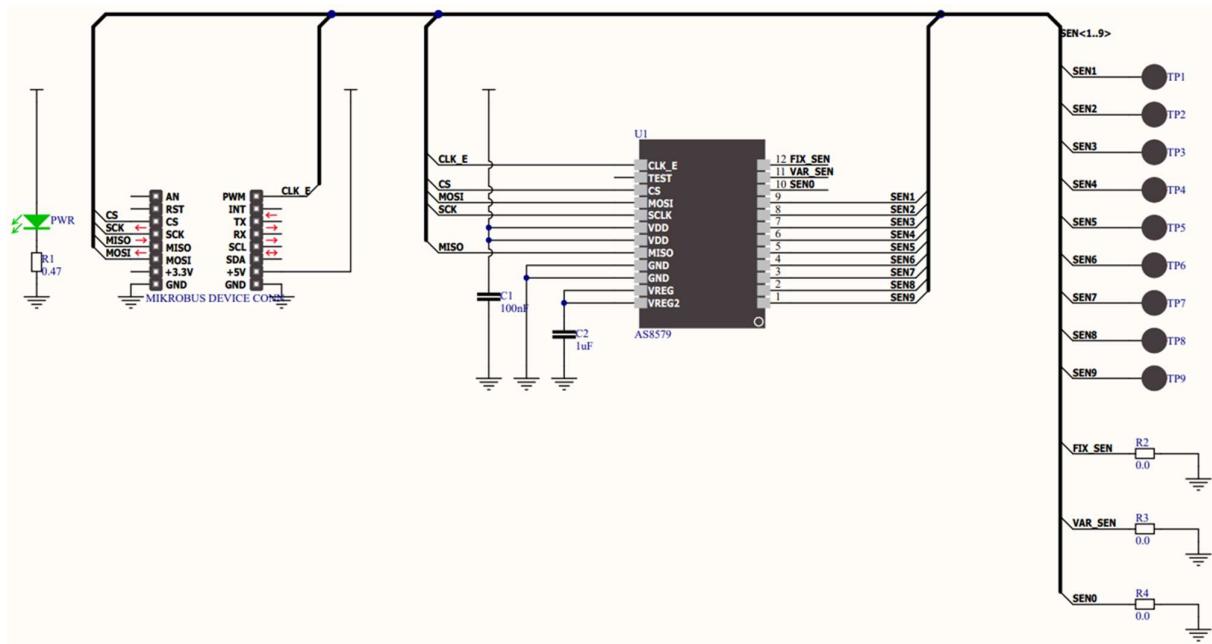
Sistemski taktni signal se konvertuje u analogni pomoću *DAC (Digital / Analog Converter)*, nakon čega se filtrira niskopropusnim filtrom. Ostali podsklopovi u kolu se sinhronizuju sa sistemskim taktnim signalom.

SEN pinovi su preko deset trokanalnih multipleksera povezani sa senzorskim drajverom, i njihove vrednosti se memorišu u registru. Multiplekseri se kontrolisu pomoću eksternog procesora. Procesor dobija I i Q komponente za svaki senzor, i određuje impedansu opterećenja.

Pomoću SPI protokola se komunicira između digitalnog procesora komponente i kontrolera kojim programiramo komponentu.

² Dva sinusoidalna signala iste frekvencije koji su međusobno pomereni za 90°

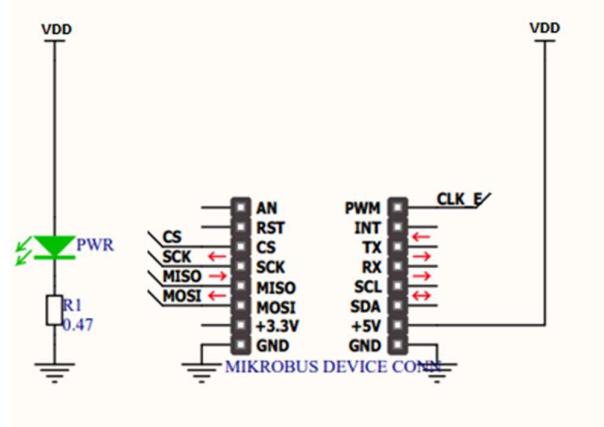
5.1.2 Dizajn šematička pločice *CapTouch 6 Click*



Slika 5.1.2.1. Kompletan dizajn šematička za *CapTouch 6 Click*

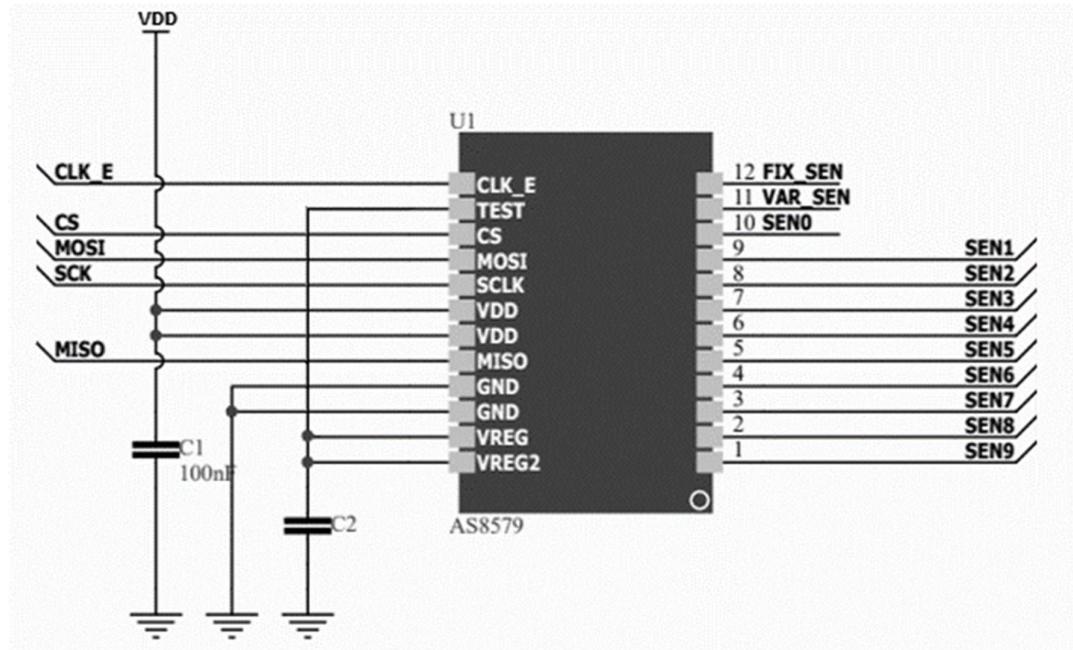
Na slici 5.1.2.1. je prikazan ceo dizajn šematička za *CapTouch 6 Click* pločicu. Dizajn se može podeliti u tri celine. Prvu čini konektor za *mikroBUS*, zajedno sa signalnom svetlećom diodom. Druga celina je glavna komponenta, to jest, *AS8579*. Treću čine *Touch Pad*-ovi (bakarne površine za detekciju promene kapacitivnosti) i povezanost pinova za eliminisanje uticaja parazitnih kapacitivnosti.

Slika 5.1.2.2. bolje prikazuje prvu celinu, odnosno povezanost konektora. Analizom komponente koja je uradena u odeljku 5.1.1., može se doći do zaključka da su nam sa konektora potrebni samo pinovi za *SPI*, napajanje i takt. Za napajanje komponente uzimamo napon vrednosti 5V sa priključka. Za *SPI* protokol su nam potrebni CS, SCK, MOSI i MISO pinovi. Krajnje, signal takta komponenti šljemo preko PWM pina sa konektora. Pored konektora imamo signalnu zelenu diodu *PWR*. Ova dioda je indikator napona, i biće upaljena ukoliko je na pločicu doveden napon.



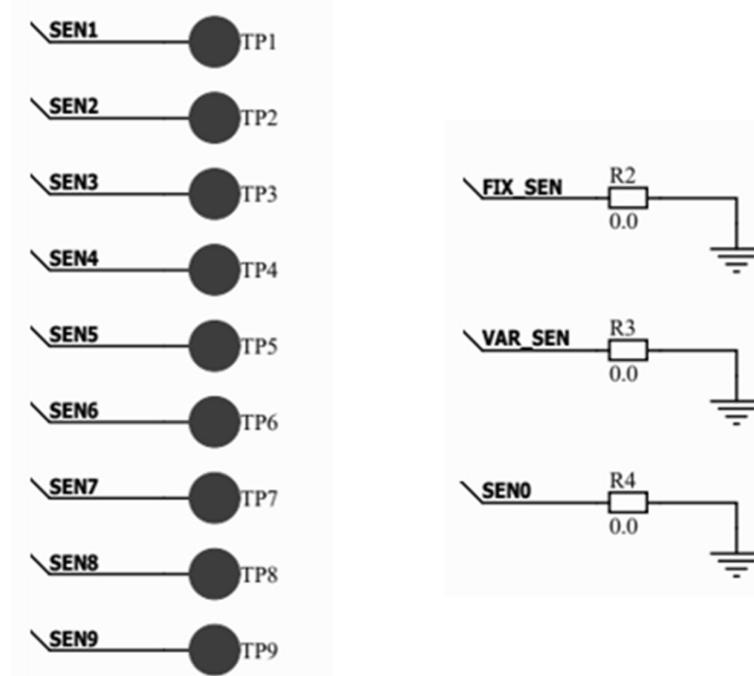
Slika 5.1.2.2. Povezanost *mikroBUS* priključka

Sledeća celina je prikazana na slici 5.1.2.3. Na pinove za *VDD* je doveden napon od 5V, uzet sa konektora. Između *VDD* - *GND* para je dodat po *decoupling* (dikapl) kondenzator, vrednosti 100nF. Dodat je još jedan dikapl kondenzator od napona *VREG* ka masi. Dikapl kondenzator smanjuje uticaj signala visokih frekvencija koji mogu slučajno nastati u kolu i sprečava pojave oscilacija. *TEST* pin je, po informacijama iz *datasheet-a* povezan na masu. Sa pinova za *SPI* i za *CLK_E* se izvlače vodovi i povezuju se sa konektorom. Senzorske pinove vodimo na treći deo kola.



Slika 5.1.2.3. Povezanost komponente AS8579

Na kraju, treću celinu čine *Touch Pad*-ovi i otpornici, prikazani na slici 5.1.2.4. Obzirom da je ideja pločice da ima 9 tastera, a komponenta pruža mogućnost implementiranja 10 različitih senzora, jedan od senzora (senzor 0) se, preko otpornika od 0Ω vodi na masu i njega u razvoju softvera zanemarujemo. Na pločici fizički nije bilo mesta za implementaciju desetog peda. U slučaju da je potreban i deseti, može se dodati eksterno, i spojiti na vod pina *SEN0* umesto otpornika. Kako je potrebno detektovati da li je taster pritisnut, na ulaze senzora su dovedeni mali pedovi. Pedovi su povezani na pinove senzora, ne zahtevaju dodatne komponente, i realizovani su tako što svaki obuhvata površinu unutar kruga. Da bi se mogla ispravno vršiti detekcija, cela unutrašnjost kruga je obložena bakrom. Dakle, svaki ped predstavlja bakarnu površinu oblika kruga, kojem se menja kapacitivnost kada se približi objekat (prst). Opcije koje pružaju pinovi *FIX_SEN* i *VAR_SEN* nisu iskorišćene, nego su i oni povezani na masu preko otpornika nulte otpornosti. Ukoliko korisnik odluči, može ih prespojiti radi otklanjanja uticaja parazitnih otpornosti kabla i pločice.

