Mjerenje mase primjenom tenzometarske trake

Risek, Roko

Undergraduate thesis / Završni rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University North / Sveučilište Sjever**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:122:420349

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-05-21**

Repository / Repozitorij:

University North Digital Repository

Sveučilište
Sjever

Završni rad br. 522/EL/2023

Mjerenje mase primjenom tenzometarske trake

Roko Risek, 4388/336

Varaždin, studeni 2023. godine

Sveučilište Sjever
Sveučilišni centar Varaždin
104. brigade 3, HR–42000 Varaždin

UNIVERSITY NORTH

Prijava završnog rada

Definiranje teme završnog rada i povjerenstva

ODJEL: Odjel za elektrotehniku

STUDIJ: preddiplomski stručni studij Elektrotehnika

PRISTUPNIK: Roko Risek | MATIČNI BROJ: 4388/336

DATUM: 14.06.2023 | KOLEGIJ: Elektronički sklopovi

NASLOV RADA: Mjerenje mase primjenom tenzometarske trake

NASLOV RADA NA ENGL. JEZIKU: Mass measurement using a tensiometric tape

MENTOR: Josip Srpak | ZVANJE: viši predavač

ČLANOVI POVJERENSTVA:

1. mr.sc. Ivan Šumiga, viši predavač
2. Josip Srpak, viši predavač
3. Doc.dr.sc. Dunja Srpak
4. Miroslav Horvatić, viši predavač
5.

Zadatak završnog rada

BROJ: 522/EL/2023

OPIS:

U ovom završnom radu potrebno je opisati principe mjerenja primjenom tenzometrijskih senzora. Nakon toga potrebno je razviti i izraditi vlastito rješenje modula za mjerenje mase, koje uključuje izradu elektroničkog sklopa zasnovanog na ADS1256 mjernom pretvorniku, kao i programsko rješenje za komunikaciju i prikaz rezultata mjerenja.
Potrebno je testirati funkcionalnost i točnost realiziranog modula, te navesti moguća unaprjeđenja u budućnosti.

REPUBLIKA HRVATSKA
SVEUČILIŠTE SJEVER

ZADATAK URUČEN: 19.06.2023. | POTPIS MENTORA: Josip Srpak

Sveučilište
Sjever
Odjel za Elektrotehniku

Završni rad br. 522/EL/2023

Mjerenje mase primjenom tenzometarske trake

Roko Risek, 4388/336

Predgovor

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

Ovom prilikom želio bih se zahvaliti mentoru i predavaču dipl.ing. Josipu Srpaku na strpljenju, pomoći i ustupljenom vremenu prilikom izrade završnog rada.

Posebno se zahvaljujem svojoj obitelji i prijateljima na podršci i strpljenju koju su mi pružili tijekom studiranja.

Sažetak

U prvom dijelu teoretskog rada opisan je princip rada mjernog osjetila i mjerenja sile. Obrađuju su podjele sila te univerzalni uređaj za mjerenje čvrstoće materijala. Potom je opisana metoda mjerenja sile primjenom tenzometrije. Naglasak se vrši na različitim primjenama tenzometarskih traka u Wheatstoneovom mostu, njihova podjela te način montaže.

U drugom dijelu praktičnog rada opisan je cijeli proces razvoja elektroničkog podsustava koji na izlazu daje brojčanu vrijednost mase na temelju mjerenja sile preko tenzometarske trake. Naglasak se vrši na samom razvijanju elektroničke pločice te njegove proizvodnje. Dodatno je opisano testiranje gotovog podsustava s različitim mjerenim vrijednostima i kalibriranjem.

Summery

In the first (theoretical) part of the paper, the principle of operation of the measuring sensor and force measurement is described. The division of forces and the universal device for measuring the strength of materials are discussed. Then the method of force measurement using tensometry is described. Emphasis is placed on the different applications of strain gauge in the Wheatstone bridge, their division and the method of assembly.

In the second (practical) part of the paper, the entire process of development of the electronic subsystem is described, which at the output gives a numerical value of the mass based on the measured force through the strain gauge. The emphasis is on the development of the electronic circuit board and its production. Testing of the finished subsystem with different measured values and calibration is additionally described.

Popis korištenih kratica

A/D Analog to Digital Converter
Analogno - digitalni pretvarač

AD Altium Designer
Programski alat za razvoj tiskanih elektroničkih pločica

ADC Analog - Digital Converter
Pretvarač analognog signala u digitalni signal

DC-DC Digital converter – Digital converter
Inverterski pretvarač digitalnog napona

DRC Design Rule Check
Funkcija unutar programskog alata Altium Designer s pomoću koje se vrši provjera dizajna elektroničke pločice prema unaprijed zadanim pravilima

EMI Electromagnetic Interference
Smetnje unutar električne mreže uzrokovane vanjskim faktorom djelovanjem elektromagnetske indukcije

HASL Hot Air Solder Leveling
Postupak nanošenja lemne paste na vodljive dijelove tiskane pločice

I/O Input/Output pins
Ulazno/izlazni pinovi

LDO Low - Dropout regulator
Regulator niskog ispada

MCU Microcontroller unit
Mikroprocesor

MUX Multiplexer
Multiplekser je selektor podatak koji bira na temelju nekoliko analognih ili digitalnih ulaznih podataka te jedan odabrani ulazni signal prosljeđuje na jedan izlaz

PCB Printed Circuit Board
Elektronička tiskana pločica

PGA Programmable Gain Amplifier
Elektroničko pojačalo čije se pojačanje može upravljati vanjskim digitalnim ili analognim signalom

RF RadioFrequency
Radiofrekvencija

RLC Spoj otpornika, zavojnice i kondenzatora
Spoj koji služi za filtriranje frekvencija unutar grane

SI Signal Interference
Ometanje signala

SPI Serial Peripheral Interface
Sučelje za serijsku komunikaciju podataka

SPS Sample per seconds (signal reading speed)
Uzorak po sekundi (brzina čitanja signala)

Popis fizikalnih veličina

a akceleracija, [m/s^2]

C kapacitet, [F]

d kružni promjer, [m]

ε_{sr} srednja deformacija/produljenje, [m]

f frekvencija, [Hz]

F sila, [N]

I jakost električne struja, [A]

l duljina, [m]

L induktivitet, [H]

m masa, [kg]

R električni otpor, [Ω]

U napon, [V]

Um mjereni napon, [V]

Uo ulazni istosmjerni napon, [V]

ε deformacija mjernog objekta, [µm,m]

σ naprezanje na mjestu deformacije, [N/mm^2, MPa]

Ao ploština poprečnog presjeka žice, [mm^2]

Ro otpor žice tenzometra, [Ω]

ρ električni otpor materijala, [Ωm]

ΔA poprečni presjek nakon produljenja, [mm^2]

ΔR otpor nakon produljenja, [Ω]

Δl produljenje, [m]

Sadržaj

1. Uvod 1
2. Mjerno osjetilo 2
2.1. Mjerenje sile 3
3. Metoda tenzometrije 5
3.1. Elektrootpornički tenzometri 6
3.2. Mjerenje u spoju Wheatstonovog mosta 10
4. Razvoj senzora 13
4.1. Altium Designer 14
4.2. Napajanje 14
4.3. Generator takta 17
4.4. Ulazni pinovi senzora 18
4.5. Izlazni pinovi/MCU 18
4.6. Uloga i rad ADS 1256 sklopa 19
5. Dizajn i raspored tiskane pločice 22
6. Proizvodnja tiskane pločice 28
7. Programski kod 30
7.1. Serijska komunikacija (SPI) 31
7.2. Asinkrona komunikacija 31
7.3. Sinkrona komunikacija 32
7.4. Prijenos podataka 33
7.5. Chip Select (CS) veza 34
7.6. Stvaranje library datoteke senzora za povezivanje sa mikrokontrolerom 35
7.6.1. Header file 35
7.6.2. Source file 38
7.7. Povezivanje senzora sa mikrokontrolerom 42
8. Testiranje senzora 46
9. Pokazna maketa 50
10. Zaključak 51
11. Literatura 53
Popis slika 54

1. Uvod

Tema završnog rada je izrada elektroničkog podsustava za mjerenje mase primjenom tenzometarske trake. Mjerna osjetila (senzori) su jedni od tehnoloških većih postignuća. Omogućuju lakše, točno i precizno mjerenje fizikalnih veličina. Napretkom i tehnološkim razvojem danas su senzori jedni od neizbježnih sustava svuda oko nas. Koriste se u svakoj industriji te su implementirani u svim uređajima kojima se čovječanstvo koristi. Kroz ovaj rad opisan je cijeli proces izrade jednog takvog sustava za primjenu u industrijske svrhe.

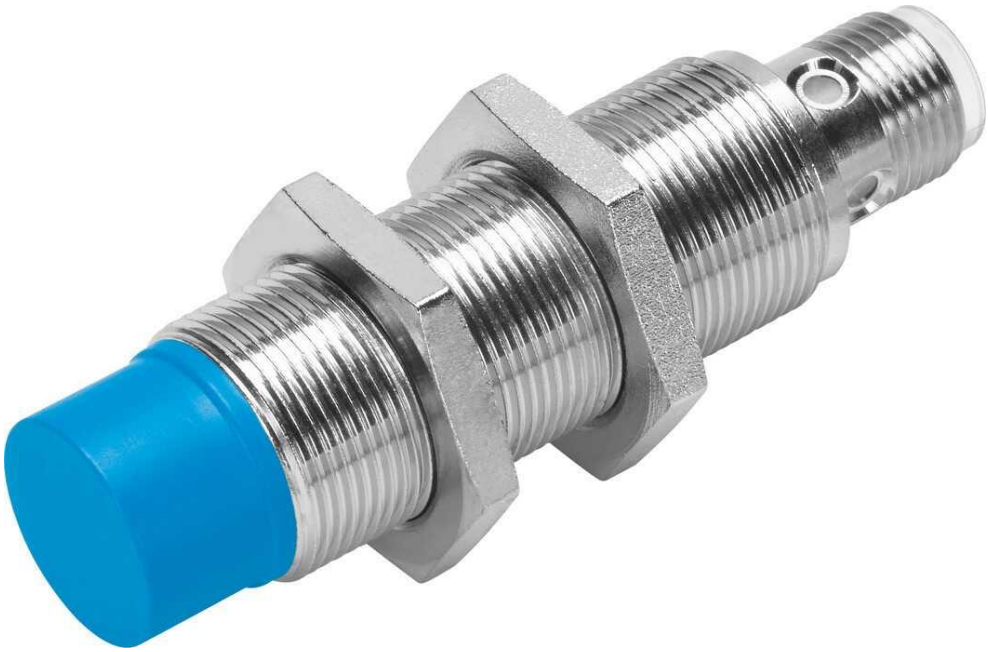
Teorijski dio rada obrađuje sami proces mjerenja sile te njegove podjele. Opisano je mjerenje sile metodom tenzometrije. Kroz rad su istaknute različite vrste tenzometarskih traka, no naglasak se vrši na elektrootporničkim mjernim trakama. Na kraju teorijskog dijela rada opisano je mjerenje u spoju Wheatstoneovog mosta te i njihova izvedba za mjerenje jedne ili više mjernih točaka.

U fizičkom dijelu rada opisan je cijeli proces izrade sustava. Od dizajniranja elektroničke pločice sve do proizvodnje. Kroz proces razvoja hardverskog dijela rada istaknute su izabrane elektroničke komponente te njihove funkcije unutar sklopa. Proces filtriranja signala izveden je korištenjem brojnih RLC krugova te kondenzatorima. Isto tako opisan je proces razvoja softverskog dijela rada u kojem je detaljno razrađen proces pisanja knjižnice za povezivanje senzora s mikrokontrolerom. Izvorni kod implementiran je unutar glavnog ADS 1256 sklopa. Na samom kraju rada obavljeno je testiranje uređaja kroz ispitna mjerenja preko pokazne makete. Mjerenje je izvedeno putem pet ispitnih objekta kroz koje se očituje točnost mjerenja sustava.

2. Mjerno osjetilo

Mjerno osjetilo ili senzor je dio mjernog sustava koji je u izravnom dodiru s mjernom veličinom te daje izlazni signal ovisan o njegovom iznosu. Senzor ima zadatak da na temelju ulaznog signala (fizičkog parametra) pretvara u signal koji je moguće mjeriti električno. Ulazni parametri senzora mogu biti toplina, vlaga, svjetlost, sila i sl. Izlazni parametri se mogu prikazati kao vrsta signala koji je čitak i razumljiv na zaslonu senzora. [1]

Mjerna osjetila su sve više zastupljena u svijetu. Njihova integriranost unutar različitih grana znanosti dovela je poboljšanje u tehnološkom svijetu. Često su ugrađeni unutar strojeva, automobila, kuhinjskih uređaja i sl. Vrlo je važno da senzori budu što manjih dimenzija ne smanjujući njihovu efikasnost i preciznost. Osjetljivost senzora s godinama je rasla dok su cijene istih padale.



Slika 2.1 Primjer induktivnog senzora [2]

2.1. Mjerenje sile

Sila je vektorska fizikalna veličina koja ima zadani smjer i intezitet. Opisuje djelovanje tijela i njegove okoline. Oznaka za silu je F, a mjerna jedinica je Newton (N).To je sila koja masi od 1 kg daje ubrzanje od 1 m/s^2. Sila se ostvaruje djelovanjem tijela mase m, lokalnom zemaljskom gravitacijskom ubrzanju g, čime djeluje na potporu postignutom silom.[3]

Definiranje sile vrši se istaknutom formulom.

$$F = m \cdot a \qquad (3 - 1)$$

- masa, m - mjera inercije ili tromosti tijela
- akceleracija, a – vektorska fizikalna veličina kojom se opisuje ubrzanje tijela

Podjela sila [3]:

- Konstantne - rezultiraju nakon fizičkog kontakta između dva tijela.
- Volumenske - vezane su uz postojanje polja sila u nekom prostoru. Djeluju istovremeno na sve čestice tijela.

Mjerenje sile predstavlja osnovu za određivanje mehaničkih svojstva nekog materijala. Na osnovu mjerenja sile dimenzioniraju se pojedini dijelovi uređaja. Kidalice su mjerni instrumenti za vlačno ispitivanje čvrstoće materijala. Univerzalna kidalica osim vlačne čvrstoće ispituje i tlačnu čvrstoću te čvrstoću na odrez i savijanje[4].

Kidalica se sastoji iz [4] :

- Uređaja za nametanje opterećenja
- Uređaja za mjerenje sile
- Uređaja za mjerenje deformacija ispitnog materijala
- Uređaja za ispis dijagrama naprezanja

Slika 2.2 Primjer univerzalne kidalice [4]

3. Metoda tenzometrije

U praktičnom mjerenju sile, mjere se veličine koje ovise o tlačnom i vlačnom naprezanju te se potom računski određuju njihovi iznosi. Mjerenjem produljenja Δl između dvije točke na površini ispitivanog objekta u odnosu na njihovo početno stanje mjeri se deformacija.

Definicija za srednju deformaciju glasi [5]:

$$\varepsilon_{sr} = \frac{\Delta l}{l} \qquad (5 - 1)$$

- produljenje, Δl - ovisi o samoj konstrukciji
- duljina, l

Tokom linearne deformacije tijela očituje se glavna deformacija koja se pruža po sredini mjerne dužine te joj dužina odgovara vrijednosti srednje deformacije. Ispravno mjerenje srednje deformacije vrši se uzevši što manju mjernu duljinu da se mjerenjem produljenja dobiju ispravne vrijednosti. Duljina l ovisi o samoj konstrukciji. U praksi se danas najviše koriste elektrootpornički tenzometri [5].

Prema principu razlikuju se razni tipovi tenzometara [5]:

- mehanički
- akustički
- optički
- električni

Električni tenzometri mogu biti [5]:

- elektrootpornički
- elektrokapacitivni
- elektroinduktivni

3.1. Elektrootpornički tenzometri

U industriji se na najviše koriste elektrootpornički tenzometri (mjerne trake), a sama metoda mjerenja se naziva metoda tenzometrije. Postavljanjem žice na površinu neopterećene konstrukcije, prilikom opterećenja će se stvoriti deformacija nad žicom [5].

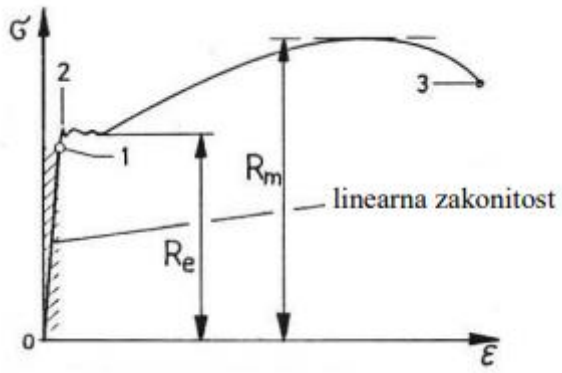
Glavni princip tenzometarskog mjerenja leži u tvrdnji da se deformacijom mjerne žice mijenja i njen otpor, odnosno određuje se iznos deformacije u pravcu osi trake [5]. Mjerenje deformacije se vodi Hookovim zakonom.