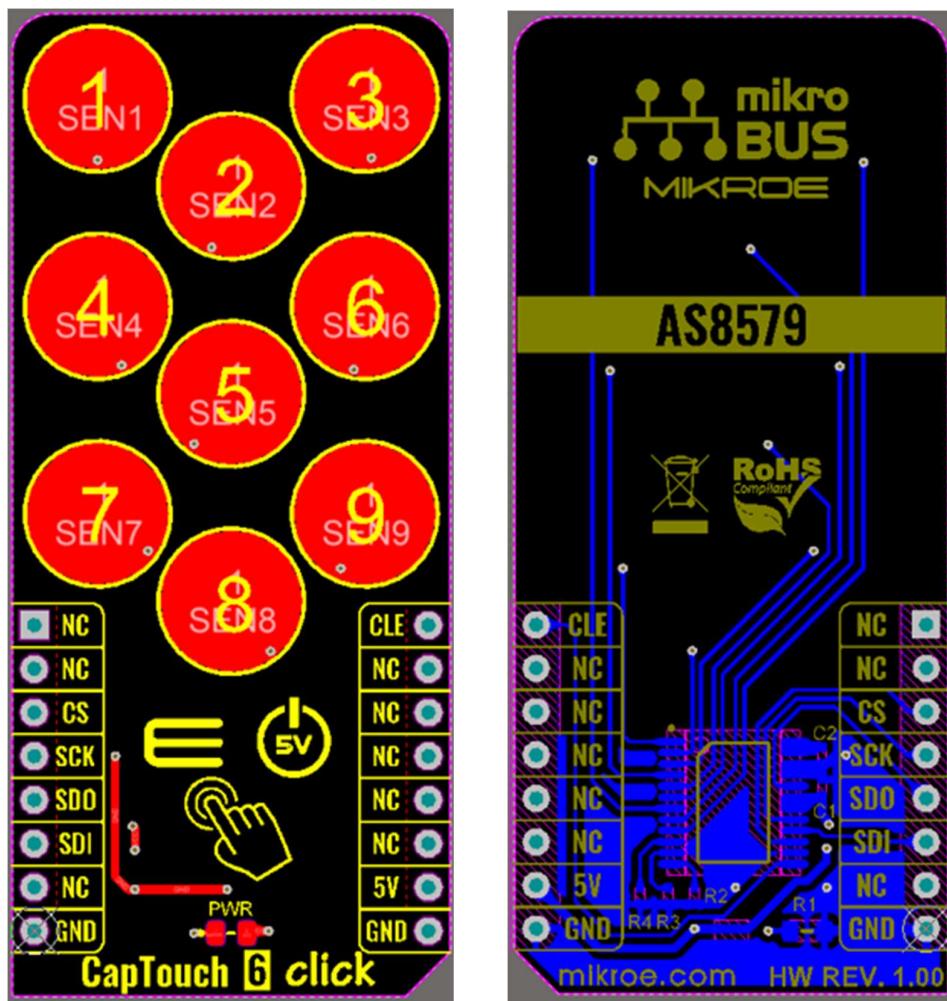


Slika 5.1.2.4. Povezanost TouchPad-ova i otpornika

5.1.3 PCB dizajn pločice CapTouch 6 Click

Kada je uspešno i ispravno kreiran šematički povezani, potrebno je napraviti njegov PCB dizajn. Dizajn PCB-a koji se kreira je od izuzetne važnosti, jer on služi za štampanje pločice. Šteta u finansijskom smislu koja nastaje prilikom lošeg PCB dizajna može biti velika. Ukoliko se neki uređaj ne dizajnira ispravno i pošalje se u proizvodnju, dobija se veliki broj uređaja koji ne služi ničemu i nije primenljiv, i kao takav ne može da se pošalje na tržište. Uređaji mogu sadržati skupe komponente, i loša serija proizvodnje dovodi do velikih finansijskih gubitaka. Zato je izuzetno važno da se sve komponente postave i povežu u skladu sa šematičkim, da se vodi računa o debljinama vodova, rasporedu komponenti... Provera dizajna pomoću softverskog alata je neophodna da bi se izbegle potencijalne greške.

Na slici 5.1.3.1. je prikazan konačni izgled PCB dizajna gornje i donje strane pločice. Za kreiranje pločice je uzeta L veličina pločice, obzirom da je potrebno mnogo prostora za pedove. Sve komponente su dodate sa donje strane, omogućavajući više mesta na gornjoj strani pločice za pedove. Dimenzije pedova su određene eksperimentalno. Širina pločice iznosi 1000 mila. U jedan red treba da stanu dva peda, i delić trećeg. Ako se širina pločice podeli na tri dela, dobija se vrednost od 333 mila. Obzirom da u redu nisu tri cela peda, pošto je srednji pomeren na dole, moguće je ovu brojku povećati. Na kraju, korišćeni prečnik za pedove je iznosio oko 350 mila.



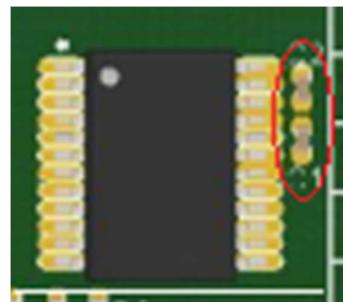
Slika 5.1.3.1. Izgled završenog PCB dizajna gornje strane pločice (slika levo) i dizajna donje strane pločice (slika desno) *CapTouch 6 Click*

Komponenta *AS8579* je postavljena u donjem delu pločice radi skraćivanja dužine vodova SPI protokola. Pinovi za SPI komunikaciju su na priključku označeni sa *SDO* (*Serial Digital Output*), *SDI* (*Serial Digital Input*), *SCK* (*Serial Clock*), i *CS*. Vodovi sa digitalnim i taktnim signalima su česti izvori šuma, i povećavanjem njihove dužine se dobija više šuma. Iz ovog razloga je komponenta postavljena tako da ovi vodovi budu što manje dužine.

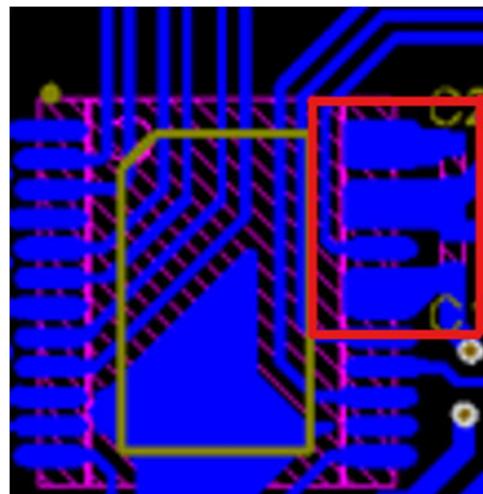
U gornjem delu pločice su razvlačeni samo vodovi do pedova. Ukoliko bi na gornjoj strani postojali i vodovi kroz koje protiče fisokofrekventni sadržaj, on bi mogao da utiče na promenu kapacitivnosti samih pedova. Zbog načina rasporeda pedova i zbog dimenzija pločice, nije bilo moguće izbeći da se vodovi koji se povezuju sa pedovima na donjoj strani ne provlače ispod drugih pedova. Ovo može dovesti do nepoželjnih malih promena kapacitivnosti na pedovima koji nisu dodirnuti. Rešenje za ovaj problem bi bilo softversko otklanjanje uticaja vodova na male promene kapacitivnosti.

Dikapl kondenzatori su postavljeni što je moguće bliže pinovima za napajanje na glavnoj komponenti, kao na slici 5.1.3.2. Ukoliko se dve nožice nalaze jedna pored druge, i pripadaju istoj mreži, umesto njihovog povezivanja samo jednim vodom, povezuju se poligonom. Poligon stvara

kvalitetniju vezu od linije voda jer teže dolazi do prekida kontakta. Način povezivanja susednih istomrežnih pinova je prikazan na slici 5.1.3.3. i uokviren je crvenom bojom. Pinovi povezani poligonima na slici su pinovi za napajanja (interno, masa, eksterno), te može doći do pregrevanja usled protoka struje ukoliko se za njihovo povezivanje vrši tankim vodovima. Njihovim povezivanjima pomoću poligona se kreira veća površina po kojoj se raspoređuje protok struje, pa je samim time grejanje smanjeno.

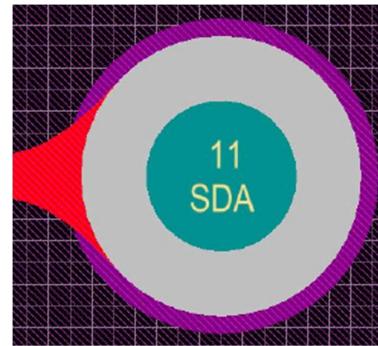


Slika 5.1.3.2. Postavljanje dikapl kondenzatora

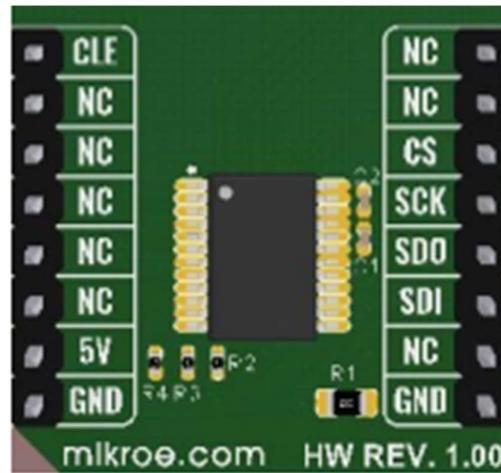


Slika 5.1.3.3. Povezivanje susednih istomrežnih pinova

Mogućnost dolaska do prekida voda kod mesta spajanja sa pinom se značajno smanjuje implementacijom "suze" (*Teardrop*) u *AltiumDesigner* alatu. Naime, prilikom rutiranja, kada se povezuje vod sa pinom, dobija se ravna linija koja izlazi iz pina pod uglom od 90 stepeni. Ovakav spoj je sklon prekidima usled nagrizanja pri štampi pločice. Prekid kod ovakvih spojeva se može desiti i pri bušenju rupa u pinovima, gde bušilica može slučajno da oguli deo bakra oko rupe. Primenom opcije *Teardrop* se kreira jača veza između pina i voda. Veza više nije pod pravim uglom, nego se vod gradijentno širi i obuhvata pin pod većim uglom. Izgled spoja voda i pina nakon primene *Teardrop* opcije prikazan je na slici 5.1.3.3.



Slika 5.1.3.3. Primer primene opcije *Teardrop* u dizajnu



Slika 5.1.3.4. Raspored komponenti sa donje strane pločice

Raspored svih komponenti na donjoj strani pločice prikazan je na slici 5.1.3.4. Otpornik R1 ograničava struju za signalnu diodu koja se nalazi sa gornje strane pločice. Obzirom da se pinovi *FIX_SED*, *VAR_SEN* i *SEN0* nalaze na donjoj levoj strani komponente, u tom predelu pločice su postavljeni i otpornici preko kojih ove pinove spajamo na masu.

Širine vodova pri rutiranju zavise od jačine struje koja kroz njih teče. Ukoliko kroz vod teče veća struja, njegovu širinu je potrebno povećati. Na ovaj način, slično kao u primeru povezivanja pinova za napajanje poligonom, povećava se površina kroz koju protiče struja i smanjuje se mogućnost pregrevanja. Vrednosti struja koje se mogu očekivati u kolu se proveravaju u *datasheet*-u. Analizom podataka o komponenti se može videti da struje ne prelaze vrednosti od 50mA, te svi vodovi, osim za napajanje, mogu biti manje debljine. Pri rutiranju je važno dodeliti veću širinu vodova pri povezivanju napajanja. Ovo je opšte pravilo pri rutiranju jer kroz vodove napajanja protiče veća struja. Zato se pri rutiranju obično definišu dve klase za rutiranje – klasa za signalne linije i klasa za linije napajanja. Obzirom da se otpornost voda računa po formuli (3),

$$R = \rho \cdot \frac{l}{s} [\Omega], \quad (3)$$

debljina voda utiče na njegovu otpornost. Potrebno je dobiti vodove što manje otpornosti, te se pri rutiranju vodi računa da dužina vodova bude što manja, dok širinu treba povećati. Mana

preteranog povećanja širine vodova je njihovo teže provlačenje pored komponenti. Deblji vodovi zauzimaju više mesta na pločici, i može se desiti da fizički nema mesta da se svi vodovi razvuku po sloju pločice. Zato je potrebno napraviti kompromis između raspoloživog prostora i debljine vodova.

Na izbor širine voda utiču još i količina bakra i porast temperature na PCB-u. Tabela 5.1.3.1. prikazuje zavisnost širine vodova od maksimalne jačine struje i porasta temperature za 10 i 20 °C. Tabela je rađena za količinu bakra od 1oz (unca). Osim tabela, postoje i softverski alati, kao i kalkulatori na internetu koji računaju optimalnu širinu vodova, u zavisnosti od unetih parametara (dužina voda, količina bakra, porast temperature, jačina struje...).

Širina [mil]	Maksimalna jačina struje pri porastu temperature od 10 °C	Maksimalna jačina struje pri porastu temperature od 20 °C
10	1	1.2
15	1.2	1.3
20	1.3	1.7
25	1.7	2.2
30	1.9	2.5
50	2.6	3.6
75	3.5	4.5

Tabela 5.1.3.1. Zavisnost širine voda od porasta temperature i maksimalne struje

Za preferiranu širinu vodova signalnih linija uzeta je vrednost od 30 mila, a za najmanju vrednost širine je uzeto 10 mila. Najmanja širina se koristi u rutiranju samo na mestima gde nije fizički izvodljivo provući vod od 30 mila. Ovom kolu je potrebna širina vodova koja može da izdrži struju do 50 mA. Iz tabele vidimo da širine od 10 i 30 mila odgovaraju i mnogo većim strujama. Rizik od pregravanja je izbegnut, a izborom veće vrednosti za širinu vodova je smanjena njegova otpornost. Za vodove napajanja je izabrana širina od 50 mila.

Za povezivanje sa masom, u donjem delu pločice, na zadnjoj strani, je izliven poligon. Obzirom da se poligon sa uzemljenjem prostire oko ostalih vodova, on smanjuje uticaj šumova između vodova. Pomaže odvođenju i ravnomernom rasporedu toplote. Pomoću poligona su uspešno povezani skoro svi delovi pločice koje je potrebno uzemljiti. Sa jednim dodatnim vodom na gornjoj strani pločice, koji se pomoću vija na tri mesta povezuje sa poligonima na donjoj strani, svi poligoni su međusobno povezani.

5.1.4 Primena *CapTouch 6 Click* pločice

Jedna od bitnijih stavki pri kreiranju svakog uređaja je upravo njegova primena. Ukoliko uređaj nema svrhu, korisnici ga neće kupovati, i uređaj će samo predstavljati nepotreban trošak finansija i komponenti. Pri osmišljavanju određenog uređaja od izuzetne je važnosti kako njegova jedinstvenost tako i mogućnost široke primene istog. Što je veća unikatnost nekog uređaja, to će se on više prodavati na tržištu. Ukoliko postoji veliki broj uređaja iste klase, na primer,

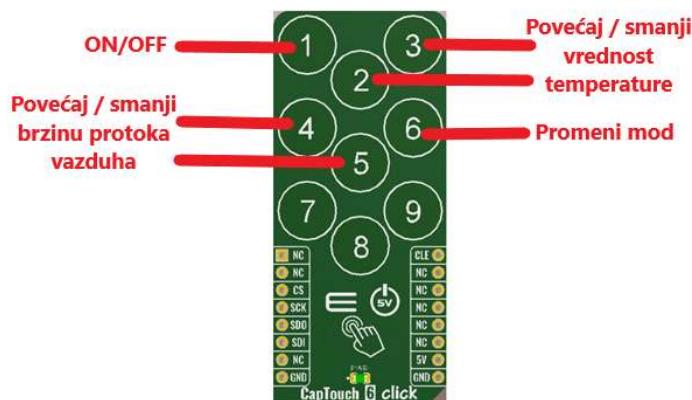
termometara, moguće je napraviti novi, ali se on mora razlikovati od prethodnih. Razlike mogu biti bolje performanse (brzina rada, cena, preciznost), manje dimenzije, promena korišćenog protokola za komunikaciju, ili po nekoj drugoj želji korisnika.

Kreirana pločica *CapTouch 6* u svom nazivu sadrži broj 6, koji označava da je pre ove pločice kreirano 5 sličnih pločica u firmi *Mikroelektronika*. Osim ove serije, postoji serija *TouchKey Click* pločica, koja je slična ovoj. Svi prethodno kreirani *Click*-ovi su sadržali tastature, slajdere (površine za detekciju prevlačenja), ili kombinacije. Neke pločice su imale okrugle brojeve, dok su druge imale kockaste. Za svaku kreiranu pločicu je korišćena drugačija komponenta. Od svih pločica koje su do tad bile kreirane za potrebe firme, ni jedna nije imala na sebi devet tastera. Upravo to čini ovu pločicu drugačijom od ostalih.

Prvenstvena ideja je bila napraviti tastaturu na kojoj te nalaze brojevi od 0 do 9, sličnu tastaturnama za unošenje pin brojeva, ali zbog fizičkih dimenzija pločice to nije bilo moguće. Pločica je ostala sa 9 tastera, što je i dalje čini unikatnom i različitom od ostalih. Sa svojih 9 različitih kapacitivnih senzora i mogućnošću osposobljavanja desetog, pločica ima širok spektar primene.

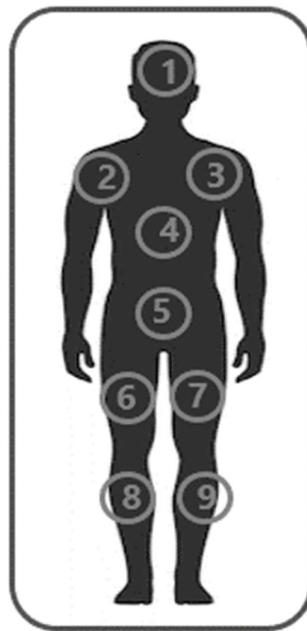
Moguće je korišćenje pločice kao tastature za brojeve, ukoliko naravno nije neophodno korišćenje broja 0. Obzirom da tasteri nije potrebno fizički pomeriti, to jest, nema mehaničkog procesa pri njihovom pritiskanju, smanjena je mogućnost mehaničkog habanja. Tastature poput onih na bankomatima imaju tasterne koje je potrebno fizički pomeriti (kliknuti) da bi se detektovalo pritiskanje. Kod takvih tastatura postoji mogućnost zaglavljivanja tastera. Ulazak prašine pod tasterne može otežati ili sprečiti registrovanje pritiskanja tastera od strane mašine. Takve tastature bi zahtevale redovno čišćenje, poput tastatura za rad na računарима. Na *CapTouch* tastaturi nema fizičkog pomeranja tastera, te ne može doći do gore navedenih problema.

Ova tastatura se može koristiti i u upravljačke svrhe. U zavisnosti od pritisnutog tastera, dešavaće se drugačija operacija. Bolje objašnjenje se može dati na konkretnom primeru. Recimo da želimo da upravljamo klima uređajem. Ukoliko se pritisne taster 1, klima se pali. Pomoću tastera 2 i 3 se može povećavati i smanjivati željena vrednost temperature. Tasteri 4 i 5 se mogu koristiti za podešavanje brzine vazduha koji izlazi iz klima uređaja, a pomoću tastera 6 se može podešavati mod u kome se klima nalazi. Ovaj princip je prikazan na slici 5.1.4.1. Ovo je samo jedan od primera u kome je moguće koristiti tastaturu u upravljačke svrhe.



Slika 5.1.4.1. Primena *CapTouch 6 Click* pločice u upravljačke svrhe

Pločica se može koristiti i za detekciju prisustva osobe. U bolnicama se sve više praktikuju senzori za detekciju pacijenata. Ukoliko pacijent, na primer, ustane iz kreveta, senzor javlja nadležima koji mogu da reaguju. Da bi se *CapTouch 6 Click* iskoristio u svrhu detekcije kretanja pacijenata, potrebno je pedove povezati sa pločicama ili folijama od metala koje bi se postavile na krevet. Devet zasebnih senzora se pomoću tih folija ili pločica mogu rasporediti po krevetu tako da se u zavisnosti od očitavanja može zaključiti u kom položaju je pacijent. Jedan od primera rasporeda senzora je prikazan na slici 5.1.4.2. Prvi senzor se stavi na krevet na poziciji ispod glave pacijenta, dva se stave ispod ramena, jedan ispod leđa, i ostali ispod donjeg dela tela. U slučaju da senzor detektuje pritisak svuda, osoba leži. Ako senzor ipak detektuje pritisak samo na senzorima kod donjeg dela (senzori od 5 do 9), znači da je osoba sela. Neki teži pacijenti ne bi smeli ustajati iz ležećeg položaja, te bi se programu naglasilo da obavesti nadležne ukoliko je pacijent seo, ili čak skroz ustao iz kreveta. Ako se detektuje pritisak samo na desnoj strani kreveta (senzori 3, 7, 9), znači da se pacijent okrenuo, što može biti nezgodno u slučaju da je priključena infuzija, da pacijent ima povredu na desnoj strani tela na koju se ne sme vršiti pritisak, ili bilo koji slučaj gde ovaj položaj ležanja nije preporučljiv i treba obavestiti nadležne. Moguće je napraviti i drugačiji raspored senzora i prilagoditi ga pacijentu i njegovom stanju i povredama. Pomoću programa naglašavamo koja pomeranja su kritična kod svakog pacijenta i pri kojim detekcijama će se alarmirati nadležni.



Slika 5.1.4.2. Primena *CapTouch 6 Click* pločice za detekciju pacijenata

Detekcija prisustva se može vršiti i u automotivu. Kao i u prethodnom primeru, senzorski kontakti se izvlače i ugrađuju u sedišta. Ovo omogućava kreiranje bezbednosnog sistema za sigurnosne pojaseve. Pri paljenju vozila se proverava prisutnost osoba na sedištima pomoću očitavanja na senzorima, zajedno sa još jednim sistemom koji proverava da li je sigurnosni pojasa povezan. Ukoliko dođe do detekcije osobe, ali ne i do detekcije priključenosti pojasa na istom

sedištu, generisao bi se zvučni signal upozorenja. Osim u klasične automobile sa pet sedišta, sistem bi se mogao ugraditi i u veća vozila koja sadrže do devet sedišta.

Još jedna moguća primena u automotivu je detekcija šaka na volanu. Nažalost, broj saobraćajnih nezgoda raste iz dana u dan. Jedan od faktora koji utiče na ovaj porast predstavlja umor vozača. Umor rezultuje smanjenom percepcijom, oslabljenim refleksima kao i opuštanjem mišića. Pri opuštanju mišića vozač mnogo slabije pridržava volan. Praćenje stiska na volanu omogućava proveravanje stanja (umora) vozača i njegovo upozoravanje da bi trebao što pre da završi sa upravljanjem vozila. Osim umornih, postoje i vozači koji ne poštuju propise o držanju volana pri vožnji. Pridržavanje volana sa jednim ili dva prsta samo jedne ruke, ili potpuno odvajanje ruku od volana je izuzetno opasno, pogotovo ukoliko vozač nije iskusan. Pri malo većem neočekivanom pomeranju vozila, poput prelaska preko kamena, rupe ili bankine, auto se može oteti kontroli. Ukoliko se volan ne drži propisno, i vreme reakcije vozača ne bude brzo, ovo može rezultovati nesrećom. Ovo je dovelo do sve češćeg ugrađivanja kapacitivnih senzora u volane radi praćenja hvata na samom volanu, kao i pritiska koji vozač ima na volan zbog detekcije njegovog umora. Pri slabijoj detekciji pritiska, ili pri detekciji lošeg držanja volana, oglašava se alarm. U ovu svrhu moguće je razvući pedove sa pločice i ugraditi ih u volan, kao na slici 5.1.4.2. U programu bi se definisalo da li je potrebno vršiti detekciju umora (jačine stiska volana), držanja volana ili oba.



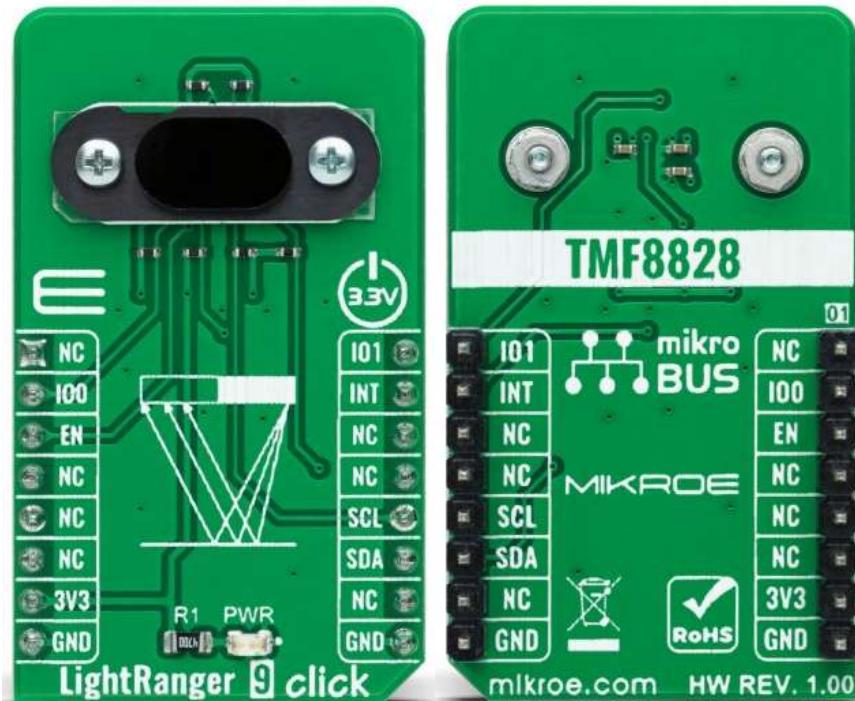
Slika 5.1.4.2. Primena CapTouch 6 Click pločice za detekciju držanja volana

Obzirom da je senzor kapacitivni, može se koristiti i za detekciju dielektričnih materijala.

Ovo su samo neki od primera primene kreirane pločice. Korisnicima se ostavlja sloboda izbora konkretnе svrhe korišćenja pločice. Najpreporučljivije upotrebe su upravo za detektovanje dodira, prisustva, kontakta ili dielektričnih materijala.

5.2 LightRanger 9 Click

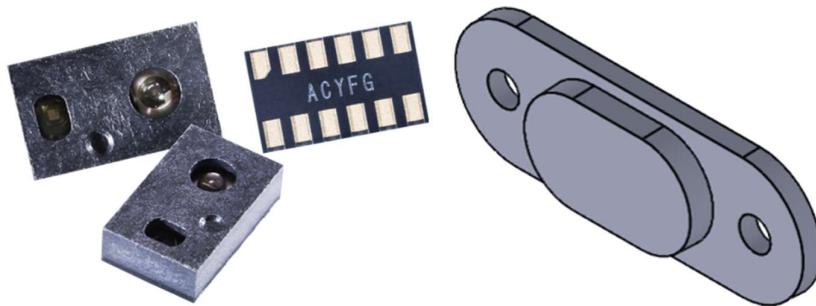
Druga pločica koja je kreirana je *LightRanger 9 Click*. Ova pločica služi za merenje rastojanja od pločice do prepreke. Bazira se na merenju vremena preleta. Pločica je izrađena od strane firme *Mikroelektronika*, i izgled njene prednje i zadnje strane se nalazi na slikama 5.2.1.