Formula Hookeova zakona za jednoosno stanje naprezanja:

$$\sigma = E \cdot \varepsilon \qquad (5 - 2)$$

- deformacija mjernog objekta, ε
- naprezanje na mjestu deformacije, σ

Izgled grafičkog dijagrama rastezanja za trgovačke čelike:



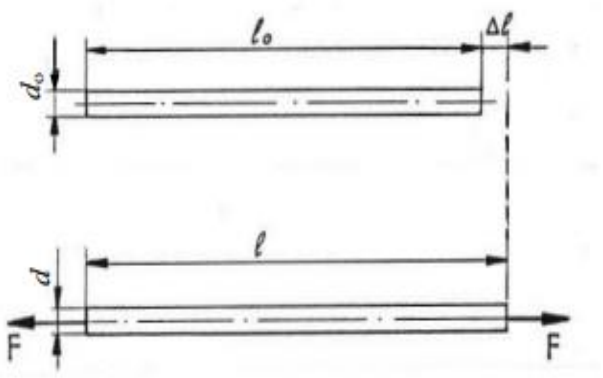
Slika 3.1 Dijagram rastezanja za niskougljični čelik [5]

- *Re* - naprezanje tečenja (σT)
- Rm - rastezna (vlačna) čvrstoća (σM)

Linearnost obuhvaćena Hookeovim zakonom [5]:

- 1 - granica proporcionalnosti (σP)
- 2 - granica tečenja
- 3 - lom

Uslijed opterećenja tenzometarske trake mijenja se njen električni otpor, a mijenjaju se i poprečni presjek te struktura materijala žice trake. Najčešći otpor žice mjerne trake iznosi 120 **Ω** [5].



Slika 3.2 Ovisnost otpora od značajki mjerne trake [5]

$$Ro = \rho \cdot \frac{lo}{Ao} \qquad (5 - 3)$$

- početna duljina žice, lo
- ploština poprečnog presjeka žice, Ao
- električni otpor materijala, ρ

$$Ao = \frac{\pi \cdot do^2}{4} \qquad (5 - 4)$$

- ploština poprečnog presjeka žice, Ao
- početna duljina, do

$$R = \rho \cdot \frac{l}{A} = \rho \cdot l \cdot \frac{4}{\pi \cdot d^2} \qquad (5 - 5)$$

- duljina žice, l
- ploština poprečnog presjeka žice, A
- električni otpor materijala, ρ
- duljina, d

Kada se žica produlji za Δl, istovremeno se smanjuje poprečni presjek ΔA, a električni otpor mijenja iznos za ΔR.

Relativna promjena otpora žice mjerne trake [5]:

$$\frac{\Delta R}{R} = \frac{\Delta \rho}{\rho} + \frac{\Delta l}{l} - \frac{\Delta A}{A} \qquad (5 - 6)$$

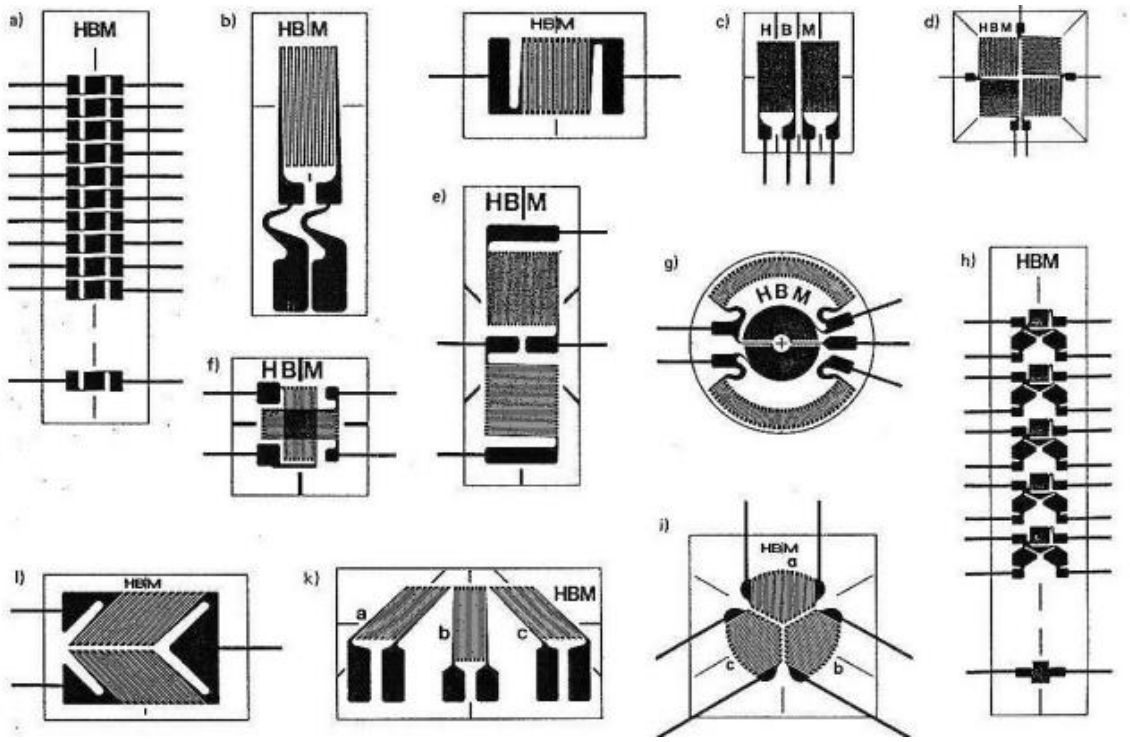
- ploština presjeka nakon produljenja, ΔA
- ploština presjeka, A
- otpor nakon produljenja, ΔR
- električni otpor, R
- produljenje, Δl
- duljina, l

Promjena samog otpora tenzometra i deformacije jednaka je deformaciji ispitivanog objekta.

$$\mathcal{E} = \frac{1}{k} * \frac{\Delta R}{R} \qquad (5 - 7)$$

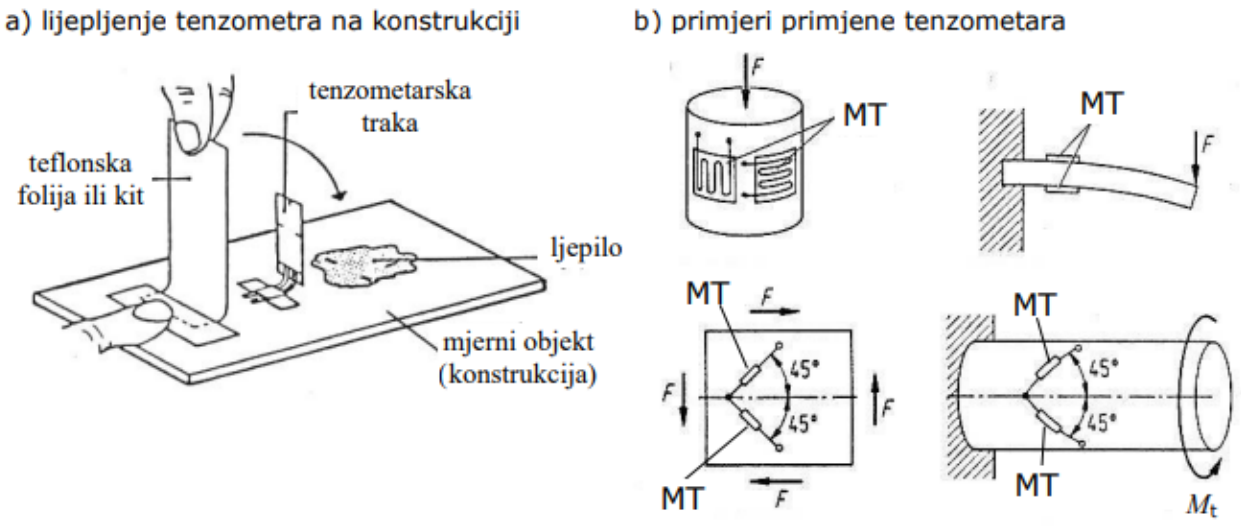
- osjetljivost mjerne trake (faktor tenzometra), k
- promjena električnog otpora, ΔR
- električni otpor, R

Unutar tenzometarske trake najčešće se koristi vodič od konstantina, promjera 20 μm. Žica je zbog svoje duljine savijena vijugavo i zalijepljena između dva listića plastike. U praksi se najviše proizvode trake u vrijednostima 350 Ω, 600 Ω ili 120 Ω. U slučaju mjerenja deformacije na čeličnim konstrukcijama koriste se folijski tenzometri [5].



Slika 3.3 Vrste elektrootporničkih mjernih traka [5]

Tenzometarska traka se lijepi na čistu površinu mjernog objekta. Kod upotrebe jednoosnih traka bitno je da su pravac glavnog naprezanja σ1 ili σ2 u ravnini s pravcem žice na traci. Ovisno o pravcu deformacije ovisi i pravac samog postavljanja tenzometarske trake [5].

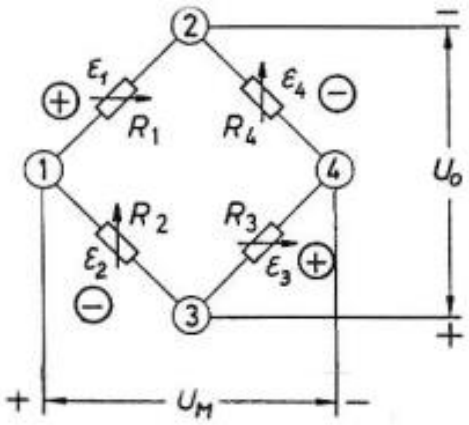


Slika 3.4 Postavljanje mjerne trake na površinu objekta [5]

Kada je potrebno izmjeriti više mjernih točaka unutar konstrukcije, sva su mjesta spojena preko prekidača te se uključuju prema potrebi u krug mjernog mosta, zapisujući izmjerene vrijednosti opterećenja. Vrijednosti iz mjernog pojačala kod neopterećene konstrukcije se oduzimaju ili zbrajaju s nul - vrijednostima očitanih te tako se određuju deformacije u pojedinim točkama. Danas se u praksi svim mjernim trakama prethodno mjerenju, postavlja nulto stanje, a kod mjerenja pod opterećenjem očitane vrijednosti su stvarne deformacije u mjernim točkama [5].

3.2. Mjerenje u spoju Wheatstonovog mosta

Za mjerenje promjene otpora tenzometra ΔR, kod deformacije konstrukcije, upotrebljavaju se najčešće potenciometar i mjerni most. Mjerni signal tenzometra se dovodi u mosni spoj kako bi se izmjerila vrijednost napona. Dok je njegova vrijednost u zavisnosti od otklona mjernog instrumenta iz ravnotežnog položaja, a određuje iz razlike potencijala između spojišta grana mosnog spoja [5].



Slika 3.5 Wheatstonov mjerni most [5]

- Uo - ulazni istosmjerni napon
- Um - mjereni napon, (signal)
- 2, 3 – naponska dijagonala mosta
- 1, 4 – mjerna dijagonala mosta
- R1..R4.. - otpori tenzometarske trake
- Ɛ1..Ɛ4 - duljinske deformacije u mjernim točkama na konstrukciji

$$Um = U_o * (\frac{R1}{R1+R2} - \frac{R4}{R3+R4}) = U_o * \frac{R1*R3-R2*R4}{(R1+R2)*(R3+R4)} \qquad (5 - 8)$$

- električni otpor, (R1, R2, R3, R4) - otpornici unutar mjernog mosta
- ulazni istosmjerni napon, U_o

Ravnoteža mjernog mosta se postiže samo kada je Um = 0 ili je omjer priključenih otpornika

$$\frac{R1}{R2} = \frac{R3}{R4}.$$

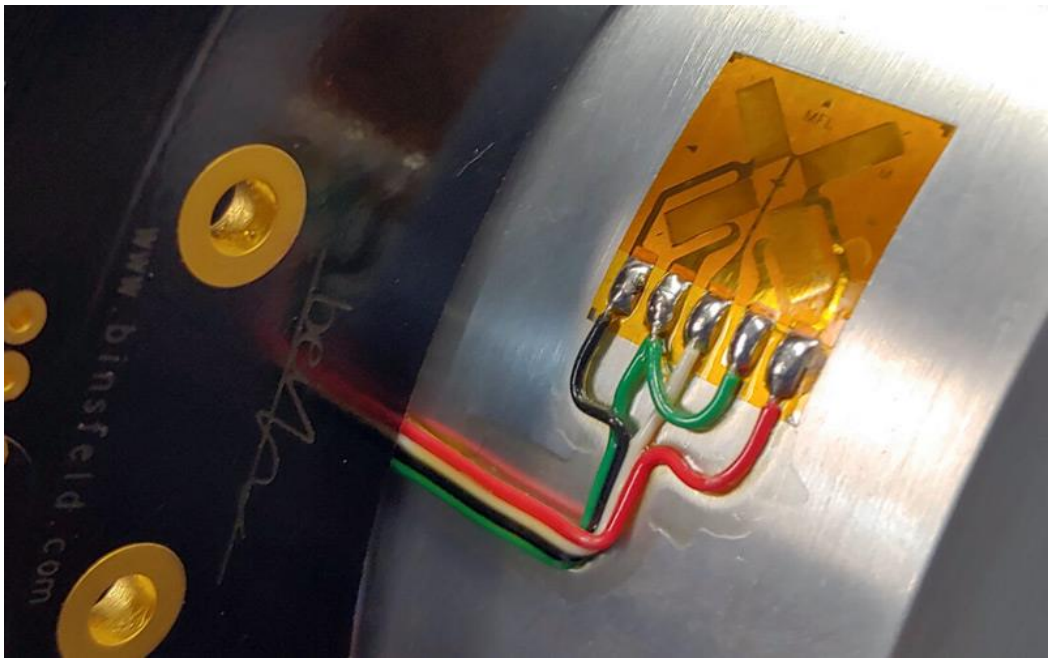
Promjenom vrijednosti otpornika R1 - R4 za iznos ΔR, ravnoteža mosta će se promijeniti te će se između točaka 1 i 4 pojaviti izlazni napon Um, čija je vrijednost proporcionalna s promjenom mjerene veličine na osjetilu [5].

Prilikom mjerenja tenzometarskom trakom na mjestima otpornika, mjereni signal na izlazu mosta može biti [5]:

- pozitivan – kada je $\mathcal{E}1 > \mathcal{E}2$ ili je $\mathcal{E}3 > \mathcal{E}4$
- negativan – kada je $\mathcal{E}1 < \mathcal{E}2$ ili je $\mathcal{E}3 < \mathcal{E}4$

Unutar uređaja koji imaju implementirani Wheatstoneov most mogu biti [5]:

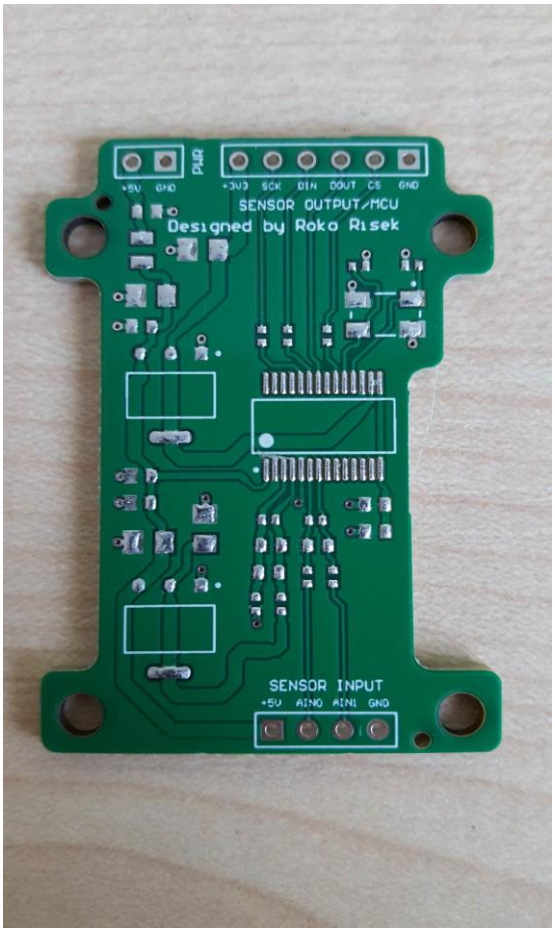
- sva četiri otpora aktivna(puni most)
- dva otpora aktivna (polu-most)
- samo jedan (četvrtinski most) aktivan.



Slika 3.6 Primjer montirane tenzometarske trake na mjerni objekt [5]

4. Razvoj senzora

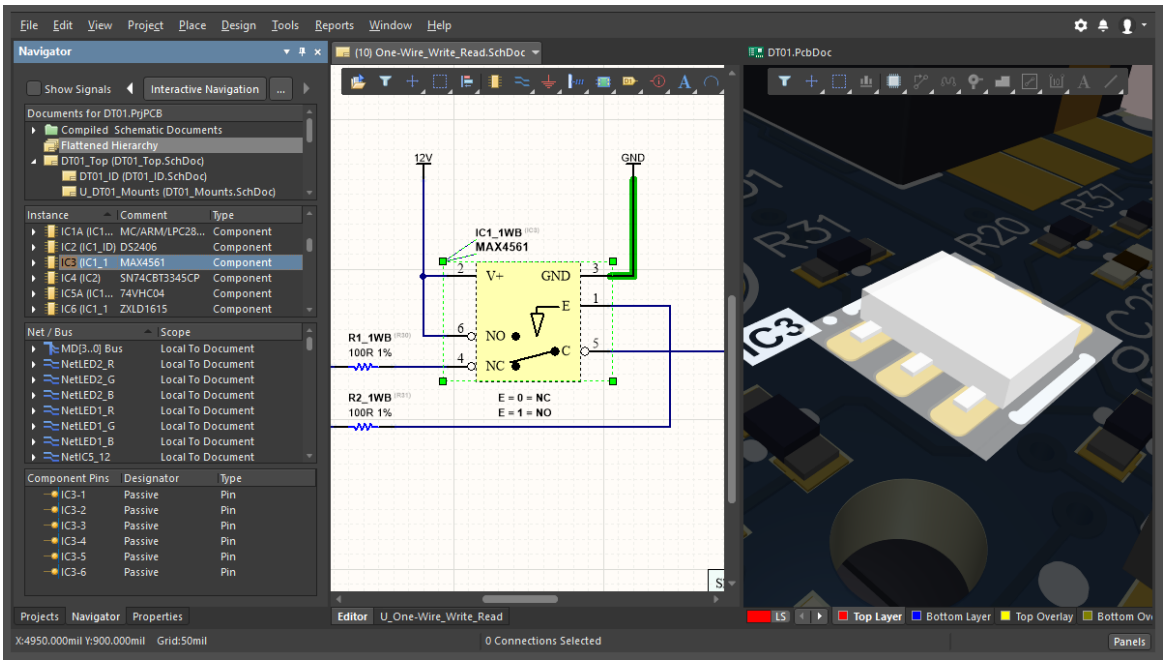
Razvoj senzora za mjerenje mase započelo je koristeći razvojni alat Altium Designer. Shema se sastoji od dva dijela. U primarnom dijelu sheme nalazi se ADC pretvarača s integriranim operacijskim pojačalom čija je svrha obrada analognog signala dobivenog s tenziometarske trake. Također s pretvarača su izvedeni glavni ulazni pinovi za napajanje i komunikaciju s tenzometarskom trakom te komunikaciju s mikrokontrolerom. Pomoću SPI serijske komunikacije postignuta je izmjena i daljnja obrada signala. Sekundarni dio sheme sastoji se od para LDO (Low-dropout regulator) koji vrše pretvaranje ulaznog napona na niže zadanu vrijednost. Uz popratne kondenzatore za filtriranje viših harmonika u mreži nalazi se i vanjski kristalni oscilator koji služi za precizno postavljanje signala na određenu frekvenciju.



Slika 4.1 Tiskana pločica senzora

4.1. Altium Designer

Altium Designer (AD) je softverski paket za dizajniranje i realizaciju elektroničkih tiskanih pločica. Razvila ga je australska tvrtka Altium Limited. AD pruža jednostavan pristup za stvaranje shema, rasporeda tiskane pločice te popratne dokumentacije za proizvodnju. Razvojni alat pruža korisniku da na jednom mjestu odradi cijeli projekt te isporuči visoko kvalitetan proizvod.

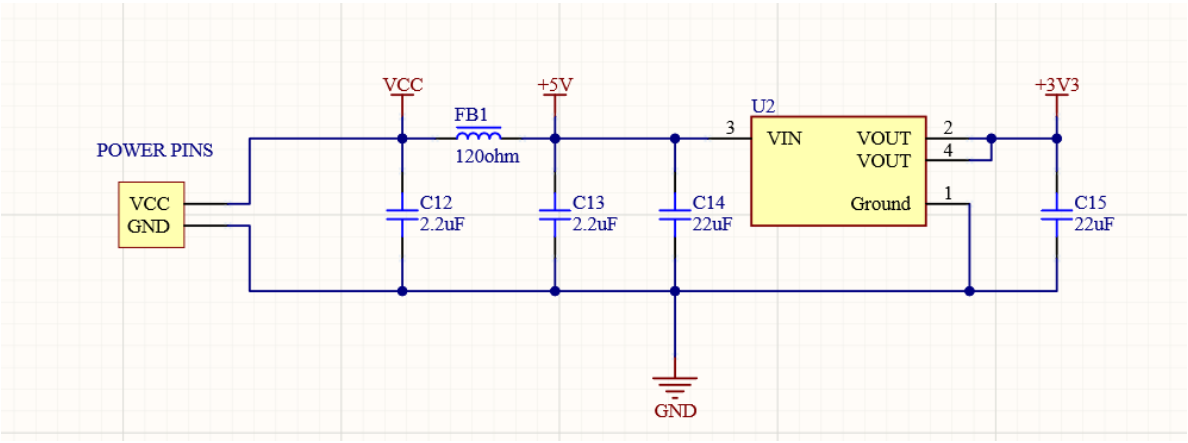


Slika 4.2 Softverski paket Altium Designer [10]

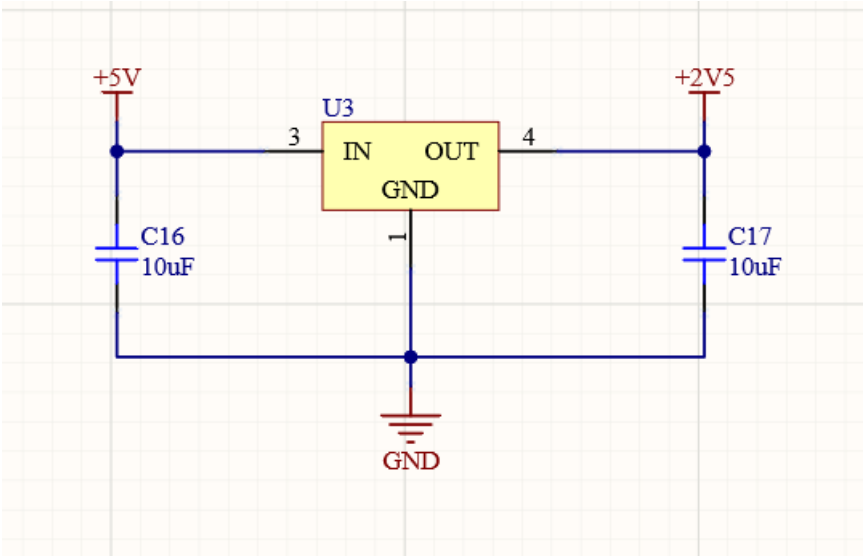
4.2. Napajanje

Izvor napajanja senzora dovodi se s bilo kojeg istosmjernog napajanja koje pruža adekvatnih 5 V/1 A. Takav napon zatim dolazi do petlje s 2 paralelno spojena kondenzatora vrijednosti 2.2 uF i RLC člana vrijednosti 120 Ω. Takvi kondenzatori u struci često imaju naziv „bulk“ kondenzatori. Funkcija takvih kondenzatora je sprječavanje prevelikog pada izlazne snage napajanja tijekom razdoblja kada je struja nedostupna. Za istosmjerne linearne izvore to bi se dogodilo kada je linijski napon blizu vrijednosti nule. Također se odnose na sklop u cjelini, koji sadrži više mrežnih sučelja. RLC član se koristi kao zaštita od EMI. Također potiskuje visokofrekventne harmonike iz mreže. Nakon što se ulazni napon „donekle“ filtrirao tada dolazi na prvi LDO. Regulator niskog ispada AMS1117 - 3.3 u ovom slučaju služi kao jednostavna izvedba DC - DC pretvarača. Pretvara ulaznih 5 V u 3.3 V. Svrha pretvaranje jest osigurati

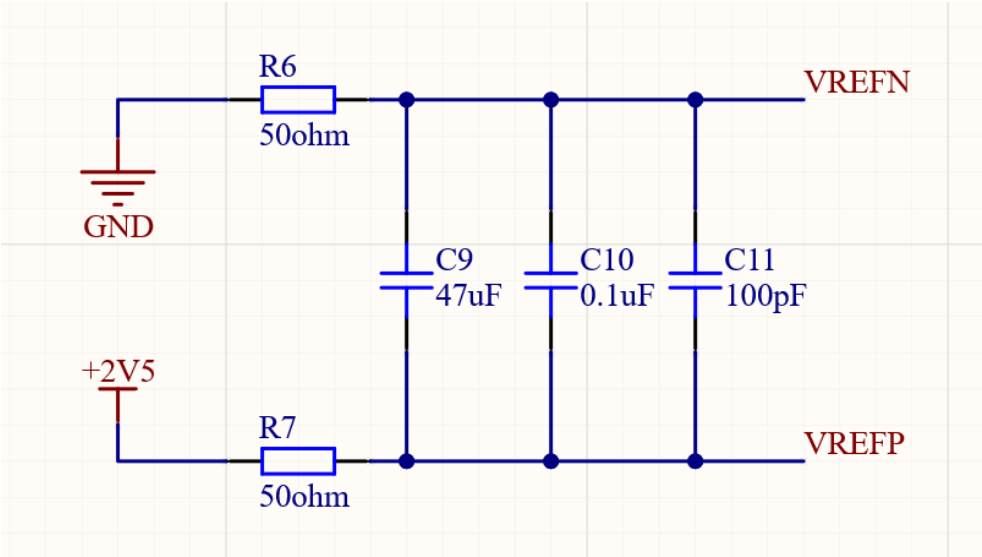
adekvatni digitalni napon glavnom ADS 1256 sklopu. Paralelno, napon se dijeli na drugi LDO. Regulator niskog ispada AMS1117 - 2.5 koji služi za opskrbu napona 2.5 V. Svrha takvog napona je osigurati maksimalni rad integriranog PGA, odnosno njegovu maksimalnu rezoluciju. Svaki LDO sadrži svoje „decoupling“ kondenzatore. Paralelni spoj takvih kondenzatora ostvaruje dovoljnu količinu naboja te normalan rad prilikom naglih strujnih povišenja. Takvi kondenzatori se najčešće postavljaju lokalno (u pravilu 1 ili 2 po logičkom čipu) te su namijenjeni opskrbi strujom u kraćim vremenskim razdobljima. U pravilu su manjih vrijednosti nego „bulk“ kondenzatori.



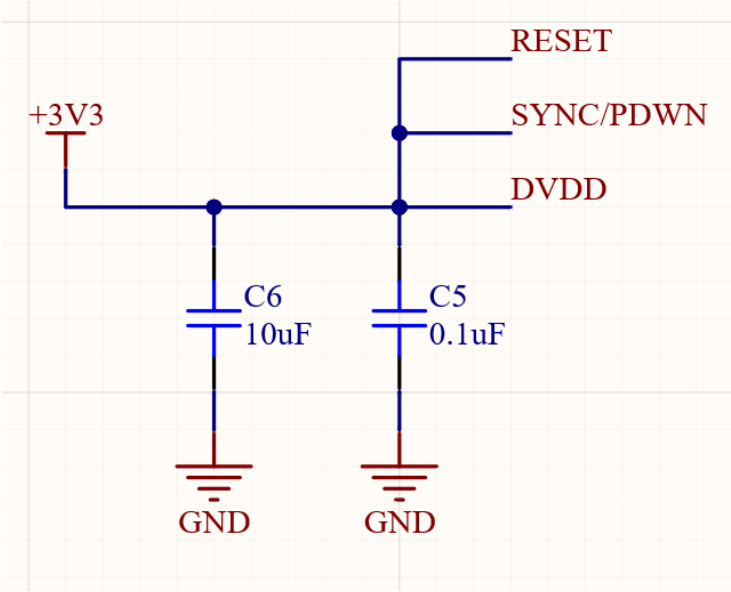
Slika 4.3 Napajanje senzora - AMS1117 - 3.3



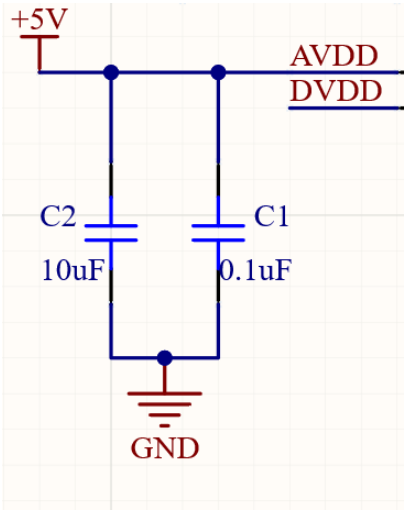
Slika 4.4 Napajanje senzora - AMS1117 - 2.5



Slika 4.5 Decoupling kondenzatori operacijskog pojačala



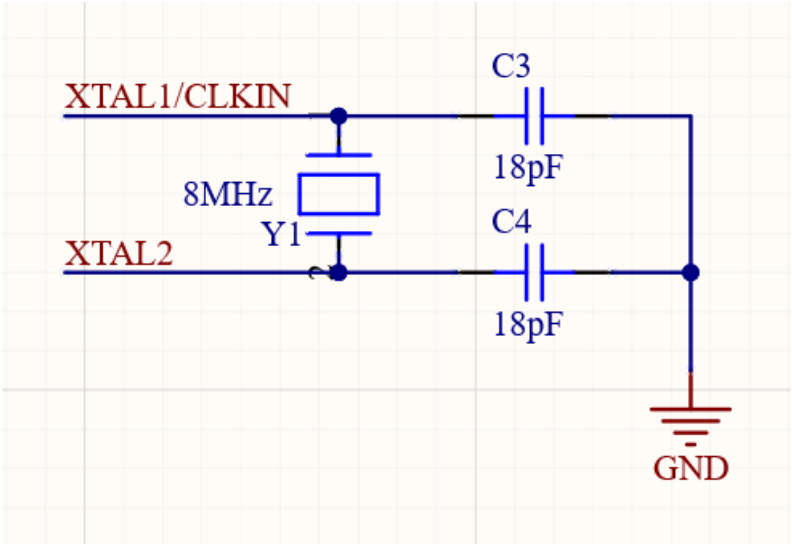
Slika 4.6 Decoupling kondenzatori na digitalnom izvoru +3.3 V



Slika 4.7 Decoupling kondenzatori na analognom izvoru +5 V

4.3. Generator takta

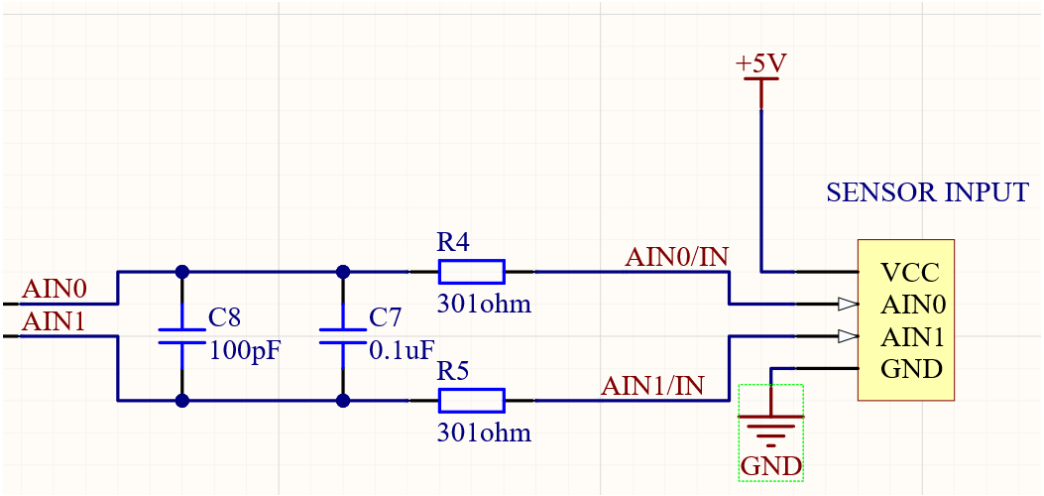
Glavni izvor takta osiguran je s pomoću vanjskog kristalnog oscilatora. Kristalni oscilator je pasivna komponenta koja ima svojstvo generiranja periodičnog oscilirajućeg izlaza. Komponenta ima određenu frekvencijsku propusnost. Koristi mehaničku rezonanciju kristala za stvaranje signala precizne frekvencije. Kod stvaranja takta potrebno je osigurati odgovarajući set „decoupling“ kondenzatora. Svrha takvih kondenzatora omogućavanje stabilne taktne frekvencije. Pozicioniranje samog kristala bitno je da bude što bliže samom ADS 1256 sklopu. Pinovi XTAL1/CLKIN i XTAL2 se koriste za komunikaciju s kristalom. Kada su iskorišteni ne mogu se koristit za pokretanje bilo koje druge logike.