Slike 5.2.1. Izgled prednje (slika levo) i zadnje strane (slika desno) gotove pločice

Izvor: zvaničan sajt *Mikroelektronike*, www.mikroe.com

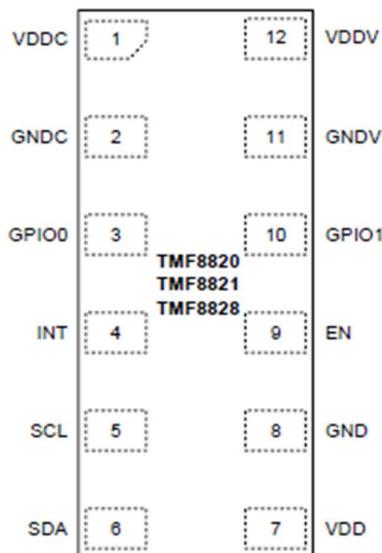
5.2.1 Opis senzora TMF8828



Slika 5.2.1.1. Izgled komponente *TMF8828* (slika levo) i zaštitnog stakla (slika desno)

Glavnu komponentu na pločici predstavlja višezonski optički senzor *TMF8828* prikazana na slici 5.2.1.1 levo, iz serije *TMF882x* kreirane od strane kompanije *ams*. U odnosu na ostale senzore

iz ove serije, *TMF8828* ima više zona izlaznih podataka i podesiv opseg. Za ovu seriju komponenti je kreirano zaštitno staklo debljine 0,7mm, prikazano na slici 5.2.1.1. desno, koje se montira iznad komponente i smanjuje uticaj interferencije, čime je poboljšana preciznost senzora. Senzor meri udaljenost od svoje površine do prepreke. Senzor je tipa *dToF*, to jest, *Direct Time of Flight*, i merenje udaljenosti se vrši merenjem vremena preleta. Krajnji rezultat je preciznosti $\pm 10\%$ i mogu se meriti udaljenosti do 5 metara. Bazira se na jednofotonskoj lavinskoj diodi (*SPAD, Single-Photon Avalanche Diode*) koja se ponaša kao fotodetektor, konvertoru vremena u digitalni signal (*TDC, Time-to- Digital Converter*) i kreiranju histograma. Ima široko vidno polje vrednosti 63° . Komponenta je malih dimenzija, i pomoću nje se može napraviti kompaktan uređaj.



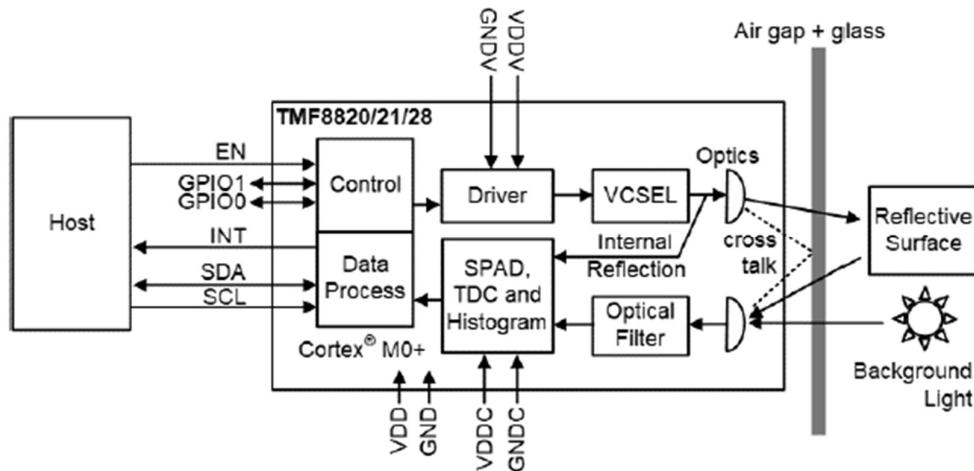
Slika 5.2.1.2. Raspored pinova na komponenti *TMF8828*

Izvor: Datasheet komponente, www.ams.com

Na slici 5.2.1.2. je prikazan raspored pinova komponente. *VDDC* je izvor napajanja za detektor (koji čini niz od 18 SPAD-a), *VDDV* je izvor napajanja za VCESL, dok je *VDD* izvor napajanja čipa. Sva napajanja se međusobno povezuju i priključuju na napon vrednosti +3,3V. Analogno naponima, postoje nožice za uzemljenja (*GNDC, GNDV, GND*) koje je potrebno međusobno povezati i priključiti na masu. *GPIO0* i *GPIO1* nožice su ulazno / izlazne nožice opšte namene. Pin za prekide je označen sa *INT*. Komponenta koristi I²C protokol za komunikaciju, te u tu svrhu sadrži pinove *SCL* i *SDA*. Na kraju, pin *EN* (*enable*) je ulazni pin koji daje dozvolu za rad, aktiviran je na visokom naponskom nivou.

Jednofotonska lavinska dioda (SPAD) predstavlja fotodetektor koji može detektovati čak i svetlosti malog intenziteta. Kao i kod običnih lavinskih dioda, na SPAD je potrebno dovesti inverzan napon. SPAD se polariše naponom znatno iznad napona probaja. Ovaj napon stvara jako električno polje zbog kojeg čak i jedan nosilac nanelektrisanja injekovan u diodu može da pokrene lavinu.

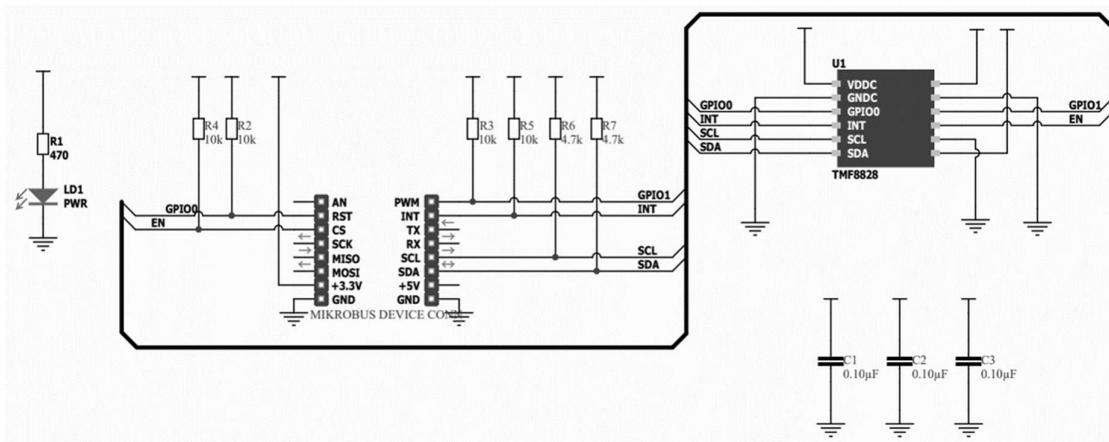
Unutrašnja struktura komponente prikazana je na slići 5.2.1.3. Drajver za VCSEL se kontroliše signalom dozvole (*EN*) napaja pomoću napona dovedenog na *VDDV*, i generiše niz pulseva svetlosti lasera. Pulsevi svetlosti se usmeravaju pomoću mikrosočiva (*MLA, Microlens Array*), čime se osvetljava polje iluminacije (*FoI, Field od Illumination*). Objekat koji se nalazi ispred senzora reflektuje zrake na prijemno optičko sočivo komponente, kao i na detektor. Jednofotonske lavinske diode u detektoru generišu pikove lavinskih struja u zavisnosti od reflektovane svetlosti. Konvertor vremena u digitalni signal meri koliko vremena je prošlo od emisije pulseva do njihovog povratka i kreira histogram. Dobijeni podaci se kontroleru prosleđuju pomoću I²C komunikacije.



Slika 5.2.1.3. Unutrašnja struktura komponente *TMF8828*

Izvor: *Datasheet* komponente, www.ams.com

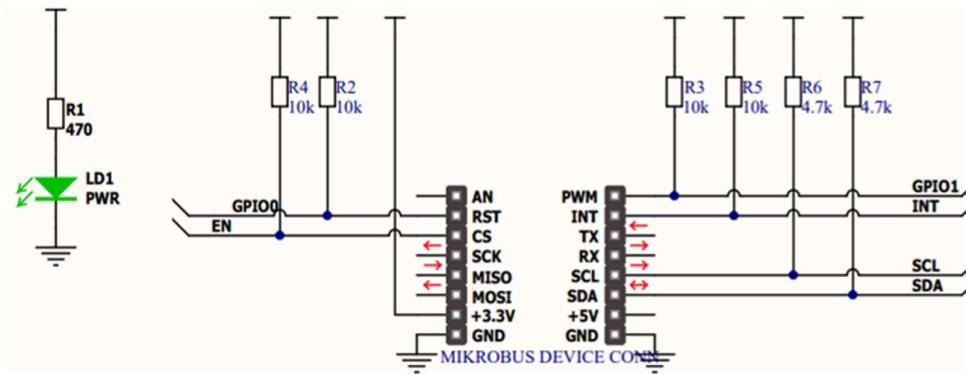
5.2.2 Dizajn šematička pločice *LightRanger 9 Click*



Slika 5.2.2.1. Kompletan dizajn šematička za *LightRanger 9 Click*

Kompletan dizajn šematička za *LightRanger 9 Click* je prikazan na slići 5.2.2.1. Dizajn možemo podeliti u dve veće celine. Prvu celinu čini komponenta *TMF8828* i dikapl kondenzatori. Drugu celinu čini konektor za *mikroBUS* zajedno sa otpornicima i signalnom diodom.

Na slici 5.2.2.2. se bolje vidi povezanost prve celine. Kao što je već spomenuto u četvrtom poglavlju, vodovi I²C komunikacije, to jest, SCL i SDA zahtevaju *pull-up* otpornike (R₆ i R₇), za čiju vrednost je uzeto 4,7kΩ. Prekid (*Int*) je *open-drain* tipa isto kao i vodovi za I²C, te je i kod njega dodat *pull-up* otpornik R₅. Kod ostalih vodova (EN, GPIO0, GPIO1) su takođe dodati *pull-up* otpornici (R₄, R₂, R₃) radi održavanja visokog logičkog nivoa. Obzirom da je *enable* pin aktiviran na visokom nivou, ovaj otpornik održava uređaj uključenim. Signalna dioda LD1 služi za svetlosnu indikaciju priključenosti napona u kolu.

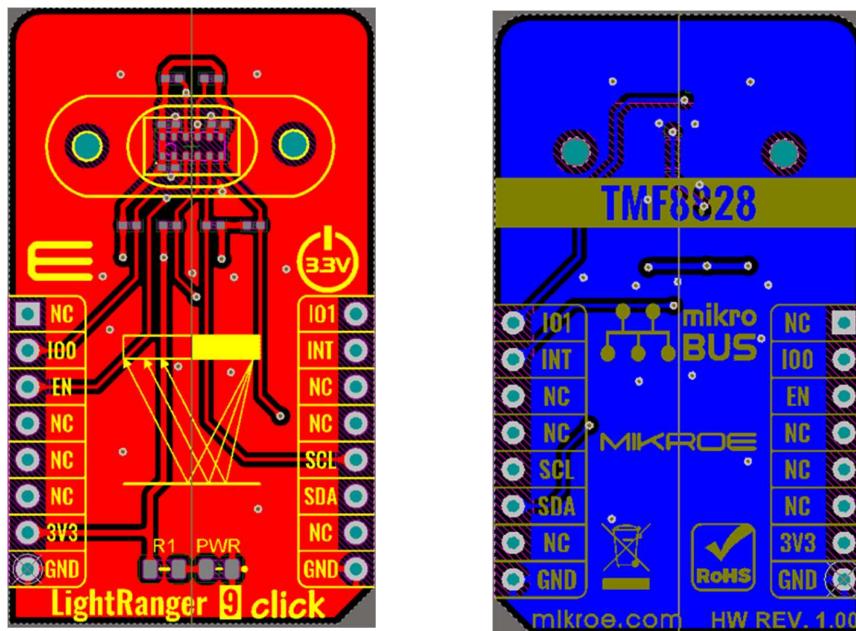


Slika 5.2.2.2. Povezanost mikroBUS priključka

Kod druge celine, to jest, glavne komponente nema dodatnih komponenti osim dikapl kondenzatora. Iz svih signalnih pinova su izvučeni vodovi, svi naponi su spojeni na isti potencijal, i svi priključci za masu su uzemljeni. Između svakog *VDD* - *GND* para je dodat dikapl kondenzator, identično primeru *CapTouch 6 Click* pločice razmatranom u odeljku 5.1.2.