Slika 4.8 Vanjski generator takta

4.4. Ulazni pinovi senzora

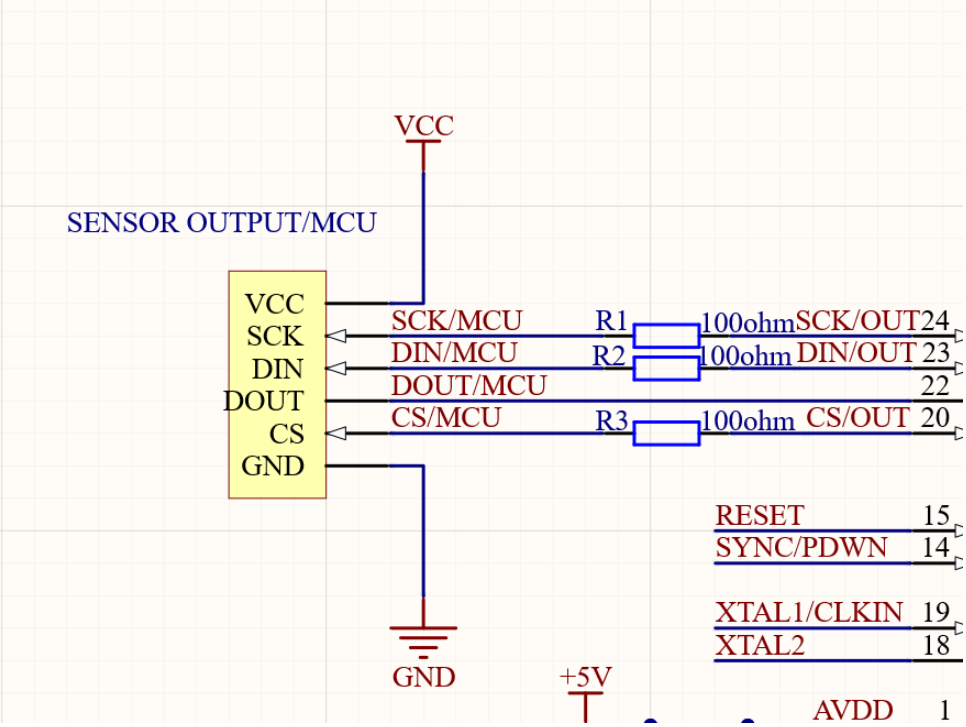
Konektor „Sensor Input“ sadrži 4 pina. Prvi i zadnji pin služe za napajanje. Analogni izvor +5 V doveden s ADS 1256 sklopa spojen je direktno na pozitivan i negativan izvod na tenzometarskoj traci. Pinovi AIN0 i AIN1 služe kao komunikacijski kanali. Kondenzatori C8 i C7 služe kao „decoupling“ kondenzatori koji filtriraju visokofrekventne smetnje. Otpornici R4 i R5 postavljaju nisko logičko stanje kada spojnice senzora nisu priključene na AIN0 i AIN1 pinove. Često se nazivaju „pull down“ otpornici. Sprječavaju krivo logičko stanje pri povezivanju s ADS 1256 sklopom.



Slika 4.9 Shema ulaznog konektora senzora

4.5. Izlazni pinovi/MCU

Konektor „Sensor Output/MCU“ sadrži 6 pinova. Prvi i zadnji pin služe za napajanje. Pin SCLK (Serial Clock) je karakteriziran „Schmitt-triggered“ ulazom s funkcijom čitanja i pisanja podataka određenim taktom s pinova DIN i DOUT. U podatkovnoj tablici je izričito naglašeno da pin SCLK ne smije imati nikakve povezane komponente. Razlog toga je da ne dolazi do slučajnog pomicanja podataka. Ako je SCLK u niskom logičkom stanju tijekom 32 DRDY razdoblja, serijsko sučelje će se resetirati te će sljedeći SCLK impuls započeti novi komunikacijski ciklus. Takva vrsta vremenskog ograničavanja koristi se za obnovu komunikacije kada je prijenos podataka prekinut. CS (Chip Select) omogućuje pojedini odabir uređaja kada isti dijele serijsku vezu. CS pin je također postavljen u nikom logičkom stanju tijekom trajanja komunikacije. Kada CS prelazi u visoko logičko stanje, serijsko sučelje se resetira. DIN pin se koristi za slanje podatak na ADS 1256 sklop dok je pin DOUT zaslužan za čitanje podataka iz ADS 1256 sklopa.



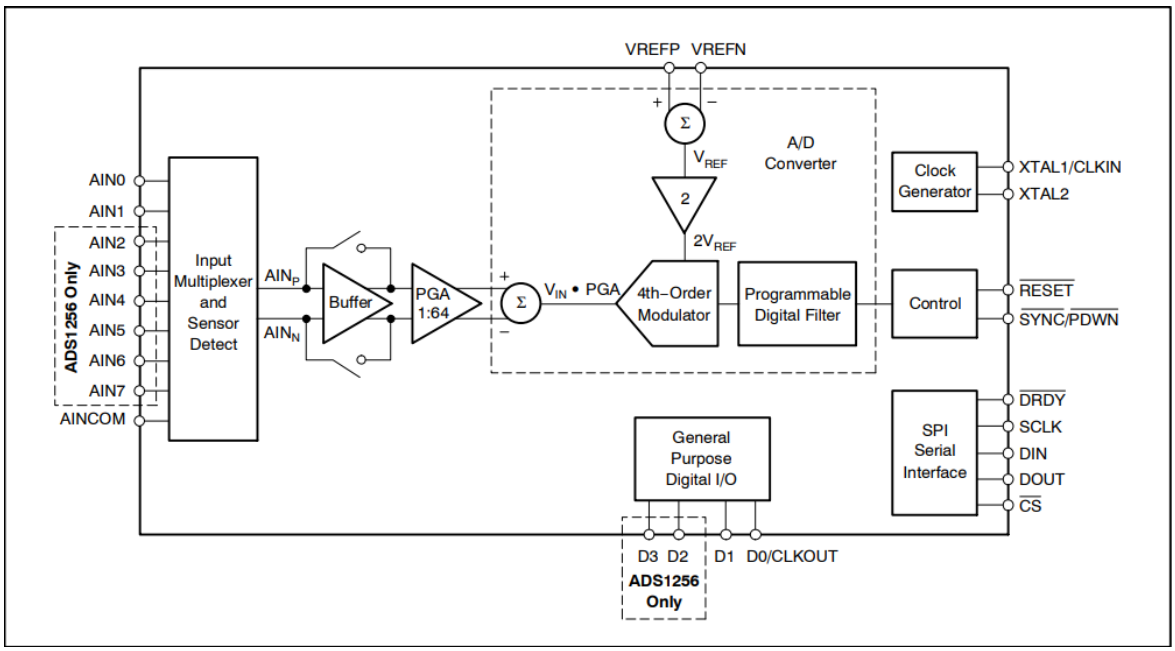
Slika 4.10 Shema Sensor Output/MCU konektora

4.6. Uloga i rad ADS 1256 sklopa

24 bitni analogno - digitalni pretvarač služi kao glavna upravljačka jedinica senzoru. Podržava četiri ili osam diferencijalna jednostrana ulaza. Pruža četiri digitalna I/O pina. Ulazni MUX određuje koji pinovi su povezani na A/D pretvarač. ADS 1256 sadrži funkciju provjere prekida ili kratkog spoja na vanjskom senzoru. Ulazni „buffer“ smanjuje opterećenje na ulazu sklopa pružajući impedanciju do 80 MΩ. Niskošumni PGA omogućuje različite razine pojačanja do indeks 64. Integrirani delta - sigma modulator služi za mjerenje pojačanog ulaznog diferencijalnog signala u odnosu na unutarnji referentni signal. Diferencijalna referenca se interno skalira faktorom 2 tako da je ulazni raspon pune skale ±2 VREF (za PGA = 1). Maksimalna brzina čitanja podataka je 30 SPS. Komunikacija se odvija preko SPI kompatibilnog serijskog sučelja. Ugrađeni registri pohranjuju brojne postavke za ulazni MUX:

- otkrivanje strujnih izvora
- PGA postavke
- brzina čitanja i pisanja podataka

Unutarnji ili vanjski kristal omogućuje izvor takta.



Slika 4.11 Blok dijagram ADS1256 sklopa [6]

ADS 1256 prima analogni signal za tenzometarske trake te ga dovodi na ulazne pinove AIN0 i AIN1. AIN0 je pozitivan pin dok je AIN0, negativan. Signal zatim ulazi na MUX gdje se radi provjera ispravnosti signala ili provjera kratkog spoja. Signal s ulaza dolazi s visokom impedancijom te prolazi kroz digitalni međuspremnik koji omogućuje povezivanje visoko - impedantnog izvora s nisko impedantnim teretom bez iskrivljenja signal. Signal zatim dolazi na PGA koje pojačava signal s indeksom 64 te ga šalje na daljnju obradu A/D pretvaraču. A/D pretvarač se sastoji od programabilnog digitalnog filtra i delta - sigma modulatora. Modulator koristi programibilni digitalni filter za obradu signala. Ulazna vrijednost signala se potom uspoređuje s referentnim diferencijalnim signalom.

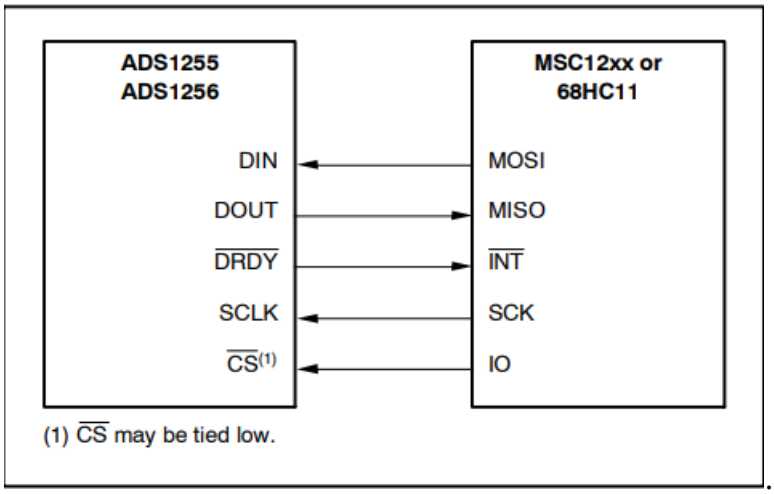
Napon na izlazu računa se prema formuli:

$$VIN = (AINP – AINN) \qquad (6 - 1)$$

Referentni diferencijalni napon se računa prema formuli:

$$VREF = (VREFP – VREFN) \qquad (6 - 2)$$

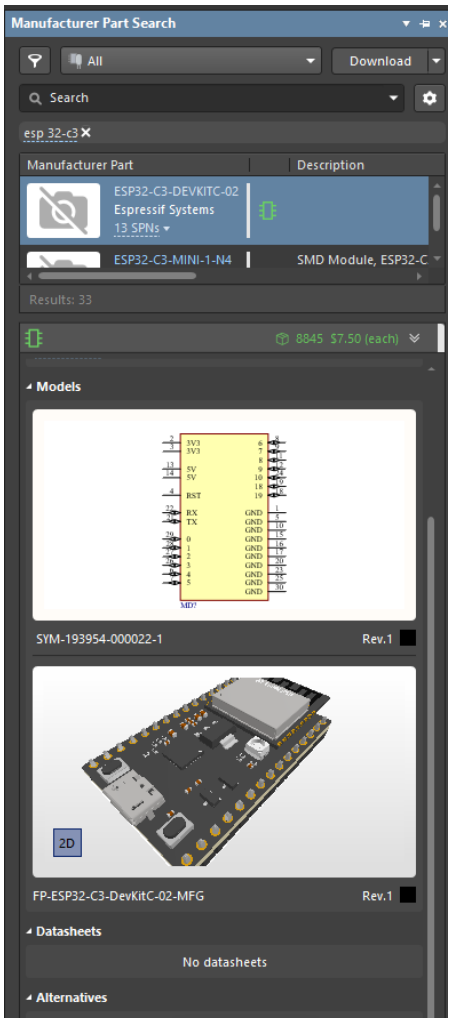
Vanjski kristalni oscilator služi za precizno postavljanje frekvencije za komunikaciju sklopa. Izlazni digitalni signal iz pretvarača šalje se na daljnju obradu preko SPI perifernog sučelja. Razlika napona dobivena mjerenjem tenzometarske trake se u sklopu pojačava. Odnosno povećava se rezolucija samog mjerenja. Takav signal se kasnije prenosi na Arduino Uno mikrokontroler koji s pomoću programskog koda mjerene vrijednosti napona pretvara u promjene mase u zadanom vremenskom periodu.



Slika 4.12 Prikazni spoj SPI sučelja s mikrokontrolerom MSC 12xx [6]

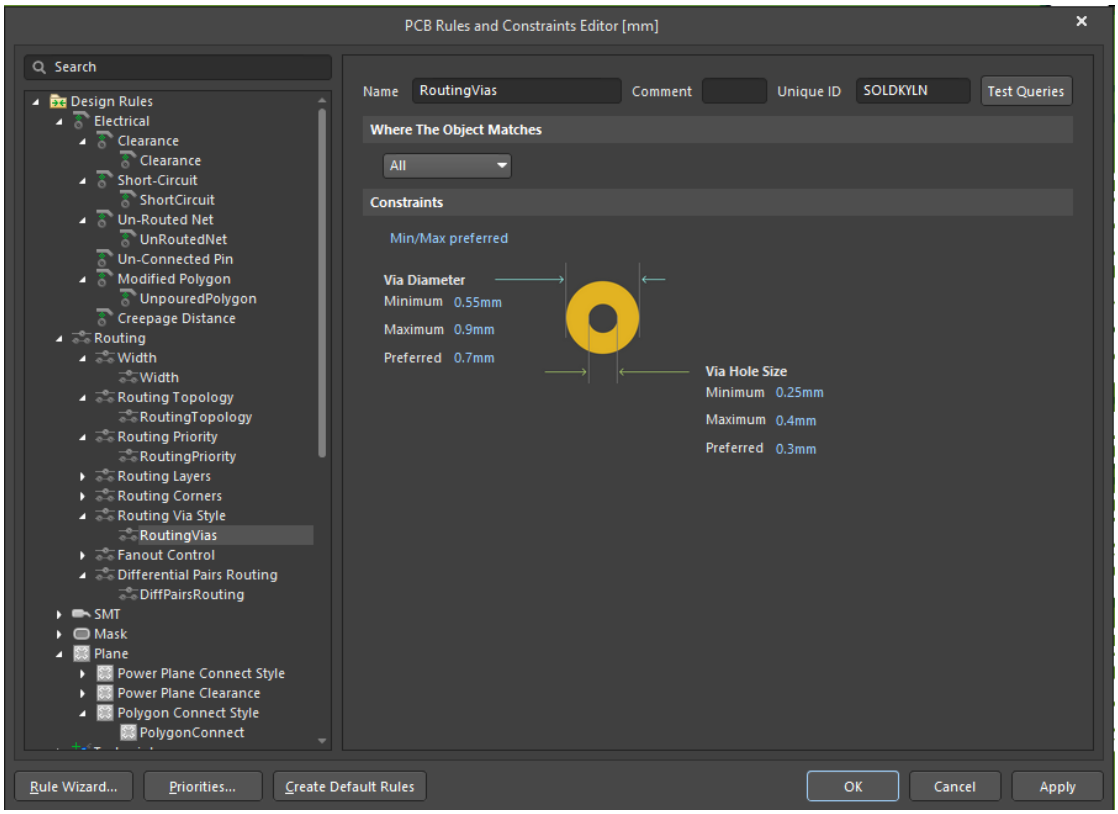
5. Dizajn i raspored tiskane pločice

Dizajniranje prototipa počinje stvaranjem novog PCB projekta unutar aktivnog Altium Designer „Workspacea“. Projektu se dodaje ime te se kreira shema projekta. Na praznu shemu se dodaju sve komponente potrebne za rad sklopovlja. Važno je uzeti u obzir vrstu paketa, izvedbu te vrijednost svake komponente. Ako unutar „Generic Library“ ne postoji komponenta koja je potrebna za izvedbu, AD omogućuje stvaranje prilagođenih komponenti unutar „custom library“. Prilikom stvaranja komponenti potrebno je voditi računa o njihovom blok dizajnu, odnosno „Footprint“ (otisku) te 3D modelu. Altium Designer na vrlo jednostavan način pruža korisniku pronalazak komponenti preko „Manufacturer Part Search“ odjeljka koji preko zajedničke on - line baze podataka prikazuje sve dostupne komponente s njihovim izvedbama te cijenom i dobavljačima.



Slika 5.1 Prikaz odjeljka za odabir komponenti preko on - line baze komponenti

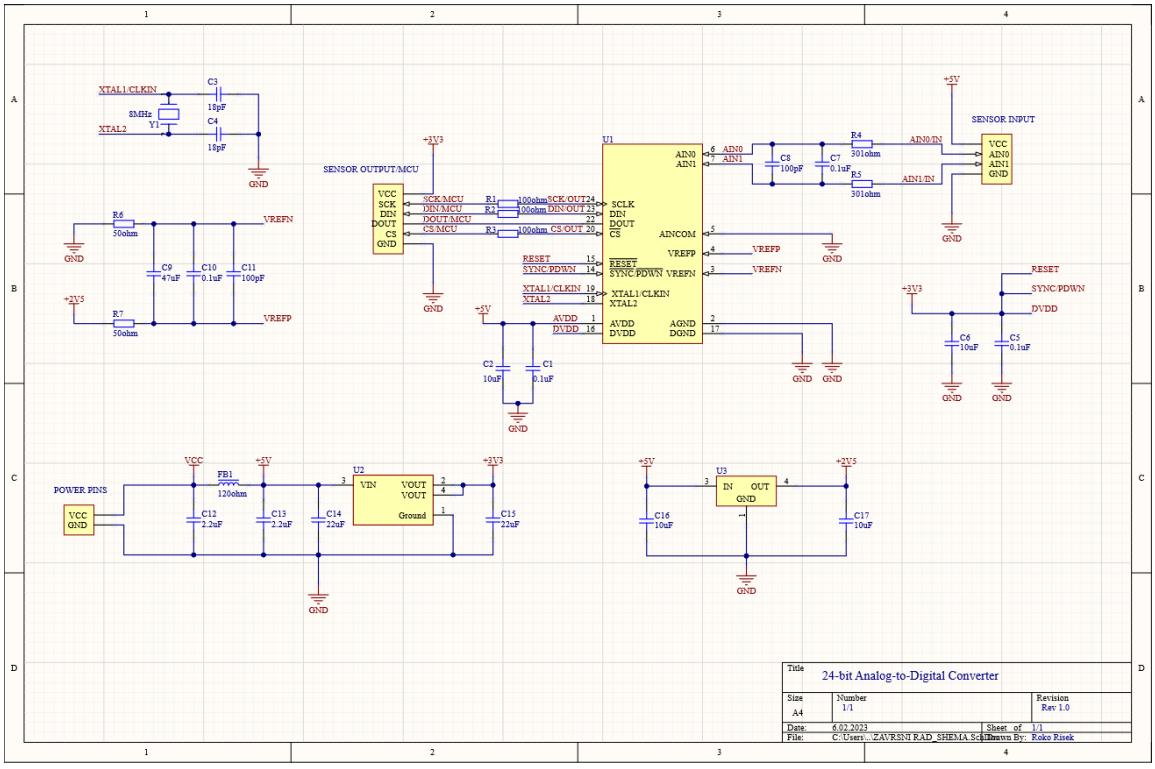
Nakon završetka projektiranja sheme istu je potrebno naknadno spremiti. Prije samog početka stvaranja rasporeda pločice, potrebno je odrediti poseban skup pravila prema kojima će se ta pločica proizvoditi. Unutar odjeljka „Rules“ definiraju se dopuštene minimalne i maksimalne udaljenosti između vodova, promjeri rupa za montiranje i proizvodnju, prespoja, debljine vodova i ostalih potrebnih detalja za ispravnu izradu. Također važno je i izabrati odgovarajući broj slojeva same pločice. To ovisi o samoj razini složenosti sheme te količini komponenata. Za manje složene sheme najčešće se koristi 2, 4 ili 6 slojne elektroničke pločice.



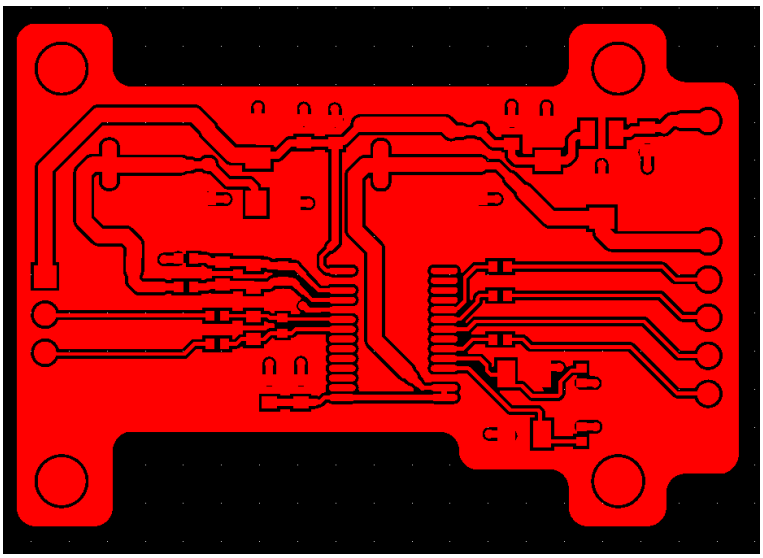
Slika 5.2 Prikaz odlomka za stvaranje pravila prilikom dizajniranja PCB Layout - a

Nakon završnog pregleda sheme potrebno je priložiti raspored svih komponenata na pločici. Unutar zadanih dimenzija postavljaju se komponente te se zatim povezuju. Povezivanje se vrši funkcijom „routing“ kojom stvaramo vodove koji fizički povezuju komponente. Za pregledniji rad ako je odabrana dvoslojna pločica potrebno je odrediti gornju ili donju stranu kao +V sekciju (pozitivni potencijal) i GND sekciju (područje nultog potencijala). Koristeći tu metodu drastično

se smanjuju potrebne dimenzije pločice te se povećava preglednost i smanjuju se EMS i SI. Kada je povezivanje završeno potrebno je još jednom provjeriti je li sve urađeno prema pravilima. Korištenjem opcije DRC, program javlja sva upozorenje i greške prema prije određenim pravilima. Vrlo je bitno naglasiti, da se sva upozorenje ili greške moraju otkloniti kako ne bi došlo do problema u proizvodnji ili ne mogućnosti izrade. Posljednji korak se odnosi na generiranje sve popratne dokumentacije za proizvodnju. Generiranjem datoteke „Gerber file“ stvaraju se sve potrebne informacije za izradu pločice. Datoteka sadrži mjerilo u kojem je pločica napravljena, dimenzije, slojeve, rupe, prespoje, vodove te pozicije svakog otiska komponente na pločici. Generiranjem datoteke „Bill of Materials“ ispisuju se sve komponente koje se nalaze na shemi. Uz komponente dolaze informacije i o samoj količini te broju ponavljanja istih. Stvaranjem „Pick and place“ ispisa dobivaju se točne koordinate na X i Y osi svih komponenti u zadanom mjerilu. Takav ispis se koristi kako bi u proizvodnji stroj mogao znati na koju poziciju postaviti koju komponentu. Generirajući „Gerber File“ datoteku stvara se prikaz svih slojeva tiskane pločice.

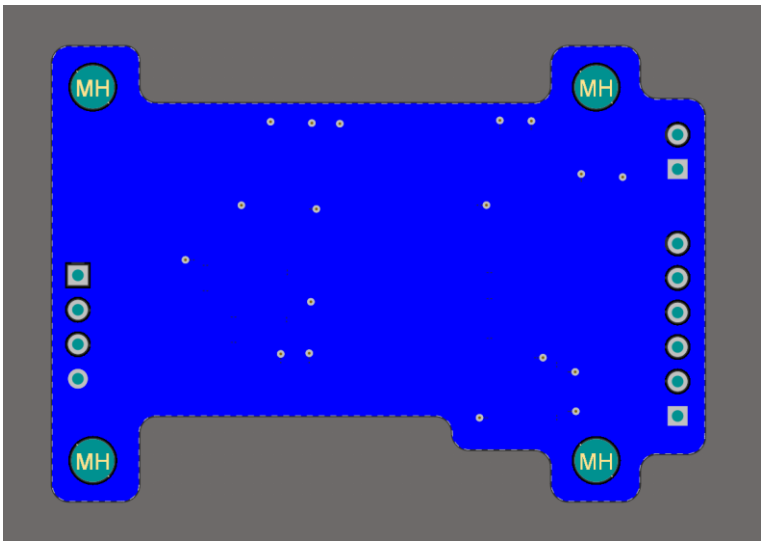


Slika 5.3 Shema senzora za mjerenje mase primjenom tenzometarske trake



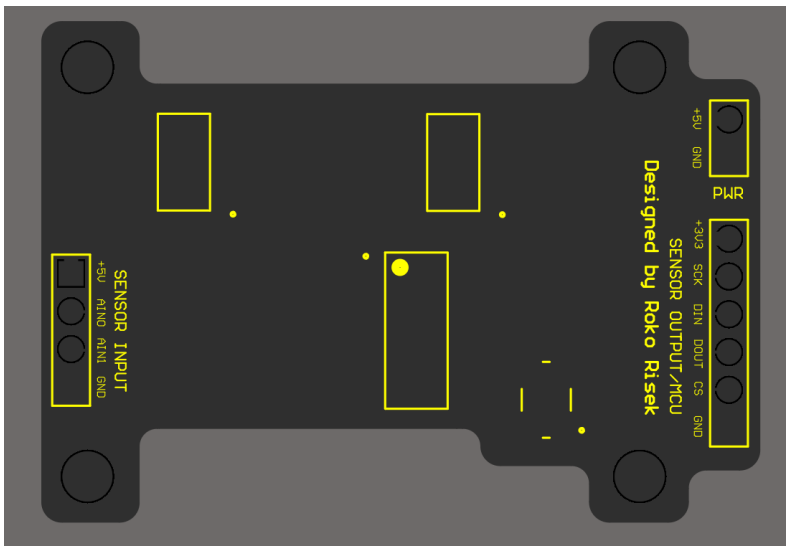
Slika 5.4 Prikaz veza i otiska komponenti na prednjoj strani tiskane pločice

Slika 5.4 prikazuje vodove i otiske komponenti s prednje strane tiskane pločice. Crveno označeno polje prikazuje „Poligon Pour“. Funkcija koja stvara odabrano područje na tiskanoj pločici te omogućuje korisniku vlastiti izbor potencijala. Smanjuje elektromagnetske smetnje i smetnje između analognog i digitalnog signala.



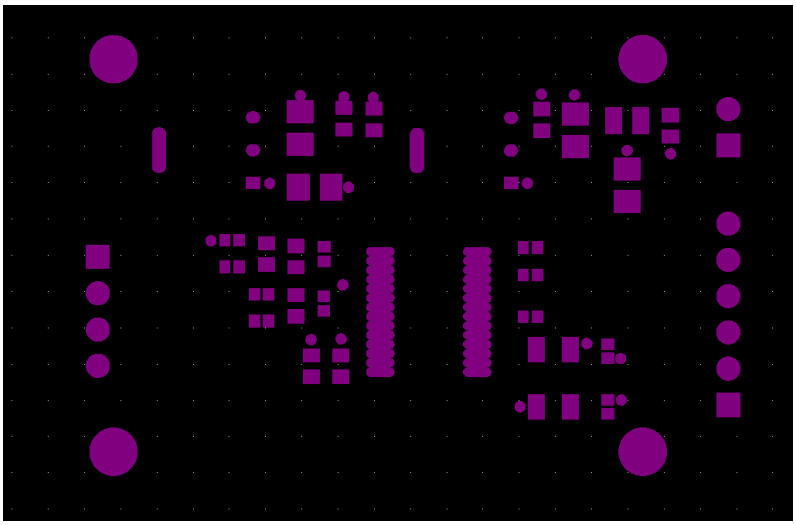
Slika 5.5 Prikaz veza na stražnjoj strani tiskane pločice

Slika 5.5 prikazuje prespojne vodove na stražnjoj strani tiskane pločice. Vodovi dolaze s komponenti koje se nalaze na prednjoj strani. Prespajanjem vodova GND s prednje strane na stražnju, postiže se efikasniji dizajn elektroničke tiskane pločice. Također se smanjuju elektromagnetske smetnje.



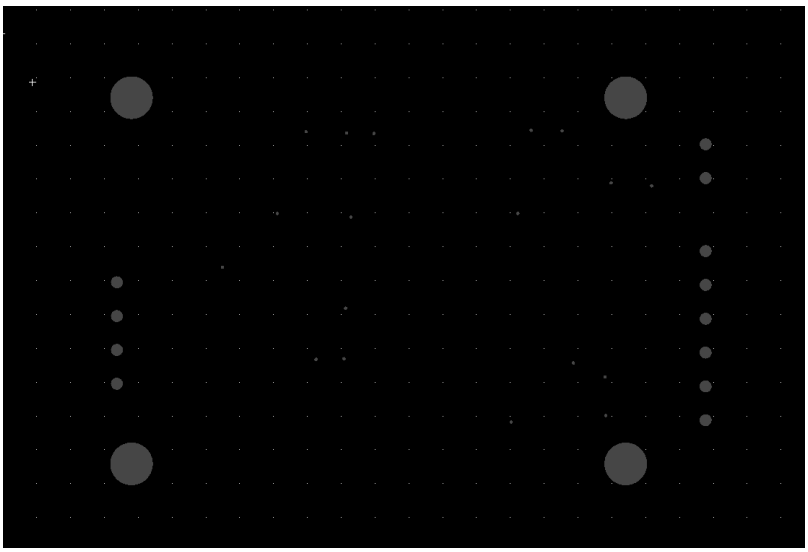
Slika 5.6 Prikaz sloja „Top Overlay“ na prednjoj strani tiskane pločice

Slika 5.6 prikazuje sloj printanih naziva na tiskanoj pločici. Koristeći sloj „Top Overlay“ omogućuje se vizualno pojasniti svrhu komponenti ili konektora na tiskanoj pločici. Smanjuje se mogućnost greške tijekom zamjene neispravne komponente ili korištenja krivog pina.



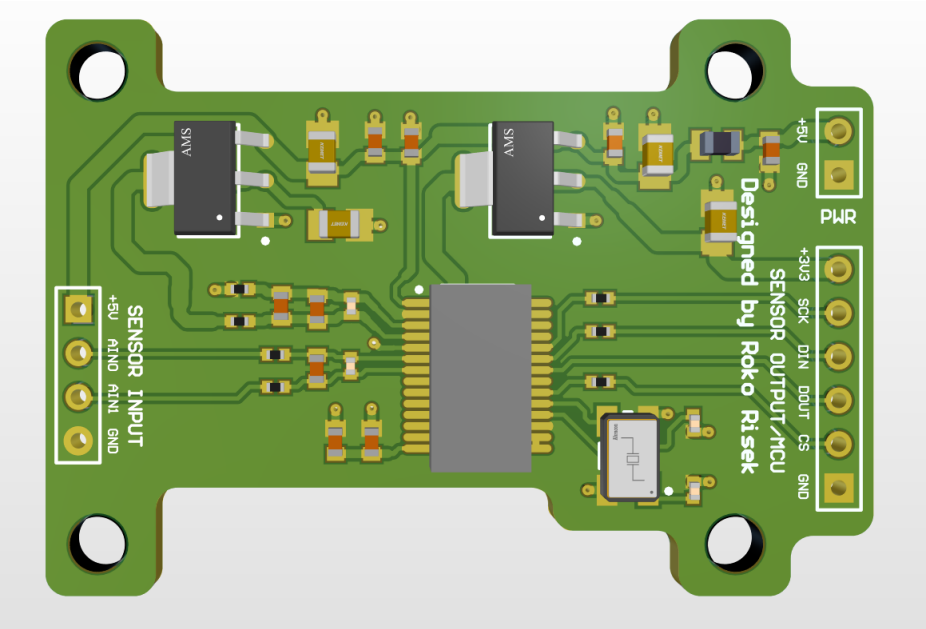
Slika 5.7 Prikaz sloja „Soldermask - a“ na prednjoj strani tiskane pločice

Slika 5.7 prikazuje sloj maske za lemljenje. Ljubičasta područja su tanak sloj polimera nalik laku koji se obično nanosi na bakrene vodove na tiskanoj pločici. Služi za zaštitu od oksidiranja bakrenih spojeva te stvaranja lemnih mostova između dva ili više usko raspoređenih bakrenih voda.



Slika 5.8 Prikaz pozicija montažnih rupa i prespoja prednje i stražnje strane tiskane pločice

Slika 5.8 prikazuje potrebne rupe na tiskanoj pločici. Rupe manjih dimenzija na slici, promjera 0.6 mm služe kao prespojni vodovi. Rupe većih dimenzija, promjera 3.2 mm služe kao montirajuće rupe za adekvatno kućište tiskane pločice.



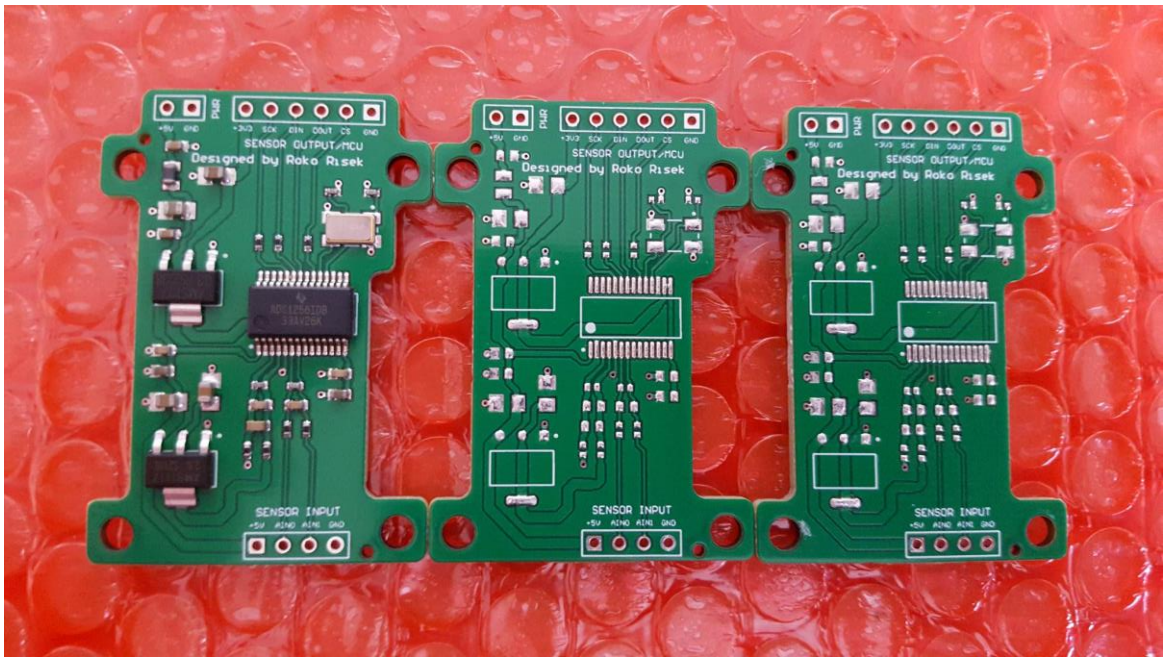
Slika 5.9 3D model završne verzije tiskane pločice

Unutar programskog alata postoji funkcija za 2D ili 3D pregled elektroničke pločice. Korisniku omogućuje jednostavniju vizualnu inspekciju svih komponenti na pločici te njihov redoslijed.