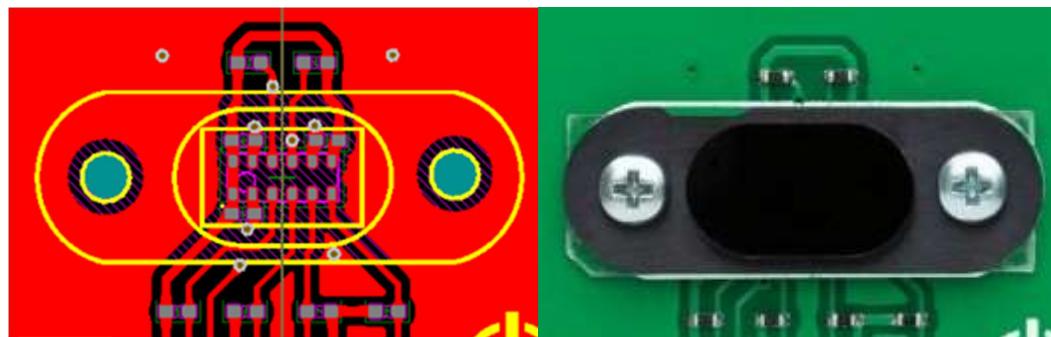
5.2.3 PCB dizajn pločice *LightRanger 9 Click*

Po kreiranju šematika pristupa se kreiranju PCB fajla *LightRanger 9 Click* pločice.



Slika 5.2.3.1. Izgled završenog PCB dizajna gornje strane pločice (slika levo) i dizajna donje strane pločice (slika desno) *LightRanger 9 Click*

Na slikama 5.2.3.1. prikazani su konačni izgledi gornje i donje strane PCB dizajna pločice. Ovde su sve komponente postavljene sa prednje strane pločice. Iako je sama komponenta TMF8828 bila mala, zaštita za nju zauzima mnogo prostora, te je korišćena veća veličina pločice pri izradi. Senzor je, zajedno sa zaštitom, stavljen iznad konektora jer zbog širine zaštita nije mogla da stane u prostoru između leve i desne strane konektora.



Slike 5.2.3.2. Uporedni prikaz komponenti na pločici na PCB dizajnu (slika levo) i na gotovom uređaju (slika desno)

Na slikama 5.2.3.2 je prikazan uvećan deo slike PCB dizajna uporedo sa fotografijom odgovarajućeg dela gotove pločice sa prednje strane, na kojima se vide sve komponente u kolu. Svi otpornici su postavljeni blizu senzora. Otpornici su poređani simetrično radi estetike pločice. Na levoj slici 5.2.3.2. se može primetiti da su dikapl kondenzatori postavljeni, zajedno sa senzorom, ispod zaštite. Zaštita ima šupljinu pravougaonog preseka ispod stakla, predviđenu za smeštanje senzora ispod nje. Žuti pravougaonik na levoj slici 5.2.3.2. označava ovu šupljinu. Iako se po računu i dizajnu pokazalo da u šupljini ima mesta za kondenzatore, ovo nažalost ipak nije bilo izvodljivo u praksi.

Prebacivanje kondenzatora na donju stranu pločice je jedina razlika između priloženog PCB dizajna i krajnjeg dizajna pločice. U originalnom dizajnu se kondenzatori nalaze na gornjoj strani pločice, direktno su povezani sa napajanjima na istoj strani pločice, dok su im drugi krajevi povezani sa masom na donjoj strani pločice (povezani na poligon koji pripada GND mreži). U konačnoj verziji je situacija obrnuta. Kondenzatori se nalaze na donjoj strani, direktno su povezani sa poligonom GND mreže na donjoj strani, dok se sa napajanjima povezuju prebacivanjem vodova na gornji sloj. Upoređivanjem slika 5.2.1. datih na početku odeljka 5.2, koje predstavljaju gotov uređaj, i slika 5.2.3.1. koje predstavljaju PCB dizajn, vidi se da je rutiranje ostalo identično. Takođe, na desnoj slici 5.2.1. se mogu primetiti kondenzatori koji su prebačeni na donju stranu.

Po podacima iz *datasheet-a*, struje u kolu ne prelaze vrednost od 100mA pa nisu potrebne velike širine vodova. Kao i kod primera za *CapTouch 6 Click*, poželjno je koristiti nešto šire vodove. I u ovom primeru je za širinu vodova za podatke korišćeno 30 mila s tim da ista može da se smanjuje do 10 mila u slučaju da vod fizički ne može da se provuče. Po podacima iz tabele 5.1.3.1, ova širina je i više nego dovoljna. Za vodove napajanja je korišćeno 50 mila. Vodovi napajanja širine 50 mila nisu mogli fizički da se povezuju na nožice samog senzora, obzirom da su dimenzije nožica i razmaka između istih izuzetno male, te je podešeno da se širina vodova napajanja može smanjiti do 30 mila. Za povezivanje sa masom, izliveni su poligoni na obe strane pločice. Na nekoliko mesta su dodate vijke koje povezuju gornji i donji poligon. Svrha ovih vija je

ujednačavanje temperature površine i središta pločice. Gornji i donji sloj pločice imaju veću provodnost nego središnji. Samim tim većina toplice pločice je raspoređena upravo po njenoj površini. Dodavanjem vija se deo toplice disipira sa površine u središnji deo pločice čime dobijamo manje temperaturne razlike i smanjujemo ukupnu temperaturu pločice. Međusobnim povezivanjem dva poligona se smanjuje njihova impedansa.

5.2.4 Primena LightRanger 9 Click pločice

Inovacija koju *LightRanger 9 Click* ima u odnosu na prethodne *LightRanger* pločice je zaštitno staklo. Staklo je debljine 0,7mm. Između stakla i senzora se nalazi vazdušni džep od 0,38mm. Implementacijom stakla dolazi do smanjenja interferencije svetlosti čime se obezbeđuje veća preciznost senzora. Ovom pločicom je moguće meriti udaljenosti u intervalu od 10mm do 5m, te se stoga ona može primenjivati kako za detekciju prisustva tako i za merenje udaljenosti.

Jedna od mogućih primena je u robotici. Uz sve veću potrebu za automatizacijom procesa proizvodnje ubrzan je razvoj robotske industrije. Kreiranjem robotskih ruku koje pomeraju objekte, pridržavaju ih ili pričvršćuju olakšava se težak fizički rad koji bi inače morali da obavljaju ljudi. Istovremeno, posao se obavlja brže, kvalitetnije i bezbednije. Osnovni preduslov pravilnog funkcionisanja robotske ruke je da ista zna tačnu poziciju objekta. Primenom ove pločice, robot može tačno znati na kojoj udaljenosti se nalazi objekat kome treba pristupiti.

Ova se pločica može koristiti u robotskoj industriji i za sprečavanje sudara. Naime, ukoliko je robot mobilan, uvek postoji mogućnost, ma koliko ona bila mala, da u nekom trenutku dođe do kolizije sa nekim drugim objektom ili robotom, što može dovesti do oštećenja na objektima kolizije. Ugradnjom pločice na robota, robotu se pružaju informacije o udaljenosti objekta prema kome se kreće. Robotu se može dati dozvola kretanja unapred sve dok se nalazi na bezbednoj distanci od objekta. Ukoliko se pločicom detektuje mala razdaljina od robota do prepreke, robot bi zaustavio svoje kretanje u tom smeru ili promenio smer kretanja, čime bi se izbegao sudar. Isti princip može da se primeni kod malih automatskih usisivača. Moderni mali usisivači ne zahtevaju asistenciju od ljudi, nego se sami kreću po sobi i usisavaju prašinu sa poda. Obzirom na to da sobe imaju zidove i nameštaj, neophodan je senzor koji će zaustaviti usisivač ispred prepreke i usmeriti ga na drugu stranu.

Osim merenja udaljenosti ova se pločica može primenjivati i za detekciju prisustva, na primer u izradi alarmnih sistema. Većina muzeja neke od svojih dragocenosti ogradi pomoću trake ili kanapa, koji ne sprečavaju ljude da se provuku ispod i priđu, ili pokušaju rukom da dohvate i dotaknu predmet. Ugradnjom senzora se može detektovati da li je neko pokušao da prođe ográđeni prostor. Povezivanjem alarma ili uređaja koji javlja nadležnim se mogu sprečiti krađe ili oštećenja koja mogu nastati prilaskom zlonamernog lica.

Senzor se može koristiti i u svrhe pomoći oko parkiranja. Velikom delu vozača parkiranje predstavlja problem zbog loše procene udaljenosti objekata od vozila. Sve veći broj vozila danas ima ugrađene pomoćne senzore za parkiranje. Senzori se obično postavljaju na prednjoj i / ili zadnjoj strani vozila. Približavanjem prepreci, na primer, sa zadnje strane vozila, dolazi do zvučne indikacije slabije učestanosti. Sve većim približavanjem prepreci zvuk postaje učestaniji i time javlja vozaču da se približava prepreci na udaljenosti koja nije bezbedna.

Ovaj senzor može takođe automatizovati određene procese. Na primer, moguće je napraviti automatsku hranilicu za kućne ljubimce. Hranilica treba da sadrži senzor, mali kontroler i jedan mali motor. Motor otvara vratanca između posude za hrani i dispenzera. Ukoliko je ljubimac lepo dresiran, moguće je naučiti ga da priđe hranilici kada je gladan, ili postavi šapu na određeno mesto ispred senzora. Senzor će detektovati ljubimca, i pokrenuće motor. Motor će otvoriti vratanca, sačekaće kratak vremenski period dovoljno da se u činiju sipa porcija hrane, nakon čega će zatvoriti vratanca.

Ovom pločicom je moguće automatizovati i paljenje svetla u prostoriji. Institucije i javna mesta sve više vode računa o uštedi elektronske energije. U određenim prostorijama gde nije konstantno neophodno upaljeno svetlo nego samo ukoliko se nalazi neko u toj prostoriji proces paljenja i gašenja svetla može se automatizovati senzorom. Česta primena ovog principa se može primetiti u toaletima. Ukoliko se detektuje ulazak osobe u prostoriju, svetlo se pali. Svetlo ostaje upaljeno ili određeni vremenski period ili sve dok se detektuje pokret u prostoriji. Nakon izlaska osobe iz prostorije, svetlo će ostati upaljeno još kratak vremenski period, nakon čega će se ugasiti. Ukoliko se pre gašenja svetla detektuje ponovno kretanje, brojač za gašenje svetla se restartuje.

6. Razvoj softvera za *Click* pločice

Da bi proizvod koji je kreiran zaživeo, potrebno ga je testirati. Testiranje se vrši pisanjem programa. U svrhe olakšanja programiranja *Click* pločica razvijen je paket biblioteka pod nazivom *mikroSDK* od strane kompanije *Mikroelektronika*.

6.1 *mikroSDK*

Na početku je potrebno razumeti šta su tačno biblioteke koje sadrži *mikroSDK* i kako se koriste. *SDK* je skraćenica za *Software Development Kit*, ili u prevodu komplet za razvijanje softvera. *MikroSDK* predstavlja kolekciju softverskih biblioteka sa alatima za razvoj softvera i programskog interfejsa aplikacije (*API, Application Programming Interface*). Biblioteke su otvorenog koda (*open - source*), što znači da svako može da im pristupi i da ih koristi. Upotreba ovih biblioteka pojednostavlja rad na razvijanju programa. Osim toga, omogućava da jedan program bez izmena (ili sa minimalnim izmenama) ima istu funkcionalnost čak iako se promeni kontroler ili razvojni sistem uz pomoću kojeg se programira određena pločica. Razvijeni program postaje nezavisан od korišćenih hardvera. Jedino mapiranje koje je potrebno je vezano za naglašavanje koji *mikroBUS* priključak na razvojnom sistemu je korišćen.

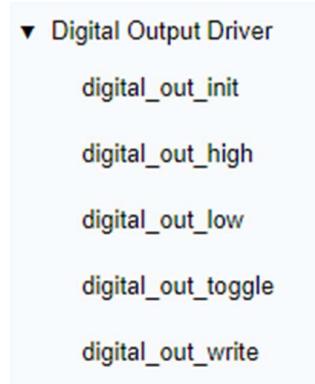
Dve važne mikrokontrolerske periferije koje sadrži *mikroSDK* su periferijske biblioteke (HAL, *Hardware Abstraction Layer*) i periferijski drajveri (DRV, *Driver*). HAL omogućava kreiranje uopštene periferijske funkcionalnosti tako što apstrahuje pristup periferijskim registrima hardvera. Fajlovi iz periferijskih biblioteka su korišćeni za izradu periferijskih drajvera. DRV je viši sloj softvera, služi za dodavanje višeg nivoa funkcionalnosti, implementaciju uobičajenih protokola, njihovu inicijalizaciju... Pod DRV smatramo softverske biblioteke koje inicijalizuju hardver i omogućavaju pristup hardveru. Biblioteke DRV obuhvataju širok spektar funkcija za rad sa pločicama i nema potrebe za korišćenjem funkcija iz nižeg sloja (HAL) pri razvoju softvera.

Svi fajlovi u sklopu *mikroSDK* su razvijeni upravo za *Click* pločice, te sadrže sve potrebne osnovne funkcije za njihovo olakšano programiranje. Najčešća implementacija *mikroSDK* u program se svodi na priključivanje odgovarajućih fajlova iz DRV i pozivanjem funkcija iz istih. DRV se sastoji iz parova *header* i *source*, funkcije su time grupisane u smislene celine. API celine koje sadrži su sledeće:

- *Analog Input Driver* – rad sa ulaznim podacima analognog tipa
- *Digital Input Driver* – rad sa ulaznim podacima digitalnog tipa
- *Digital Output Driver* – rad sa izlaznim podacima digitalnog tipa
- *I²C Master Driver* – rad sa komunikacijom pomoću I²C protokola
- *Name Driver* – definisanje imena za pinove i portove
- *Port Driver* – rad sa *GPIO (General Purpose Input Output)* portovima
- *PWM Driver* – rad sa PWM signalima
- *SPI Master Driver* – rad sa komunikacijom pomoću SPI protokola
- *UART Driver* – rad sa UART komunikacijom

Za razvoj softvera za *Click* pločice neophodno je prvenstveno razumeti funkcije u navedenim bibliotekama, kao i način njihove implementacije. Ovo je najlakše naučiti pisanjem običnih kodova u kojima se samo pozivaju funkcije iz datih biblioteka. Kretanjem od jednostavnijih primera ka komplikovanijima se stiče bolje razumevanje datih funkcija, kao i način i svrha njihove upotrebe. Opisi svih biblioteka i svih funkcija koje sadrže dat je detaljno na zvaničnom *github* sajtu firme *Mikroelektronika*.

Svaka biblioteka sadrži najčešće korišćene funkcije za određenu celinu. Sadržaj (funkcije) biblioteke *Digital Output Driver* prikazan je na slici 6.1.1. Kreirane funkcije služe za inicijalizaciju izlaznih pinova, njihovo podizanje na logičku jedinicu ili spuštanje na logičku nulu, inverziju vrednosti, ili upisivanje željene vrednosti.



Slika 6.1.1. Sadržaj biblioteke *Digital Output Driver*

Na slici 6.1.2. je prikazana jedna od funkcija iz *Digital Output Driver* biblioteke - *digital_out_high*. Ova funkcija se često nalazi u praksi i koristi se ukoliko je potrebno da se na izlaznom pinu podigne vrednost na logičku jedinicu. Funkciji se prosleđuje željeni pin, to jest, parametar tipa *digital_out_t*. Struktura *digital_out_t* u sebi definiše samo jedan parametar (*pin*) tipa *hal_gpio_pin_t* (slika 6.1.3). Ovde se DRV fajl nadovezuje na HAL fajl, obzirom da je

hal_gpio_pin_t struktura definisana upravo u nižem sloju i čine je *base* (adresa podnožja pina) i *mask* (bit maska pina).

Vraćanjem na funkciju sa slike 6.1.2. se može primetiti da se proverava da li vrednost adrese (*pin.base*) prosleđenog pina *out* nije jednaka sa nulom (*NULL*). Ukoliko je jednaka nuli, onda pin nije pronađen i neće se ništa izvršiti. U suprotnom će se pozvati funkcija iz HAL biblioteke za setovanje vrednosti pina. Ovde se ponovo može primetiti povezanost DRV i HAL slojeva i koliku potporu HAL sloj pruža DRV sloju.

```
void digital_out_high( digital_out_t *out )
{
    if ( NULL != out->pin.base )
    {
        ...
        hal_gpio_set_pin_output( &out->pin );
    }
}
```

Slika 6.1.2. Telo funkcije *digital_out_high*

```
typedef struct
{
    hal_gpio_pin_t pin;
} digital_out_t;
```

Slika 6.1.3. Struktura *digital_out_t*

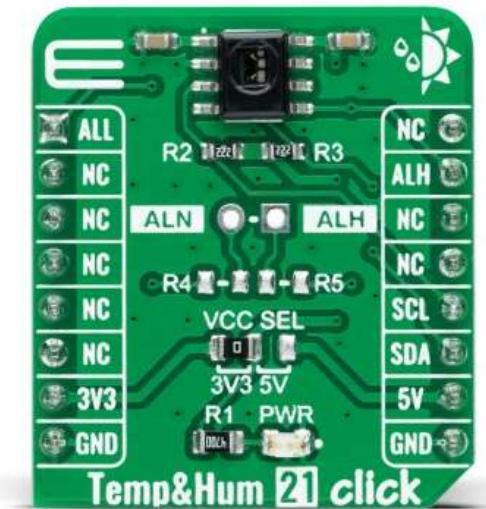
Na identičan način pozivanja funkcija iz HAL biblioteka za potrebe DRV biblioteka, pri izradi softvera za pločice je moguće pozivati i koristiti gotove funkcije iz DRV. Ovim je omogućeno da se korisnik fokusira samo na kreiranje funkcija karakterističnih za željenu pločicu bez trošenja vremena na kreiranje sporednih funkcija. Primer razvijanja programa za *Click* pločicu uz korišćenje *mikroSDK* biblioteka će biti opisan u odeljku 6.2.

Radi korišćenja *mikroSDK* celim potencijalom, najbolje je instalirati softverski alat *NECTO Studio 2.0*. Prilikom instalacije *NECTO* alata se korisniku pruža mogućnost izbora verzije paketa *mikroSDK* koju želi da instalira. Proces instalacije kratko traje, zauzima oko 3GB memorije na kompjuteru i pruža kvalitetno okruženje za razvoj programa za pločice. *NECTO* omogućava kreiranje *Setup-a* za trenutnu postavku hardvera koja se programira, te se biraju kompjajler, SDK, razvojni sistem, MCU, displej i programator. Prednost korišćenja *NECTO Studio* programa u kombinaciji sa *mikroSDK* je mogućnost prelaska sa jedne postavke hardvera na drugu samo pomoću izmene definisanog *Setup-a*. Na primer, ukoliko se promeni kontroler na razvojnem sistemu, potrebno je u trenutnom *Setup-u* odabrati drugi kontroler ili kreirati novi *Setup* za trenutnu postavku. Kreirani program će imati istu funkcionalnost nakon ovih promena.

6.2 Program za Temp&Hum 21 Click pločicu

U ovom odeljku će biti detaljno opisan jedan primer razvoja softvera za pločicu, konkretno *Temp&Hum 21 Click* prikazanoj na slici 6.2.1. Pločica služi za merenje temperature i vlažnosti. Kao što se može videti sa slike 6.2.1, pločica koristi I²C komunikacioni protokol (pinovi *SCL* i

SDA), te je potrebno iz *mikroSDK* uključiti biblioteku za I²C. Glavna komponenta pločice je senzor *HIH8130-021-001*. Imamo i dva dodatna pina za alarme: *ALL (Alarm Low)* na mestu gde je inače pin *AN* pin i *ALH (Alarm High)* na mestu gde je inače *PWM* pin. Ovi pinovi su izlazni pinovi pločice, ali su našem kontroleru to ulazni pinovi, te je potrebno iz *mikroSDK* takođe uključiti i biblioteku za *Digital Output*.



Slika 6.2.1. Temp&Hum 21 Click

Sve potrebne funkcije za rad sa pločicom se pišu u bibliotekama, nakon čega se pozivaju u glavnom kodu. Potrebno je kreirati .c (*source*) i .h (*header*) fajlove, kao i glavni program main.c u kojem se pozivaju biblioteke i demonstrira rad funkcija. *Header* fajl sadrži deklaracije funkcija, njihove detaljne opise, spisak svih potrebnih registara, adresa, brojnih vrednosti konstanti... U *source* fajlu definišemo funkcije.

6.2.1 Kreiranje *header* fajla

Kreirani *header* fajl mora imati naziv identičan pločici za koju se razvija softver - *tempum21.h*. Na početku fajla je potrebno dodati biblioteke iz *mikroSDK* koje sadrže funkcije neophodne za razvoj koda.

```
#include "drv_digital_in.h"
#include "drv_i2c_master.h"
```

Nakon toga se definisu svi registri, konstante, adrese, statusi... Definisanih podataka ima mnogo i neće svi biti nabrojani. U isečku koda ispod je dato par primera definisanja.

#define TEMPHUM21_RESOLUTION	16382.0
#define TEMPHUM21_TEMP_MULTIPLIER	165
#define TEMPHUM21_TEMP_OFFSET	40
#define TEMPHUM21_STATUS_NORMAL_OPERATION	0x00
#define TEMPHUM21_STATUS_STALE_DATA	0x01
#define TEMPHUM21_STATUS_COMMAND_MODE	0x02
#define TEMPHUM21_STATUS_NOT_USED	0x03

Osim ovih definicija, potrebno je definisati i mapiranje. Mapiranju se prosleđuje konfiguracioni fajl zajedno sa informacijom koji *mikroBUS* je korišćen, nakon čega ih on povezuje. Potrebno je povezati nazine pinova na pločici (iz konfiguracionog fajla) sa nazivima pinova koji su na *mikroBUS* konektoru. Iz tog razloga se *SCL* mapira na *MIKROBUS_SCL*, dok se *ALL* mapira na *MIKROBUS_AN*.

```
#define TEMPHUM21_MAP_MIKROBUS( cfg, mikrobus ) \
    cfg.scl = MIKROBUS( mikrobus, MIKROBUS_SCL ); \
    cfg.sda = MIKROBUS( mikrobus, MIKROBUS_SDA ); \
    cfg.all = MIKROBUS( mikrobus, MIKROBUS_AN ); \
    cfg.alh = MIKROBUS( mikrobus, MIKROBUS_PWM );
```

Kada se završi sa svim definicijama, potrebno je kreirati dve strukture. Prva struktura je *temphum21_t* i ona će predstavljati sam objekat, to jest pločicu. Struktura treba sadržati parametre od važnosti koje dobijamo od objekta (ulazni pinovi, promenljive poput temperature i vlažnosti), ali i parametre čije nam je menjanje od značaja (moduli i adrese).

```
typedef struct
{
    // Input pins
    digital_in_t all;           /**< Alarm Low pin. */
    digital_in_t alh;          /**< Alarm High pin. */
    // Modules
    i2c_master_t i2c;          /**< I2C driver object. */
    // I2C slave address
    uint8_t slave_address;     /**< Device slave address (used for I2C driver). */
    // Variables
    uint16_t temperature;
    uint16_t humidity;
} temphum21_t;
```

Druga struktura je *temphum21_cfg_t* i predstavlja konfiguracioni objekat. Ovaj objekat se prosleđuje pri mapiranju i od izuzetne važnosti je da se u strukturi navedu svi pinovi, kako za komunikaciju, tako i dodatni. Pinovi napajanja se ne spominju nigde u kodu. I u ovoj strukturi se takođe, pored pinova, navode i promenljive (podaci komunikacije poput brzine i adrese, temperatura, vlažnost).

```
typedef struct
{
    // Communication gpio pins
    pin_name_t scl;           /**< Clock pin descriptor for I2C driver. */
    pin_name_t sda;           /**< Bidirectional data pin descriptor for I2C driver. */
    // Additional gpio pins
    pin_name_t all;           /**< Alarm Output Low */
    pin_name_t alh;           /**< Alarm Output High */
    // Variables
    uint32_t i2c_speed;       /**< I2C serial speed. */
    uint8_t i2c_address;      /**< I2C slave address. */
    uint16_t dev_temperature;
    uint16_t dev_humidity;
} temphum21_cfg_t;
```

Preostaju još samo deklaracije funkcija. Deklarisane su sledeće funkcije:

- **void temphum21_cfg_setup (temphum21_cfg_t *cfg);** – funkcija za podešavanje konfiguracionog fajla, kao parametar joj se prosleđuje objekat *cfg* strukture

temphum21_cfg_t tipa. Funkcija inicijalizuje konfiguracionu strukturu *Click-a* na inicijalne vrednosti.

- **err_t temphum21_init (temphum21_t *ctx, temphum21_cfg_t *cfg);** – Funkcija inicijalizuje sve potrebne pinove i periferije korišćene za ovu pločicu. Prosleđuju joj se objekti obe prethodno spomenute klase, a kao povratnu vrednost vraća broj 0 ukoliko je inicijalizacija uspešna ili broj -1 ukoliko je došlo do greške.
- **err_t temphum21_start_measurement (temphum21_t *ctx);** – Funkcija koja šalje poruku sa *master-a* i započinje proces merenja. Prosleđujemo joj samo objekat *ctx* klase temphum21_t. Takođe vraća 0 ili -1 u zavisnosti od toga da li je došlo do greške pri izvršavanju ili ne.
- **err_t temphum21_read_measurement (temphum21_t *ctx);** – Funkcija za očitavanje izmerenih veličina. Očitava poruku primljenu na *slave-u*, i upisuje vrednosti temperature i vlažnosti vazduha iz 14-bitnog ADC-a (*Analog Digital Converter*) u objekat *ctx*. Osim povratne vrednosti -1 ukoliko je došlo do greške, ova funkcija kao povratnu vrednost vraća u kom stanju se nalazi uređaj (*Normal Operation, State Data, Command Mode, Not Used*). Stanje zavisi od očitavanja iz registra, i sva stanja su definisana u podacima od proizvođača za korišćeni senzor.
- **float temphum21_get_temperature (temphum21_t *ctx, uint8_t temp_unit);** – Funkcija koja očitava vrednost temperature sa konvertora upisanu u objekat *ctx* i konvertuje je u vrednost temperature u realnom svetu. Temperatura se računa po formuli (4) iz *datasheet-a* glavne komponente.

$$T[{}^{\circ}\text{C}] = \frac{\text{Temperatura očitana na konvertoru}}{2^{14}-2} \times 165 - 40 \quad (4)$$

U formuli rezolucija iznosi $2^{14}-2$, množitelj (*multiplier*) 165, i odstupanje (*offset*) 40. Osim konverzije, ova funkcija može pretvoriti konvertovanu vrednost temperature iz stepena celzijusa u drugu mernu jedinicu, u zavisnosti od prosleđenog parametra *temp_unit*. Za povratnu vrednost se dobija vrednost temperature u željenoj mernoj jedinici.