6. Proizvodnja tiskane pločice

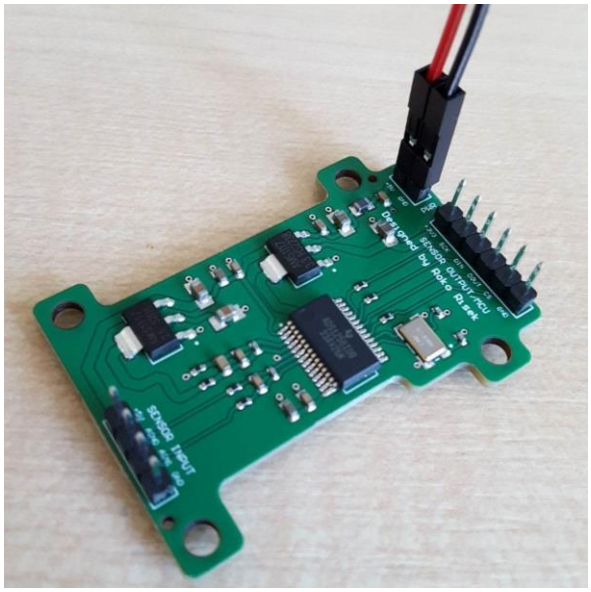
Proizvodnja tiskane pločice izvedena je u suradnji s JLC PCB tvrtkom. Sjedište tvrtke nalazi se u Kini. Duži niz godina predstavljaju vodeće lidere za proizvodnju tiskanih pločica. Sama proizvodnja se odvija u nekoliko etapa. Prvi postupak je rezanje FR - 4 ploča te bušenje montažnih rupa za postavljanje na stroj. Nakon bušenja dolazi na red taloženje bakra na samu ploču. Zatim se na ploče postavlja uzorak tiskane pločice sa svim vodovima i otiscima komponenti. Kasnije prolazi kroz završni pregled gdje radnici vizualno pregledavaju ploče u slučaju kakve greške ili nepravilnosti u nanošenju tinte. Na dijelu ploče koji nije otisnut uzorkom skida se bakreni sloj. Završni rezultat je tiskana pločica s istaknutim vodovima te se na nju kasnije dodaje „Solder Mask“ i „Silkscreen“. „Solder Mask“ je posebna vrsta električno ne vodljivog premaza koji štiti ostatak elektroničke pločice koji nema izvedene bakrene vodove za povezivanje od kratkog spoja i oksidacije. Postavljanje lemne paste radi se postupkom HASL. Ploče prolaze kroz automatiziranu traku koja upušta ploče u rastaljeni lem.

Završni proces čine vizualni i električni pregled. Radnici vizualno pregledavaju ploče te koriste automatizirane sonde za ispitivanje ispravnosti vodova. Ispravnost vodova se testira mjerenjem otpora i testom provodljivosti. Montaža tiskane pločice se odvija automatiziranim robotima. Radnici imaju ulogu postavljanja filmova s komponentama unutar spremnika stroja. Zatim robot dobivenom „Pick and Place“ datotekom postavlja komponente na predviđene pozicije. Nakon što je robot završio postavljane komponenta, ploče se zagrijavaju te spajaju komponente s pločom s pomoću lemne paste. Završni test odnosi se na ponovni vizualni i električni pregled te je zatim tiskana pločica spreman za dostavu. Sama dostava tiskane pločice protekla je unutar tjedan dana zahvaljujući brzoj usluzi poštarske službe.



Slika 6.1 Završna izvedba tiskanih pločica

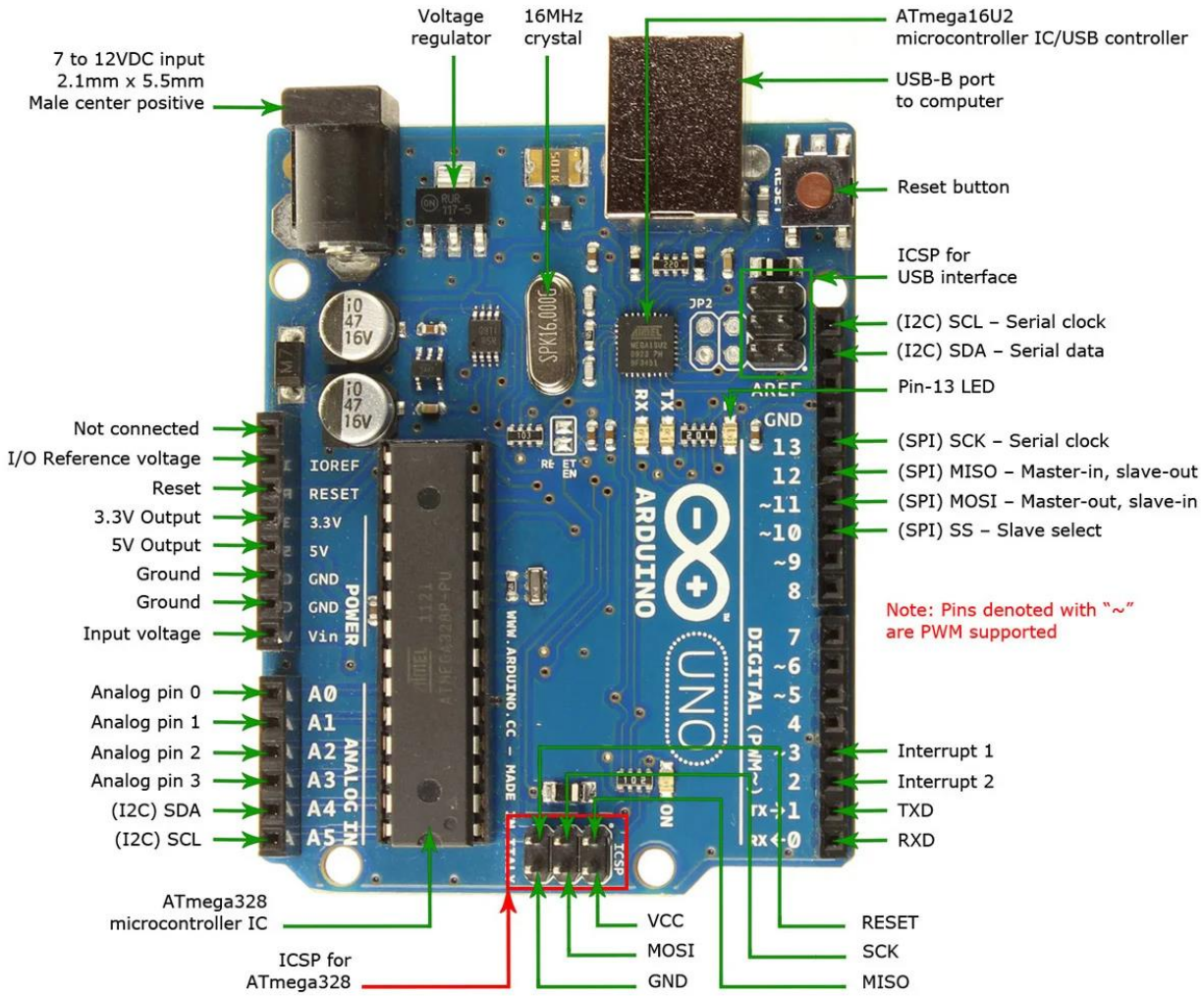
Nakon što su tiskane pločice stigle. Trebalo je postaviti THT komponente jer tvrtka ne nudi takve usluge. U ovom slučaju to si bili izlazni „header“ konektori.



Slika 6.2 Prikaz postavljenih „header“ konektora na tiskanoj pločici

7. Programski kod

Programski kod za mjerenje mase izveden je pomoću Arduino IDE sučelja. Povezivanjem senzora i Arduino mikrokontrolera omogućuje se upravljanje senzorom te daljnjom obradom signala (mjerenjem mase) putem jednog okruženja. Povezivanjem senzorskog konektora „Sensor Output“ s digitalnim pinovima na Arduino mikrokontroleru započinje se proces komunikacije. Otvaranjem Arduino IDE okruženja stvara se novi „Sketch“ profil unutar kojeg se upisuje programski kod. U zaglavlju se uključuju potrebni knjižnice za povezivanje i komunikaciju hardware-a sa software-om.



Slika 7.1 Prikaz Arduino Uno mikrokontrolera [7]

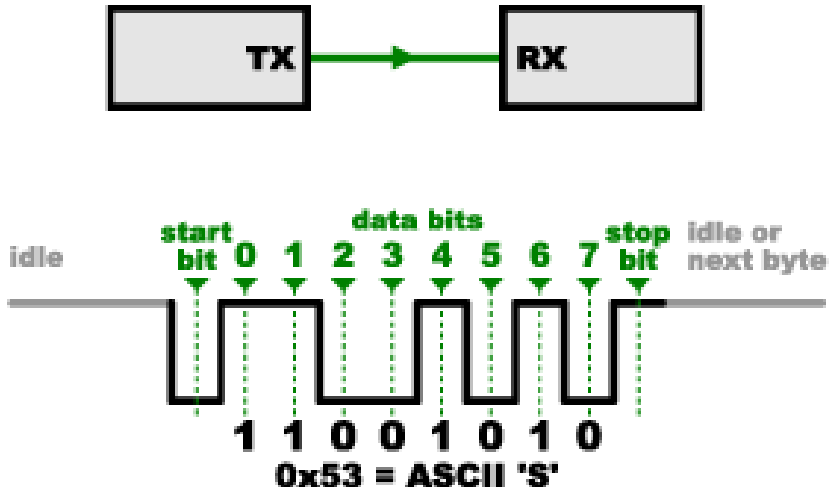
Slika 7.1 prikazuje raspodjelu i značenje svih konektora na Arduino Uno mikrokontroleru.

7.1. Serijska komunikacija (SPI)

Povezivanje samog senzora s Arduino Uno mikrokontrolerom omogućeno je putem SPI komunikacije. SPI je sabirnica serijskog sučelja koja se najčešće koristi za slanje podatak između mikrokontrolera i malih periferija kao što su razni senzori, memorijske kartice i registri pomaka. Koristi zasebne linije za podešavanje takta i podatka sa linijom za odabir uređaja s kojim komunicira.

7.2. Asinkrona komunikacija

Uobičajeni serijski port, s TX i RX linijama, naziva se asinkroni jer nema kontrole nad podacima kada se šalju, niti jamstva da obje strane rade točno istom brzinom. Budući da se računala obično oslanjaju na to da je sve sinkronizirano istim taktom, to može biti problem kada dva sustava s malo različitim taktom pokušavaju komunicirati jedan s drugim [8]. Kako bi se zaobišao ovaj problem, asinkrone serijske veze dodaju dodatne početne i zaustavne bitove svakom bajtu koji pomažu prijemniku da se sinkronizira s podacima kada stignu. Obje strane također se moraju unaprijed dogovoriti o brzini prijenosa (npr.115200 bita u sekundi). Male razlike u brzini prijenosa nisu problem jer se prijemnik ponovno sinkronizira na početku svakog bajta [8].

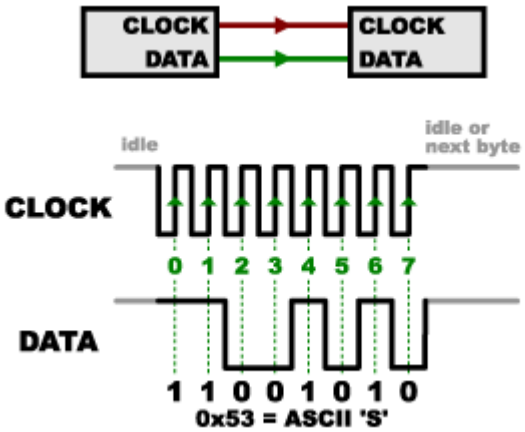


Slika 7.2 Prikaz asinkronog serijskog prijenosa podataka [8]

Asinkroni serijski prijenos ima puno opterećenja u dodatnim početnim i zaustavnim bitovima koji se šalju sa svakim bajtom. Također ovisi o složenom hardveru potrebnom za slanje i primanje podataka. Primjenom takvog prijenosa dolaze greške ako obje strane za komunikaciju nemaju podešenu istu brzinu prijenosa podataka. Primljeni podaci bit će neiskoristivi. To nastaje zato što prijemnik obrađuje bitove u vrlo određeno vrijeme. Ako prijemnik gleda pogrešno vrijeme, vidjet će pogrešne bitove [8]. U slici 7.2 prikazan je takav prijenos.

7.3. Sinkrona komunikacija

SPI je sinkrona podatkovna sabirnica, što znači da koristi zasebne linije za podatke i određivanje takta koji održava prijamnik i predajnik u sinkronizaciji. Takt je oscilirajući signal koji govori prijamniku točno kada treba uzrokovati bitove na podatkovnoj liniji. To bi mogao biti uzlazni (od niskog prema visokom) ili opadajući (od visokog prema niskom) rub signala takta. Kada prijemnik otkrije taj rub, odmah će pregledati podatkovnu liniju kako bi pročitao sljedeći bit. Budući da se takt šalje s podacima, određivanje brzine nije važno, iako će uređaji imati najveću brzinu pri kojoj mogu raditi.

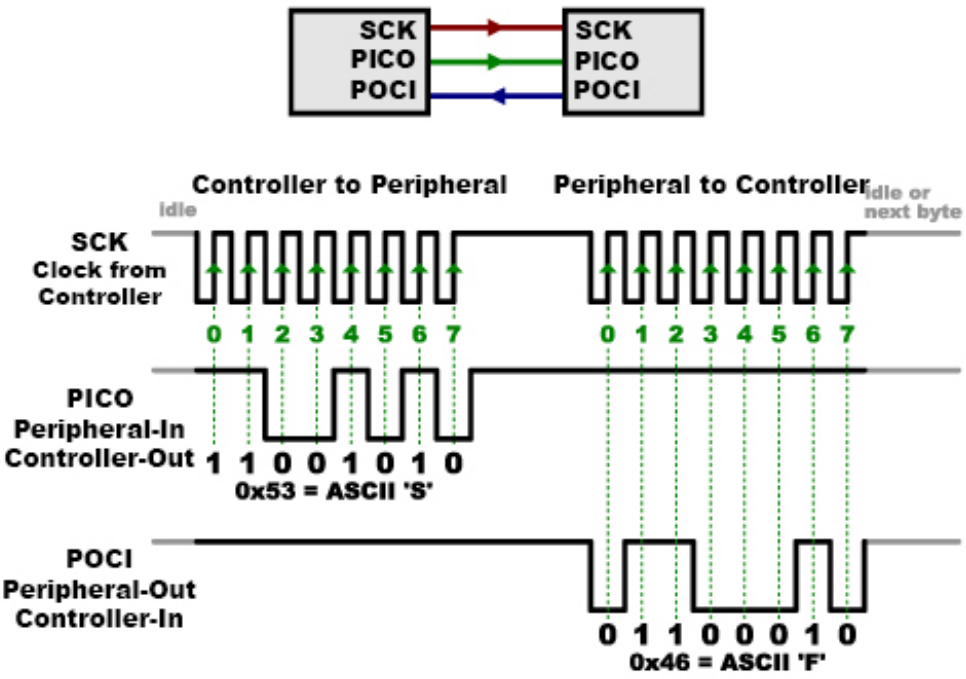


Slika 7.3 Prikaz sinkronog serijskog prijenosa podataka [8]

Jedan od razloga zašto je SPI toliko popularan je taj što prijemni hardver može biti jednostavan registar pomaka. Ovo je puno jednostavniji i jeftiniji dio hardware - a od kompletnog UART - a.

7.4. Prijenos podataka

Unutar SPI komunikacije samo jedna strana generira signal takta (najčešće se naziva CLK ili SCK) te predstavlja „Serial Clock“. Strana koja generira naziva se kontroler, a druga strana se naziva periferija. Uvijek postoji samo jedan kontroler koji je najčešće mikrokontroler, ali može biti više perifernih uređaja. Kada se podaci šalju s kontrolera na periferni uređaj, šalju se putem podatkovne linije koja se zove PICO, te predstavlja „Peripheral In / Controller Out“. Ako periferija treba poslati odgovor natrag kontroleru, kontroler će nastaviti generirati unaprijed dogovoreni broj ciklusa takta, a periferija će staviti podatke u treću podatkovnu liniju nazvanu POCI, odnosno „Peripheral Out / Controller In“.