- **float temphum21_get_humidity (temphum21_t *ctx);** – Funkcija koja očitava vrednost vlažnosti vazduha sa konvertora upisanu u objekat *ctx* i konvertuje je u vrednost vlažnosti vazduha u realnom svetu, koju prosleđuje kao povratnu vrednost. Formula (5) za računanje vrednosti vlažnosti vazduha je sledeća:

$$\text{Vlažnost}[\%RH] = \frac{\text{Vlažnost očitana na konvertoru}}{2^{14}-2} \times 100 \quad (5)$$

- **uint8_t temphum21_get_alarm_high_pin (temphum21_t *ctx);** – Funkcija koja vraća logičko stanje pina *Alarm High*.
- **uint8_t temphum21_get_alarm_low_pin (temphum21_t *ctx);** – Funkcija koja vraća logičko stanje pina *Alarm Low*.

6.2.2 Kreiranje *source* fajla

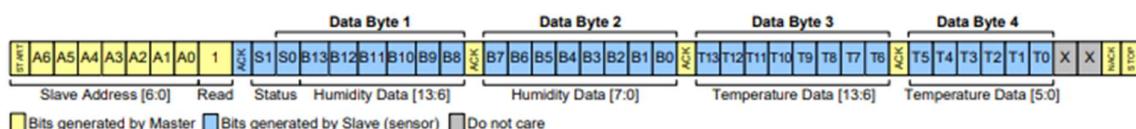
Potrebno je sve funkcije deklarisane u *header* fajlu definisati u *source* fajlu. *Source* fajl takođe naziv dobija po pločici za koju se razvija softver - *temphum21.c*. Funkcije *temphum21_cfg_setup* i *temphum21_init* su šablonske, i identične su za sve pločice uz minimalne izmene (nazivi pinova). Ove dve funkcije nisu ručno kucane, nego su automatski izgenerisane pri kreiranju fajlova od strane jednog od softverskih alata u firmi *Mikroelektronika* koji olakšava rad sa kucanjem programa, te neće biti detaljnije opisane.

Definicija funkcije *temphum21_start_measurement* je prikazana ispod. Telo funkcije pokušava da upišemo običan bit (*TEMPHUM21_DUMMY*, koji je definisane vrednosti 0x00) na I²C magistralu. Ukoliko je upisivanje podatka uspelo, komunikacija je omogućena, povratna vrednost će biti *TEMPHUM21_OK* (definisano kao 0) i moguće je otpočeti merenja. Sa druge strane, ukoliko je došlo do greške pri uspostavljanju komunikacije, povratna vrednost *TEMPHUM21_ERROR* (definisana kao -1) će nam javiti da je došlo do greške. Može se primetiti da je u telu ove funkcije korišćena funkcija pod nazivom *i2c_master_write*. Ta funkcija je pozvana iz *mikroSDK* biblioteka, konkretno iz *drv_i2c_master.h*, potrebno joj je proslediti objekat klase *i2c_master_t* (klasa je definisana u samoj I²C biblioteci), nakon čega sledi podatak koji se šalje, i dužina podatka (broj bajtova koji se šalje). Detaljniji opis *i2c_master_write* funkcije se može naći u samoj *mikroSDK* biblioteci.

```
err_t temphum21_start_measurement ( temphum21_t *ctx )
{
    if ( i2c_master_write( &ctx->i2c, TEMPHUM21_DUMMY, 1 ) == I2C_MASTER_ERROR )
    {
        return TEMPHUM21_ERROR;
    }

    Delay_100ms( );
    return TEMPHUM21_OK;
}
```

Da bi se ponoću funkcije *temphum21_read_measurement* očitala vrednosti koje meri senzor, potrebno je prvo proučiti kako se generišu bitovi. Sa slike 6.2.2.1. se može videti da je potrebno prvo generisati poruku sa masterom. Nakon toga pristignu 4 bajta, od čega su prva dva za očitavanje vlažnosti, dok su druga dva za očitavanje temperature. Kod prvog bajta na slici vidimo dva statusna bita, oni nisu od važnosti za očitavanje merenja, i biće otklonjeni maskom. Kod poslednjeg bajta postoje dva siva bita (po legendi datoј uz sliku, sivo označava nebitne bite) koji se takođe otklanjaju maskom.



Slika 6.2.2.1. Očitavanje podataka sa senzora

Izvor: sps.honeywell.com – I2C Comms with Honeywell HumidIcon™ Digital Humidity/Temperature Sensors

Sada kada je objašnjen princip očitavanja podataka, može se preći na implementaciju istog u telu funkcije. Rečeno je da je potrebno generisati poruku sa mastera da bi se inicirala komunikacija, što se obavlja pozivanjem prethodno kreirane funkcije *tempum21_start_measurement*. Sada je potrebno pročitati podatke koje je slave generisao na magistrali. Ovo se vrši pozivanjem funkcije *i2c_master_read* iz *drv_i2c_master.h* koja će očitane podatke zabeležiti u *read_reg* promenljivu. Promenljiva *read_reg* je *uint8_t* tipa, i napravljena je da može skladištiti 4 podatka, te se u njoj mogu skladištiti sva četiri bajta koja se prime putem komunikacije. Za vrednost vlažnosti (*humidity*) se očitava nulti bajt, pomera osam mesta uлево i zatim se dodatno upisuje vrednost prvog bajta pomoću *bitwise or* operacije, te sad imamo očitana oba bajta u jednoj promenljivoj. Nakon toga se maskom (vrednosti 0x3FFF definisane u *header* fajlu) otklanjaju statusni biti. Sličan postupak se ponavlja za očitavanje temperature. Obzirom da su poslednja dva bita kod temperature nebitna, svi očitani biti se pomeraju za dva mesta udesno. Ovako se dobiju dva nebitna bita na početku umesto na kraju koji se otklanjaju primenom maske. Status uređaja se očitava i smešta u promenljivu *status*. Proverom vrednosti ove promenljive u funkciji se može utvrditi u kom stanju se nalazi uređaj i poslati to stanje kao povratnu vrednost.

```

if ( i2c_master_read( &ctx->i2c, read_reg, 4 ) == I2C_MASTER_ERROR )
{
    return TEMPHUM21_ERROR;
}

ctx->humidity = read_reg[ 0 ];
ctx->humidity <<= 8;
ctx->humidity |= read_reg[ 1 ];
ctx->humidity = ctx->humidity & TEMPHUM21_MASK ;

ctx->temperature = read_reg[ 2 ];
ctx->temperature <<= 8;
ctx->temperature |= read_reg[ 3 ];
ctx->temperature = ( ctx->temperature >> 2 ) & TEMPHUM21_MASK ;

status = ( read_reg[ 0 ] & 0xC0 ) >> 6;

```

Da bi se očitala pravu vrednost temperature, mora se primljena poruka je konvertovati po formuli (4) iz odeljka 6.2.1. Kao što je već spomenuto, ovo konvertovanje se vrši u funkciji *tempum21_get_temperature*. U telu funkcije se prvo konvertuje temperatura očitana sa ADC-a po formuli (4). Nakon konvertovanja se dobija temperatura u jedinici °C. Vrši se provera parametra *temp_unit* koji se prosleđuje funkciji, i u zavisnosti od toga se već dobijena temperatura jedinici °C može konvertovati u °F ili °K. Kao povratnu vrednost funkcije se dobija vrednost temperature u željenoj jedinici. Funkcija *tempum21_get_humidity* samo konvertuje očitanu vlažnost po formuli (5) na način identičan kao u ovoj funkciji, bez daljeg prebacivanja u druge jedinice, te nema potrebe za njenim detaljnim objašnjavanjem.

```

float tempHum21_get_temperature ( tempHum21_t *ctx, uint8_t temp_unit )
{
    float temp;

    temp = ( float ) ( ctx->temperature / TEMP_HUM21_RESOLUTION ) * TEMP_HUM21_TEMP_MULTIPLIER - TEMP_HUM21_TEMP_OFFSET;

    if ( temp_unit == TEMP_HUM21_TEMP_DATA_IN_KELVIN )
    {
        return temp + TEMP_HUM21_CONST_KEL;
    }
    else if ( temp_unit == TEMP_HUM21_TEMP_DATA_IN_FAHRENHEIT )
    {
        return ( temp * TEMP_HUM21_CONST_FAHRI ) + TEMP_HUM21_CONST_FAHR2;
    }
    return temp;
}

```

Poslednje dve funkcije su veoma slične. One su *tempHum21_get_alarm_high_pin* i *tempHum21_get_alarm_low_pin*. U telu ovih funkcija se samo očitava trenutno stanje na pinu i prosleđuje se. Očitavanje se vrši pomoću funkcije *digital_in_read* iz biblioteke *drv_digital_in.h*. Funkciji za očitavanje je potrebno proslediti sa kog pina treba da očita vrednost. U funkciji za *alarm_high_pin* je očitana vrednost sa pina *alh*, dok je u funkciji za *alarm_low_pin* očitana vrednost sa pina *all*, što je jedina razlika u telima ovih dveju funkcija.

```

uint8_t tempHum21_get_alarm_high_pin ( tempHum21_t *ctx )
{
    return digital_in_read ( &ctx->alh );
}

```

6.2.3 Glavna funkcija

Kreirane biblioteke sada treba uključiti u glavni program (*main.c*), u kome se pozivaju i testiraju funkcije. Glavni program diktira rad samog uređaja. Program je kreiran tako da podatke štampa preko UART komunikacije. Za ispis podataka na UART se koristi biblioteka *log.h* iz *mikroSDK* paketa. Na početku se uključuju potrebne biblioteke i kreira se objekat klase *tempHum21_t* za rad sa pločicom i objekat klase *log_t* za ispis podataka.

```

#include "log.h"
#include "tempHum21.h"

static tempHum21_t tempHum21;
static log_t logger;

```

Glavni program se sastoji iz dve celine, to jest funkcije:

- **void application_init (void)**: vrši se inicijalizacija svih drajvera i izvršavaju se osnovne konfiguracije. Prvo se kreiraju konfiguracioni fajlovi za *log* i za *tempHum21*.

```

log_cfg_t log_cfg; /*< Logger config object. */
tempHum21_cfg_t tempHum21_cfg; /*< Click config object. */

```

Nakon kreiranja ovih objekata, potrebno je pozvati funkcije za konfiguracije, inicijalizacije, kao i izvršiti mapiranje. Pločica je bila priključena u prvi *mikroBUS* priključak na razvojnem sistemu, te je funkciji za mapiranje prosleđeno *MIKROBUS_1*. Ukoliko je korišćen drugi priključak, potrebno je promeniti broj 1 u redni broj odgovarajućeg priključka.

```

LOG_MAP_USB_UART( log_cfg );
log_init( &logger, &log_cfg );
log_info( &logger, " Application Init " );

// Click initialization.
temphum21_cfg_setup( &temphum21_cfg );
TEMPHUM21_MAP_MIKROBUS( temphum21_cfg, MIKROBUS_1 );

if ( I2C_MASTER_ERROR == temphum21_init( &temphum21, &temphum21_cfg ) )
{
    log_error( &logger, " Communication init." );
    for ( ; ; )
}
log_info( &logger, " Application Task " );

```

- **void application_task (void):** izvršava željeni zadatak. U ovom konkretnom kodu se u aplikativnom tasku očitava temperatura u dve različite merne jedinice, kao i vlažnost vazduha, nakon čega se svi parametri se ispisuju jednom po sekundi preko UART komunikacije. Na početku se kreiraju dve promenljive: *temperature* i *humidity* u kojima će se skladištiti tražene vrednosti merenih veličina.

```

float temperature;
float humidity;

```

Poziva se funkciju *temphum21_read_measurement* da bi se očitale vrednosti generisane od senzora. Te vrednosti se konvertuju pomoću funkcija *temphum21_get_temperature* i *temphum21_get_humidity*. Dodaje se pauzu od jedne sekunde između generisanja dveju poruka komandom *Delay_ms(1000)*.

```

temphum21_read_measurement( &temphum21 );

temperature = temphum21_get_temperature( &temphum21, TEMPHUM21_TEMP_DATA_IN_CELSIUS );
log_printf( &logger, "Temperature in Celsius: %.2f C \r\n", temperature );

temperature = temphum21_get_temperature( &temphum21, TEMPHUM21_TEMP_DATA_IN_KELVIN );
log_printf( &logger, "Temperature in Kelvin: %.2f K \r\n", temperature );

humidity = temphum21_get_humidity( &temphum21 );
log_printf( &logger, "Humidity: %.1f RH \r\n", humidity );

Delay_ms( 1000 );

```

Na kraju se kreira **void main (void)** funkcija, u kojoj se poziva *application_init* jednom, dok se *application_task* izvršava u beskonačnoj petlji radi konstantnog očitavanja merenih veličina.

```

void main ( void )
{
    application_init();
    for ( ; ; )
    {
        application_task();
    }
}

```

7. Zaključak

Važnost razvoja, kako u ostalim granama nauke, tako i u elektrotehnici obrazložena je u uvodu ovog rada.. Živimo u svetu gde se svakim danom pojavljaju nove tehnologije, te ljudska potreba za inovacijama uvek raste. Teži se kreiranju jeftinijih uređaja koji imaju bolje performanse i koje će korisnici moći da koriste sa lakoćom. Firma *Mikroelektronika* je kreiranjem standarda *mikroBUS* na inovativan način napravila veliki korak kako u razvoju hardvera, tako i u razvoju softvera. Samom implementacijom *mikroBUS* priključka korisnik sebi otvara mogućnost za rad sa preko 1.300 različitih uređaja (*Click* pločica) iz velikog broja kategorija.