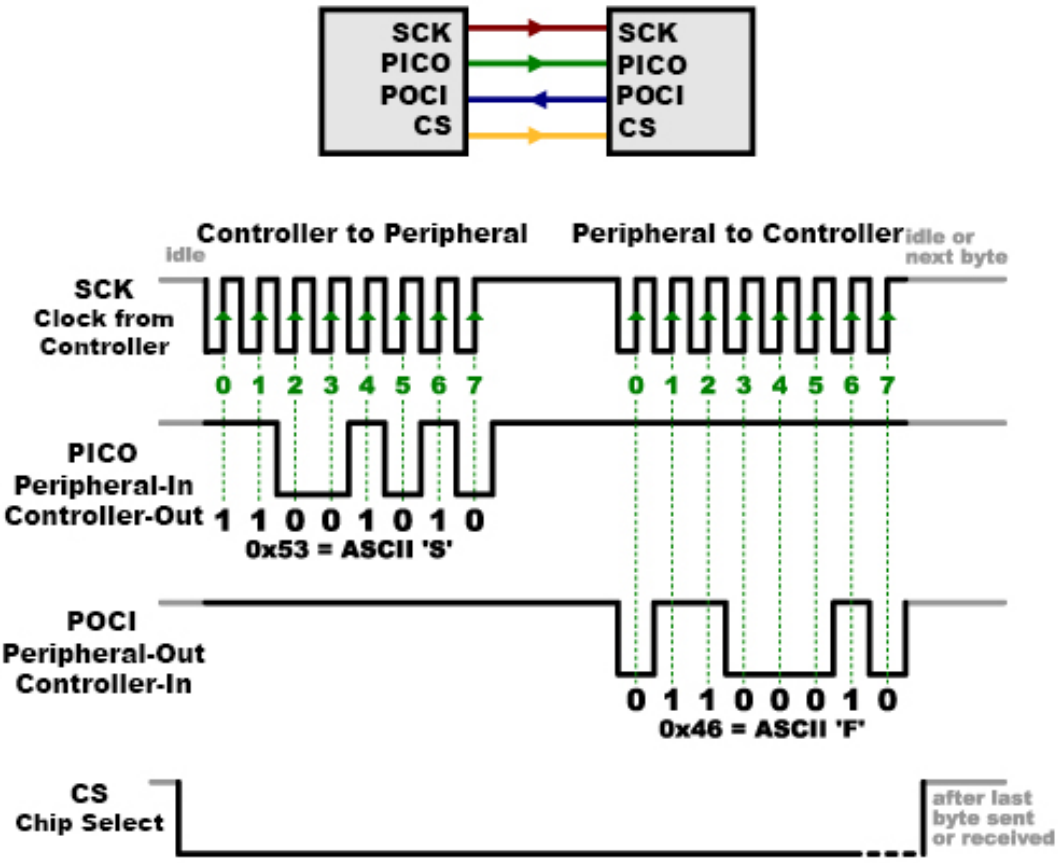


Slika 7.4 Prikaz povezivanja kontrolera i periferije sinkronom vezom [8]

Budući da kontroler uvijek generira signal sata, mora unaprijed znati kada periferija treba vratiti podatke i koliko će podataka biti vraćeno. Ovo je vrlo različito od asinkronog serijskog, gdje se nasumične količine podataka mogu slati u bilo kojem smjeru u bilo kojem trenutku. U praksi to nije problem, jer se SPI općenito koristi za komunikaciju sa senzorima koji imaju vrlo specifičnu strukturu naredbi. Na primjer, ako se nekom uređaju pošalje naredba za "čitanje podataka", uređaj će uvijek poslati npr. dva bajta zauzvrat.

7.5. Chip Select (CS) veza

Veza koja je odgovorna za početak primanja ili slanja podataka, a također se koristi kada je prisutno više periferija za odabir one s kojom želi komunicirati.



Slika 7.5 Prikaz čitanja i pisanja informacija SPI komunikacijom te logičko stanje CS linije [8]

CS linija se obično drži visoko, što odspaja periferiju od SPI. Takva vrsta logike poznata je kao aktivna niska razina, a često se koristi za omogućavanje i resetiranje veza. Neposredno prije nego što se podaci pošalju periferiji, linija se spušta na nisku razinu, što aktivira periferiju. Kada se završi prijenos, linija se ponovno postavlja visoko [8].

7.6. Stvaranje library datoteke senzora za povezivanje sa mikrokontrolerom

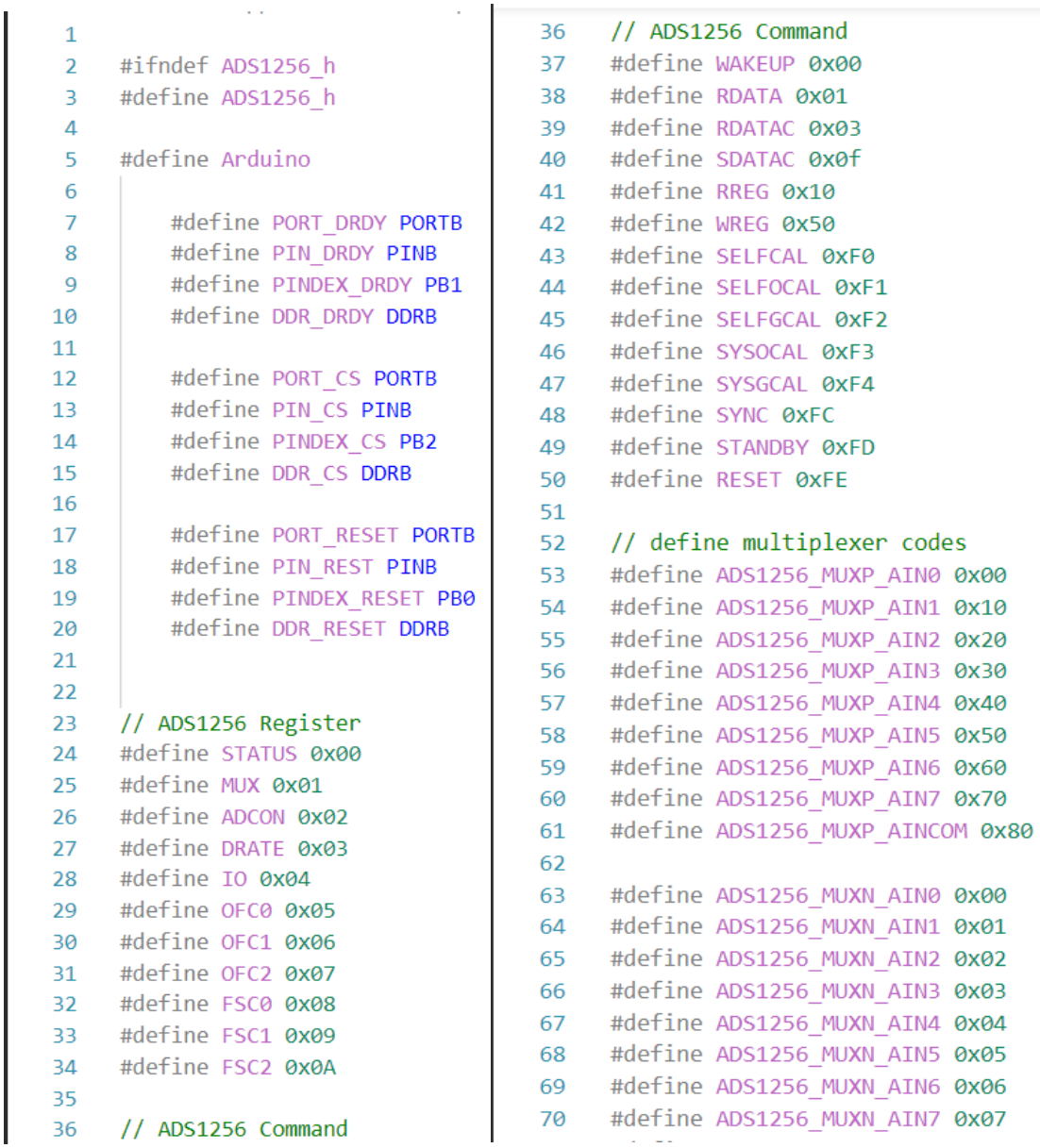
Kako bi sam senzor mogao komunicirati s Arduino mikrokontrolerom potrebno je napisati knjižnicu (library).

Knjižnica se sastoji od nekoliko glavnih dijelova:

- Header file (.h) - vrsta datoteke gdje su ispisane sve definicije knjižnice.
- Source code (.cpp) - vrsta datoteke koja sadrži izvorni program senzora.
- Keyword file (.txt) – datoteka unutar koje se povezuju sve ključne riječi korištene u programskom kodu.
- Readme file (.txt) – datoteka u kojoj se nalaze ostale bitne informacije o samoj knjižnici(npr. način povezivanja, otklanjanje mogućih nastalih grešaka itd.).
- Example file (.ino) – datoteka u kojoj se nalazi kod za primjer (npr. opća primjena).

7.6.1. Header file

Datoteka zaglavlja je datoteka s ekstenzijom .h koja sadrži deklaracije C/C++ funkcija i definicije makronaredbi koje se dijele između nekoliko izvornih datoteka. Korištenje zaglavlja unutar programa vrši se direktivom #include. Jednostavna praksa u C ili C++ programima je da sadrže konstante, makronaredbe, globalne varijable i prototipe funkcija u datotekama zaglavlja te se po potrebi uključuju. Takve vrste datoteka vrše važnu funkciju u samom programiranju jer se kroz njih definiraju jedni od najbitnijih parametara za kasniju obradu.



```
#ifndef ADS1256_h
#define ADS1256_h

#define Arduino

    #define PORT_DRDY PORTB
    #define PIN_DRDY PINB
    #define PINDEX_DRDY PB1
    #define DDR_DRDY DDRB

    #define PORT_CS PORTB
    #define PIN_CS PINB
    #define PINDEX_CS PB2
    #define DDR_CS DDRB

    #define PORT_RESET PORTB
    #define PIN_REST PINB
    #define PINDEX_RESET PB0
    #define DDR_RESET DDRB


// ADS1256 Register
#define STATUS 0x00
#define MUX 0x01
#define ADCON 0x02
#define DRATE 0x03
#define IO 0x04
#define OFC0 0x05
#define OFC1 0x06
#define OFC2 0x07
#define FSC0 0x08
#define FSC1 0x09
#define FSC2 0x0A

// ADS1256 Command
```

```
// ADS1256 Command
#define WAKEUP 0x00
#define RDATA 0x01
#define RDATAC 0x03
#define SDATAC 0x0f
#define RREG 0x10
#define WREG 0x50
#define SELFCAL 0xF0
#define SELFOCAL 0xF1
#define SELFGCAL 0xF2
#define SYSOCAL 0xF3
#define SYSGCAL 0xF4
#define SYNC 0xFC
#define STANDBY 0xFD
#define RESET 0xFE

// define multiplexer codes
#define ADS1256_MUXP_AIN0 0x00
#define ADS1256_MUXP_AIN1 0x10
#define ADS1256_MUXP_AIN2 0x20
#define ADS1256_MUXP_AIN3 0x30
#define ADS1256_MUXP_AIN4 0x40
#define ADS1256_MUXP_AIN5 0x50
#define ADS1256_MUXP_AIN6 0x60
#define ADS1256_MUXP_AIN7 0x70
#define ADS1256_MUXP_AINCOM 0x80

#define ADS1256_MUXN_AIN0 0x00
#define ADS1256_MUXN_AIN1 0x01
#define ADS1256_MUXN_AIN2 0x02
#define ADS1256_MUXN_AIN3 0x03
#define ADS1256_MUXN_AIN4 0x04
#define ADS1256_MUXN_AIN5 0x05
#define ADS1256_MUXN_AIN6 0x06
#define ADS1256_MUXN_AIN7 0x07
```

Slika 7.6 Prikaz .h programskog djela koda – 1.dio [9]

Kod pisanja header datoteke bitno je na samom početku definirati ime glavnog upravljačkog sklopa te definirati mikrokontroler putem kojeg će komunicirati. Potrebno je također definirati sve register te njihova stanja za izlazne pinove iz senzora. Postavljanje početnog stanja registra odvija se prema naputcima izvedenih iz podatkovne tablice ADS 1256 sklopa.

Glavni programski dijelovi su:

- definiranje registara
- komandnih funkcija
- svih 8 ulaznih kanala multipleksora

```
71  #define ADS1256_MUXN_AINCOM 0x08
72
73  // define gain codes
74  #define ADS1256_GAIN_1 0x00
75  #define ADS1256_GAIN_2 0x01
76  #define ADS1256_GAIN_4 0x02
77  #define ADS1256_GAIN_8 0x03
78  #define ADS1256_GAIN_16 0x04
79  #define ADS1256_GAIN_32 0x05
80  #define ADS1256_GAIN_64 0x06
81
82  // define drate codes
83  #define ADS1256_DRATE_30000SPS 0xF0
84  #define ADS1256_DRATE_15000SPS 0xE0
85  #define ADS1256_DRATE_7500SPS 0xD0
86  #define ADS1256_DRATE_3750SPS 0xC0
87  #define ADS1256_DRATE_2000SPS 0xB0
88  #define ADS1256_DRATE_1000SPS 0xA1
89  #define ADS1256_DRATE_500SPS 0x92
90  #define ADS1256_DRATE_100SPS 0x82
91  #define ADS1256_DRATE_60SPS 0x72
92  #define ADS1256_DRATE_50SPS 0x63
93  #define ADS1256_DRATE_30SPS 0x53
94  #define ADS1256_DRATE_25SPS 0x43
95  #define ADS1256_DRATE_15SPS 0x33
96  #define ADS1256_DRATE_10SPS 0x23
97  #define ADS1256_DRATE_5SPS 0x13
98  #define ADS1256_DRATE_2_5SPS 0x03
```

Slika 7.7 Prikaz .h programskog djela koda – 2. dio [9]

U nastavku header datoteke definiraju se stanja registra za opcije pojačanja pojačala. Pojačalo ima funkciju pojačanja signala za indeks 1, 2, 4, 8, 16, 32 i 64. Također bitno je definirati brzinu izlazne stope signala, čija maksimalna vrijednost iznosi 30 kSPS.

```
100 #include "Arduino.h"
101 #include "SPI.h"
102
103 class ADS1256 {
104  public:
105   ADS1256(float clockspdMhz, float vref, bool useresetpin);
106   void writeRegister(unsigned char reg, unsigned char wdata);
107   unsigned char readRegister(unsigned char reg);
108   void sendCommand(unsigned char cmd);
109   float readCurrentChannel();
110   void setConversionFactor(float val);
111   void setChannel(byte channel);
112   void setChannel(byte AIP, byte AIN);
113   void begin(unsigned char drate, unsigned char gain, bool bufferenable);
114   void waitDRDY();
115   void setGain(uint8_t gain);
116   void readTest();
117
118  private:
119   void CSON();
120   void CSOFF();
121   unsigned long read_uint24();
122   long read_int32();
123   float read_float32();
124   byte _pga;
125   float _VREF;
126   float _conversionFactor;
127 };
128
129 #endif
```

Slika 7.8 Prikaz .h programskog djela koda – 3. dio [9]

Završni dio header datoteke sadrži predložak (klasa) za stvaranje objekta. Sadrži se od varijabli i funkcija. Unutar klase se definiraju glavni podaci za postavljanje senzora za mjerenje. Klasa se dijeli na privatnu i javnu. Unutar javne klase dopušta se izravan pristup bilo kojem kodu unutar elementa klase. Definira se brzina obrade podataka, odabir ulaznih kanala, pojačanje itd. Dok kod privatne klase pristup je ograničen samo za klasu, odnosno pristup je ograničen samo metodama koje su dio iste klase.

Unutar privatne klase se definira uključivanje:

- CS veze
- podatkovni tipovi za inicijalizaciju
- vrijednost referentnog napona

7.6.2. Source file

Source file je vrsta datoteke koja sadrži izvorne podatke za metodu klase. Koristi se za odvajanje sučelja i metoda. Source datoteka se implementira u samom zaglavlju programa te može uključivati razne knjižnice koje će se koristiti od bilo koje deklarirane metode.

```
#include "ADS1256.h" //uključivanje ADS 1256 header datoteke
#include "Arduino.h" //uključivanje Arduino knjižnice
#include "SPI.h" //uključivanje SPI komunikacije

ADS1256::ADS1256(float clockspdMhz, float vref, bool useResetPin) {
  // Postavljanje DRDY kao ulaz
  DDR_DRDY &= ~(1 << PINDEX_DRDY);
  // Postavljanje CS kao izlaz
  DDR_CS |= (1 << PINDEX_CS);

  if (useResetPin) {
    // Postavljanje RESETPIN kao izlaz
    DDR_RESET |= (1 << PINDEX_RESET);
    // Postavljanje RESETPIN signala u visoko stanje
    PORT_RESET |= (1 << PINDEX_RESET);
  }

  //Referentni napon
  _VREF = vref;

  // Default faktor pretvaranja
  _conversionFactor = 1.0;

  // Start SPI komunikacije, 1/4 brzine ADC-a
  SPI.begin();
  SPI.beginTransaction(
      SPISettings(clockspdMhz * 1000000 / 4, MSBFIRST, SPI_MODE1));
}

void ADS1256::writeRegister(unsigned char reg, unsigned char wdata) {
  CSON();
  SPI.transfer(WREG | reg);
  SPI.transfer(0);
  SPI.transfer(wdata);
  __builtin_avr_delay_cycles(8); //8MHz takt
  CSOFF();
```

Slika 7.9 Prikaz .c++ programskog djela koda – 1. dio [9]

U prvom dijelu source datoteke uključuju se header datoteke senzora, serijske komunikacije i mikrokontrolera. Potom se definira tip podataka za brzinu obrade signala, referentni napon i izlazni pin za resetiranje. Određuje se brzina prijenosa podataka preko SPI komunikacije te njegova „WakeUp“ funkcija. Funkcija buđenja služi da komunikacija počinje raditi kada mu na ulazu kanala dolazi signal (visoko, srednja ili niska pobuda).

```
37  }
38
39  unsigned char ADS1256::readRegister(unsigned char reg) {
40    unsigned char readValue;
41
42    CSON();
43    SPI.transfer(RREG | reg);
44    SPI.transfer(0);
45    __builtin_avr_delay_cycles(200);
46    readValue = SPI.transfer(0);
47    __builtin_avr_delay_cycles(8);  // t11 kašnjenje
48    CSOFF();
49
50    return readValue;
51  }
52
53  void ADS1256::sendCommand(unsigned char reg) {
54    CSON();
55    waitDRDY();
56    SPI.transfer(reg);
57    __builtin_avr_delay_cycles(8);  // t11
58    CSOFF();
59  }
60
61  void ADS1256::setConversionFactor(float val) { _conversionFactor = val;
62  }
63  void ADS1256::readTest() {
64    unsigned char _highByte, _midByte, _lowByte;
65    CSON();
66    SPI.transfer(RDATA);
67    __builtin_avr_delay_cycles(200);  // t6 kašnjenje
68
69    _highByte = SPI.transfer(WAKEUP);
70    _midByte = SPI.transfer(WAKEUP);
71    _lowByte = SPI.transfer(WAKEUP);
72
```