Jedan priključak omogućava korisniku da radi sa senzorima, ekranima, LED matricama, mikforonima, zvučnicima, satovima, memorijom... Ovim je kreiran univerzalni standard za sve vrste integrisanih kola koji korisniku daje nebrojane mogućnosti. Veliki razvoj *Click* pločica je doveo do toga da je kreirano više pločica iz iste kategorije, te korisnik može da bira uređaj sa specifikacijama koje mu odgovaraju. Ukoliko korisnik, na primer, želi temperaturni senzor, ispred njega se nalazi širok izbor pločica koje mere temperaturu. Izbor zavisi od potreba korisnika koji može birati karakteristike pločice poput:

- vrste komunikacije – SPI, I²C ili UART,
- napajanja – 3.3V ili 5V,
- da li su poželjne i dodatne funkcionalnosti na istoj pločici (poput merenja vlažnosti vazduha paralelno sa temperaturom)
- brzine izvršavanja željene funkcije
- preciznosti merenja
- proizvođača komponente
- veličine pločice – mala, srednja ili velika

Ručnim postupkom kreiranja pločica je izuzetno teško povezati komponente malih dimenzija ili sa nepristupačnim pinovima pomoću lemilice. Rad sa ovakvim komponentama zahteva drugačije tehnike lemljenja, dodatne uređaje i materijale (pištolj za vreo vazduh, lemlna pasta, maska za nanošenje paste...). *Click* pločice se kreiraju pomoću mašina. Ceo postupak

mašinske proizvodnje je detaljno opisan u trećem poglavlju. Proizvodnja je izuzetno precizna, dimenzijske kreirane uređaje su minimizovane zbog korišćenja SMD komponenti. Najvažnija prednost ovakvog postupka proizvodnje je što izrada pločice nije ograničena fizičkim dimenzijama i vrstom kućišta komponenti, te je moguće od svakog integriranog kola kreirati pločicu. Korisnik dobija gotovu programibilnu pločicu koja ne zahteva dodatna povezivanja sa drugim komponentama, kao ni znanja iz elektronike potrebna da bi se kreirala pločica. Samim time se skraćuje posao kreiranja gotovog uređaja jer je potrebno samo napisati program za datu pločicu. Moguće je kombinovati više pločica na jednom razvojnem sistemu čime se pravi veliki uređaj koji može izvršavati više funkcija.

Detaljan postupak dizajniranja pločice je opisan u četvrtom poglavlju. Opisan je detaljan postupak kreiranja šematika i PCB fajla preko osnovnih stvari do finesa koje poboljšavaju funkcionalnost dizajnirane pločice. Poznavanje kako analognih, tako i digitalnih kola olakšava dizajniranje pločice i razumevanje *datasheet-a*. Neophodno je izučiti *datasheet* sa razumevanjem kako bi se ispoštovala sva ograničenja korišćenog integriranog kola i kako bi svi pinovi komponente bili ispravno povezani.

Izuzetno bitne stavke o kojima treba razmišljati pre kreiranja pločice su kvalitet i primena. Pri kreiranju pločice ne bi trebalo koristiti komponente koje su se pojavile davno na tržištu. Upravo zbog razvoja u elektro industriji se svakim danom na tržištu pojavljuju nove komponente sa boljim performansama. Kreirani uređaj mora nuditi inovacije i određena poboljšanja u odnosu na druge uređaje iz iste kategorije da bi uspeo na tržištu. Potrebno je napraviti kompromis između kvaliteta korišćenog integriranog kola i cene same pločice. Poželjno je analizom tržišta utvrditi kolika je potreba za uređajima iz kategorije za koju se kreira uređaj. Ukoliko nema potražnje za sličnim uređajima, nije isplativo kreirati još uređaja iste kategorije. Do male potražnje dolazi ukoliko uređaj ima uski opseg primene ili ukoliko se ne može koristiti u svakodnevnicu ili uobičajenim situacijama (primer: detektor gama zraka se koristi za merenje radioaktivnog zračenja koje je štetno i nije česta pojava). Što je opseg primene širi, to će uređaj bolje prolaziti na tržištu jer se može koristiti u različite svrhe.

U šestom poglavlju je opisan razvoj softvera za pločice. Za razvoj softvera pločica je neophodno barem osnovno poznavanje programskih jezika. Biblioteke koje sadrži *mikroSDK* pokrivaju velik broj funkcija koje se uobičajeno koriste pri pisanju programa. Ove biblioteke su dostupne za preuzimanje i korišćenje. Primenom ovih biblioteka korisnik može fokus postaviti na kreiranje funkcija usko vezanih za integrisano kolo na pločici bez razmišljanja o kreiranju sporednih funkcija za komunikacije, rad sa ulaznim i izlaznim pinovima, portovima, generisanjem PWM signala... Sam proces razvoja programa biva pojednostavljen i manje vremenski zahtevan. Velika prednost *mikroSDK* biblioteka u odnosu na druge biblioteke je ta što kôd koji sadrži funkcije iz *mikroSDK* biblioteka postaje hardverski nezavisran. *MikroSDK* biblioteke su razvijene tako da bez ikakvih izmena imaju istu funkcionalnost pri korišćenju različitih hardvera. Implementacijom funkcija iz *mikroSDK* se dobija portabilni kôd koji se može koristiti na različitim platformama i arhitekturama bez promene koda.

Po mom mišljenju je pomoću *mikroBUS* standarda uvedena jedna izuzetno korisna i edukativna inovacija u svetu elektronike. *Click* pločice same, ili u kombinaciji sa drugim pločicama pružaju nebrojane mogućnosti za korisnike. Zbog programibilnosti je moguće samo jednom pločicom kreirati više različitih projekata. Programeru se ostavlja sloboda oko izbora

primene za koju će koristiti pločicu. Kombinacijom više pločica se mogu kreirati izuzetno dobri multifunkcionalni sistemi koje je moguće personalizovati po želji.

Kreiranje hardvera je izuzetno zahtevno ali i poučno. Zahteva određena predznanja bez kojih bi ovaj proces bio otežan i trajao duže. Znanja iz predmeta poput *Analogna mikroelektronska kola*, *Digitalna elektronika*, *Praktična elektronika*, *Senzori i aktuatori* stvaraju dobru podlogu znanja i pomažu razumevanju kreiranja uređaja, bilo da li su ti uređaji *Click* pločice ili neka druga vrsta hardvera. Svakom inžinjeru je potrebno dodatno praktično znanje kojim će usavršiti svoje veštine i već stečena znanja. Još jedna stavka u znanju koju bi svaki student elektronike ili embeded programiranja trebao da usavrši je analiza *datasheet*-ova komponenti bez kojih je skoro pa i nemoguće kreirati funkcionalni uređaj bez grešaka.

Gledano iz ugla programiranja, ove pločice su osim za kreiranje uređaja idealne za učenje programiranja pomoću mikrokontrolera. Razvijanjem biblioteka i programa za različite vrste pločica se stiče dobro iskustvo oko proučavanja *datasheet*-a, rada sa registarima, komunikacijom između dva ili više uređaja... Iako u početku bude potrebno više vremena za razvijanje programa, stekne se dovoljno znanja i veštine da se pisanje svakog narednog programa odvija mnogo brže. Dostupan je veliki broj već gotovih biblioteka i programa za pločice koji se mogu koristiti kao smernice za kreiranje sopstvenog programa. Osim proučavanja načina programiranja različitih integriranih kola, stiče se ili unapređuje i znanje oko principa rada istih.

Od dve kreirane pločice opisane u dokumentaciji, za sad je jedna zadovoljila potrebe firme i našla se na tržištu. Potencijalni problemi do kojih bi moglo doći kod rada sadrugom pločicom (*CapTouch 6 Click*) nastaju zbog male udaljenosti između susednih pedova. Uređaj radi po principu merenja promene kapacitivnosti. Pedovi su izuzetno blizu i malih dimenzija i postojala bi šansa da se, ukoliko se pritisne jedan ped, registruje promena kapacitivnosti i na susednom pedu. Ova promena bi bila znatno manja od promene koja nastaje direktnim dodirivanjem peda. Problem bi mogao da se reši softverski – naglašavanjem u kodu da se male promene zanemaruju.

Kreirani softver je testiran i pokazao je zadovoljavajuće rezultate. Traženo je kreiranje funkcija koje će ispisivati tačnu vrednost temperature i vlažnosti vazduha. Temperatura i vlažnost vazduha koji su prikazani nakon pokretanja programa su bili izuzetno precizni. Pri zagrevanju pločice se prikazivao porast u temperaturi, a njenim hlađenjem se temperatura smanjivala. Slični rezultati su dobijeni i za merenje vlage čija je vrednost rasla kada se iznad senzora postavila vlažna maramica. Poštovana su pravila lepog pisanja kodova radi njegovog lakšeg razumevanja, te je vođeno računa o intentaciji, razmacima, dodeljivanju smislenih imena promenljivima...

Uspešnim kreiranjem i završavanjem projekata i iz hardvera i iz softvera je stečeno znanje o kreiranju uređaja od početka do njegovog kraja i stavljena je tačka na jednu veliku celinu. Ovim radom je obuhvaćen je rad iz više naučnih oblasti održenih na fakultetu koje su dale dovoljnu osnovu znanja potrebnog za kreiranje projekata.

8. Literatura

- [1] ams: *AS8579 Datasheet*, <https://ams.com/as8579>, 2022.
- [2] ams: *TMF8820/21/28 Datasheet*, <https://ams.com/tmf8828>, 2022.
- [3] Branislav Batinić: *Praktikum za softverski paket Altium Designer 17 na predmetu Praktična elektronika*, Univerzitet u Novom Sadu, Fakultet Tehničkih Nauka, 2020.
- [4] Electronics and You: *SMD Components – Types of SMD Components*, <http://www.electronicsandyou.com/>
- [5] ElectronicsHub: *Basics of Serial Peripheral Interface (SPI)*, <https://www.electronicshub.org>
- [6] Goran Stojanović: *Istorijat elektronike* – prezentacija, Univerzitet u Novom Sadu, Fakultet Tehničkih Nauka
- [7] Honeywell: *HIH8130-021-001 Datasheet*, <sps.honeywell.com/>
- [8] Honeywell: *I2C Comms with Honeywell HumidIcon™ Digital Humidity/Temperature Sensors*, <sps.honeywell.com/>
- [9] Honeywell: *Using Alarms on the Honeywell HumidIcon™ Digital Humidity / Temperature Sensors*, <sps.honeywell.com/>
- [10] Ilija S. Stojanović: *Osnovi telekomunikacija*, Naučna knjiga, Beograd, 1987.
- [11] Ivan Mezei: materijal za predavanja iz predmeta *Računarska elektronika*, Univerzitet u Novom Sadu, Fakultet Tehničkih Nauka 2022.
- [12] Jovan Bajić: prezentacije sa predavanja iz predmeta *Optoelektronika*, Univerzitet u Novom Sadu, Fakultet Tehničkih Nauka 2022.
- [13] Jovan Bajić: prezentacije sa predavanja iz predmeta *Senzori i aktuatori*, Univerzitet u Novom Sadu, Fakultet Tehničkih Nauka 2022.

- [14] Mikroelektronika: *mikroBUSTM Standard specifications*,
<https://www.mikroe.com/mikrobus>
- [15] Mikroelektronika: *mikroSDK* – github repozitorijum,
https://github.com/MikroElektronika/mikrosdk_v2
- [16] Mikroelektronika: *mikroSDK 2.0 User manual*, <https://docs.mikroe.com/mikrosdk/user-manual/>
- [17] Ning-Cheng Lee: *Reflow Soldering, Processes and Troubleshooting: SMT, BGA, CSP and Flip Chip Technologies*, BH Newnes 2002.
- [18] sparkfun: *Getting Started with Necto Studio*,
<https://learn.sparkfun.com/tutorials/getting-started-with-necto-studio/all>
- [19] Texas Instruments: *I2C Bus Pullup Resistor Calculation*, <https://www.ti.com>, 2015.
- [20] Vladimir Rajs: prezentacije sa predavanja iz predmeta *Primenjena elektronika*, Univerzitet u Novom Sadu, Fakultet Tehničkih Nauka 2022.