Slika 7.10 Prikaz .c++ programskog djela koda – 2. dio [9]

```
73    CSOFF();
74  }
75
76  float ADS1256::readCurrentChannel() {
77    CSON();
78    SPI.transfer(RDATA);
79    __builtin_avr_delay_cycles(200);  // t6 kašnjenje
80    float adsCode = read_float32();
81    CSOFF();
82    return ((adsCode / 0x7FFFFF) * ((2 * _VREF) / (float)_pga)) *
83                _conversionFactor;
84  }
85
86  // Pozivanje tek nakon što je RDATA funkcija izvršena
87  unsigned long ADS1256::read_uint24() {
88    unsigned char _highByte, _midByte, _lowByte;
89    unsigned long value;
90
91    _highByte = SPI.transfer(WAKEUP);
92    _midByte = SPI.transfer(WAKEUP);
93    _lowByte = SPI.transfer(WAKEUP);
94
95    //Pretvori sve 3-bajtne u 24-bitne podatke pomoću pomaka bajtova.
96    value = ((long)_highByte << 16) + ((long)_midByte << 8) + ((long)_lowByte);
97    return value;
98  }
99
100 // Pozivanje tek nakon što je RDATA funkcija izvršena
101 long ADS1256::read_int32() {
102   long value = read_uint24();
103
104   if (value & 0x00800000) {
105     value |= 0xff000000;
106   }
107
108   return value;
```

Slika 7.11 Prikaz .c++ programskog djela koda – 3. dio [9]

```
109  }
110
111  // Pozivanje tek nakon što je RDATA funkcija izvršena
112  float ADS1256::read_float32() {
113    long value = read_int32();
114    return (float)value;
115  }
116
117  // Prebacivanje kanala za jednostruki način rada
118  //Negativni ulazni kanal su automatski postavljen na AINCOM
119  void ADS1256::setChannel(byte channel) { setChannel(channel, -1); }
120
121  //Promjena kanala za diferencijalni način rada.
122  //Koristite -1 za postavljanje ulaznog kanala na AINCOM
123  void ADS1256::setChannel(byte AIN_P, byte AIN_N) {
124    unsigned char MUX_CHANNEL;
125    unsigned char MUXP;
126    unsigned char MUXN;
127
128    switch (AIN_P) {
129      case 0:
130        MUXP = ADS1256_MUXP_AIN0;
131        break;
132      case 1:
133        MUXP = ADS1256_MUXP_AIN1;
134        break;
135      case 2:
136        MUXP = ADS1256_MUXP_AIN2;
137        break;
138      case 3:
139        MUXP = ADS1256_MUXP_AIN3;
140        break;
141      case 4:
142        MUXP = ADS1256_MUXP_AIN4;
143        break;
144      case 5:
145        MUXP = ADS1256_MUXP_AIN5;
146        break;
147      case 6:
148        MUXP = ADS1256_MUXP_AIN6;
149        break;
150      case 7:
151        MUXP = ADS1256_MUXP_AIN7;
152        break;
153      default:
154        MUXP = ADS1256_MUXP_AINCOM;
155    }
156
157    switch (AIN_N) {
158      case 0:
159        MUXN = ADS1256_MUXN_AIN0;
160        break;
161      case 1:
162        MUXN = ADS1256_MUXN_AIN1;
163        break;
164      case 2:
165        MUXN = ADS1256_MUXN_AIN2;
166        break;
167      case 3:
168        MUXN = ADS1256_MUXN_AIN3;
169        break;
170      case 4:
171        MUXN = ADS1256_MUXN_AIN4;
172        break;
173      case 5:
174        MUXN = ADS1256_MUXN_AIN5;
175        break;
176      case 6:
177        MUXN = ADS1256_MUXN_AIN6;
178        break;
179      case 7:
180        MUXN = ADS1256_MUXN_AIN7;
```

Slika 7.12 Prikaz .c++ programskog djela koda – 4. dio [9]

```
      break;
    default:
      MUXN = ADS1256_MUXN_AINCOM;
  }

  MUX_CHANNEL = MUXP | MUXN;

  CSON();
  writeRegister(MUX, MUX_CHANNEL);
  sendCommand(SYNC);
  sendCommand(WAKEUP);
  CSOFF();
}

void ADS1256::begin(unsigned char drate, unsigned char gain, bool buffenable) {
  _pga = 1 << gain;
  sendCommand(
      SDATAC);  //pošalji SDATAC naredbu za zaustavljanje kontinuiranog načina čitanja.
  writeRegister(DRATE, drate);  // ispuni DATA RATE registar
  uint8_t bytemask = B00000111;
  uint8_t adcon = readRegister(ADCON);
  uint8_t byte2send = (adcon & ~bytemask) | gain;
  writeRegister(ADCON, byte2send);
  if (buffenable) {
    uint8_t status = readRegister(STATUS);
    bitSet(status, 1);
    writeRegister(STATUS, status);
  }
  sendCommand(SELFCAL);  // početak automatske kalibracije
  waitDRDY();
  ;  // pričekaj, nakon automatske kalibracije
}

void ADS1256::CSON() {
  PORT_CS &= ~(1 << PINDEX_CS);
}  // digitalWrite(_CS, nisko stanje); }
```

Slika 7.13 Prikaz .c++ programskog djela koda – 5. dio [9]

7.7. Povezivanje senzora sa mikrokontrolerom

Senzor se povezuje putem SPI sučelja. Sam senzor ima izvedene izlazne pinove za komunikaciju te se povezuje na također izvedene pinove na Arduino mikrokontroleru. Nakon uključenja knjižnice senzora unutar Arduino IDE okruženja, moguće je povezati i provjeriti rad senzora. Završna faza pokretanja senzora je implementacija glavnog izvršnog koda. Sam kod provjerava promjenu unutar već zadanih funkcija i parametra memorije ADS 1256 sklopa, te njegove promjene šalje glavnom programu koji dalje obrađuje podatke. Obrađeni podaci (mjerena masa) prikazuje se na serijskom prikazivaču unutar okruženja. Te je tako vrlo jednostavno vidjeti ispravnost senzora.

ZAVRSNI_RAD_KOD_20231113.ino

```
#include <ADS1256.h>
#include <EEPROM.h>
#if defined(ESP8266)|| defined(ESP32)

const int ADS1256_dout = 5; //dout pin
const int ADS1256_sck = 4; //sck pin

//Elektronički podsustav(Senzor):
ADS1256_ADC LoadCell(ADS1256_dout, ADS1256_sck);

const int calVal_eepromAdress = 0;
unsigned long t = 0;

void setup() {
  Serial.begin(57600); delay(10);
  Serial.println();
  Serial.println("Pokretanje...");

  LoadCell.begin();
  unsigned long stabilizingtime = 2000;
  boolean _tare = true;
  LoadCell.start(stabilizingtime, _tare);
  if (LoadCell.getTareTimeoutFlag() || LoadCell.getSignalTimeoutFlag()) {
    Serial.println("Proteklo vrijeme, provjeri izlazne pinove sa elektroničkog podsustava");
    while (1);
  }
  else {
    LoadCell.setCalFactor(1.0);
    Serial.println("Učitavanje izvršeno");
  }
  while (!LoadCell.update());
  calibrate();
}
```

Slika 7.14 Prikaz programskog koda mikrokontrolera – 1. dio [9]

ZAVRSNI_RAD_KOD_20231113.ino

```
33  }
34
35  void loop() {
36    static boolean newDataReady = 0;
37    const int serialPrintInterval = 0;
38
39
40    if (LoadCell.update()) newDataReady = true;
41
42
43    if (newDataReady) {
44      if (millis() > t + serialPrintInterval) {
45        float i = LoadCell.getData();
46        Serial.print("Vrijednost tenzometarske trake: ");
47        Serial.println(i);
48        newDataReady = 0;
49        t = millis();
50      }
51    }
52
53      if (Serial.available() > 0) {
54      char inByte = Serial.read();
55      if (inByte == 't') LoadCell.tareNoDelay();
56      else if (inByte == 'r') calibrate();
57      else if (inByte == 'c') changeSavedCalFactor();
58    }
59    if (LoadCell.getTareStatus() == true) {
60      Serial.println("Mjerenje mase neopterećenog objekta izvršeno");
61    }
62
63  }
64
```

Slika 7.15 Prikaz programskog koda mikrokontrolera – 2. dio [9]

ZAVRSNI_RAD_KOD_20231113.ino

```
65  void calibrate() {
66    Serial.println("***");
67    Serial.println("Početak inicijalizacije senzora :");
68    Serial.println("Postaviti tenzometarsku traku na ravnu površinu.");
69    Serial.println("Maknuti bilo kakvu masu sa tenzometarske trake.");
70    Serial.println("Poslati 't' preko serijskog sučelja za postavljanje mjerne pogreške mjernog objekta .");
71
72    boolean _resume = false;
73    while (_resume == false) {
74      LoadCell.update();
75      if (Serial.available() > 0) {
76        if (Serial.available() > 0) {
77          char inByte = Serial.read();
78          if (inByte == 't') LoadCell.tareNoDelay();
79        }
80      }
81      if (LoadCell.getTareStatus() == true) {
82        Serial.println("Mjerenje mase neopterećenog objekta izvršeno");
83        _resume = true;
84      }
85    }
86
87    Serial.println("Postaviti poznatu masu na tenzometarsku traku.");
88    Serial.println("Potom, unijesti poznatu masu preko serijskog sučelja.");
89
90    float known_mass = 0;
91    _resume = false;
92    while (_resume == false) {
93      LoadCell.update();
94      if (Serial.available() > 0) {
95        known_mass = Serial.parseFloat();
96        if (known_mass != 0) {
97          Serial.print("Known mass is: ");
```

Output Serial Monitor

Slika 7.16 Prikaz programskog koda mikrokontrolera – 3. dio [9[

ZAVRSNI_RAD_KOD_20231113.ino

```
97            Serial.print("Known mass is: ");
98            Serial.println(known_mass);
99            _resume = true;
100         }
101       }
102     }
103
104     LoadCell.refreshDataSet();
105     float newCalibrationValue = LoadCell.getNewCalibration(known_mass);
106
107     Serial.print("Nova kalibracijska vrijednost je postavljena: ");
108     Serial.print(newCalibrationValue);
109     Serial.println(", koristi to za kalibracijsku vrijednost unutar sketch zapisa programa.");
110     Serial.print("Spremi vrijednost unutar EEPROM memorije ");
111     Serial.print(calVal_eepromAdress);
112     Serial.println("? y/n");
113
114     _resume = false;
115     while (_resume == false) {
116       if (Serial.available() > 0) {
117         char inByte = Serial.read();
118         if (inByte == 'y') {
119   #if defined(ESP8266)|| defined(ESP32)
120           EEPROM.begin(512);
121   #endif
122           EEPROM.put(calVal_eepromAdress, newCalibrationValue);
123   #if defined(ESP8266)|| defined(ESP32)
124           EEPROM.commit();
125   #endif
126           EEPROM.get(calVal_eepromAdress, newCalibrationValue);
127           Serial.print("Vrijednost ");
128           Serial.print(newCalibrationValue);
129           Serial.print("Spremi u EEPROM memoriju: ");
```

Output Serial Monitor

Slika 7.17 Prikaz programskog koda mikrokontrolera – 4. dio [9]

ZAVRSNI_RAD_KOD_20231113.ino

```
129           Serial.print("Spremi u EEPROM memoriju: ");
130           Serial.println(calVal_eepromAdress);
131           _resume = true;
132
133         }
134         else if (inByte == 'n') {
135           Serial.println("Vrijednost nije spremljena u EEPROM memoriju");
136           _resume = true;
137         }
138       }
139     }
140
141     Serial.println("Kalibracija sustava završena");
142     Serial.println("***");
143     Serial.println("Za ponovnu kalibraciju unesi 'r' .");
144     Serial.println("Za manualnu kalibraciju unesi 'c'.");
145     Serial.println("***");
146   }
147
148   void changeSavedCalFactor() {
149     float oldCalibrationValue = LoadCell.getCalFactor();
150     boolean _resume = false;
151     Serial.println("***");
152     Serial.print("Trenutna vrijednost: ");
153     Serial.println(oldCalibrationValue);
154     Serial.println("Pošalji novu vrijednost na serijsko sučelje");
155     float newCalibrationValue;
156     while (_resume == false) {
157       if (Serial.available() > 0) {
158         newCalibrationValue = Serial.parseFloat();
159         if (newCalibrationValue != 0) {
160           Serial.print("Nova kalibracijska vrijednost: ");
```

Output Serial Monitor

Slika 7.18 Prikaz programskog koda mikrokontrolera – 5. dio [9]

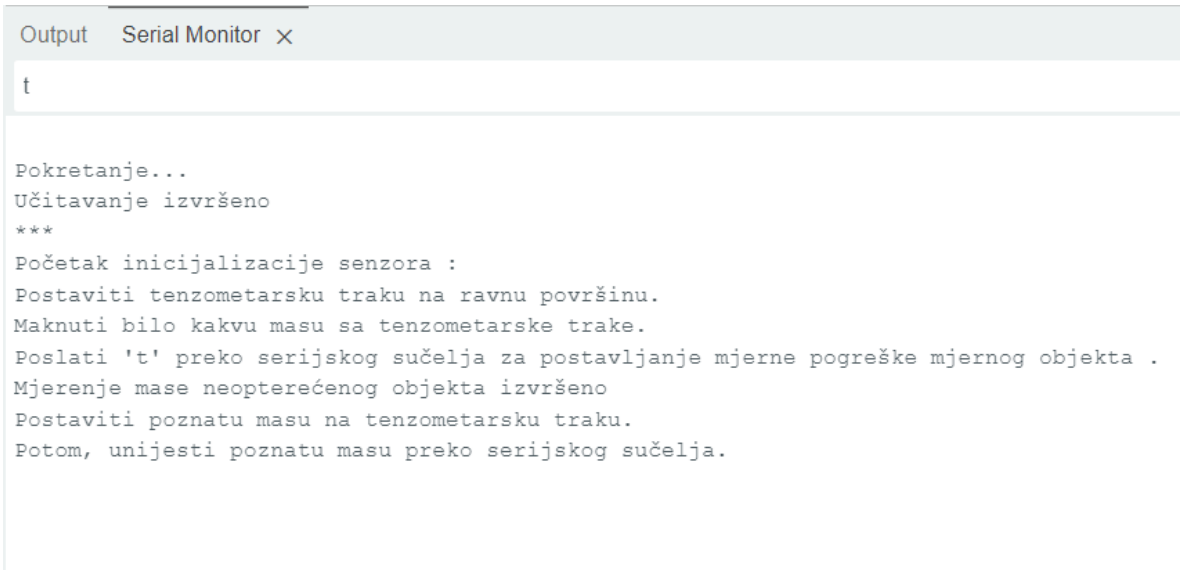
ZAVRSNI_RAD_KOD_20231113.ino

```
161           Serial.println(newCalibrationValue);
162           LoadCell.setCalFactor(newCalibrationValue);
163           _resume = true;
164         }
165       }
166     }
167     _resume = false;
168     Serial.print("Spremi tu vrijednost unutar EEPROM memorije ");
169     Serial.print(calVal_eepromAdress);
170     Serial.println("? y/n");
171     while (_resume == false) {
172       if (Serial.available() > 0) {
173         char inByte = Serial.read();
174         if (inByte == 'y') {
175 #if defined(ESP8266)|| defined(ESP32)
176           EEPROM.begin(512);
177 #endif
178           EEPROM.put(calVal_eepromAdress, newCalibrationValue);
179 #if defined(ESP8266)|| defined(ESP32)
180           EEPROM.commit();
181 #endif
182           EEPROM.get(calVal_eepromAdress, newCalibrationValue);
183           Serial.print("Vrijednost ");
184           Serial.print(newCalibrationValue);
185           Serial.print(" Spremi u EEPROM memoriju: ");
186           Serial.println(calVal_eepromAdress);
187           _resume = true;
188         }
189         else if (inByte == 'n') {
190           Serial.println("Vrijednost nije spremljena u EEPROM memoriju");
191           _resume = true;
192         }
193       }
```

Slika 7.19 Prikaz programskog koda mikrokontrolera – 6. dio [9]

8. Testiranje senzora

Testiranje senzora obavljeno je u laboratoriju za ispitivanje. Za početak potrebno je povezati sam senzor s Arduino mikrokontrolerom preko odgovarajućih ulazno/izlaznih pinova. Potom je senzor potrebno spojiti na adekvatno linearno napajanje s nazivnih 5 V. Nakon što je mikrokontroler povezan s računalom te sve veze povezane testiranje će započeti. Unutar Arduino IDE okruženja povezuje se mikrokontroler s računalnom. Prije testiranja potrebno je obaviti završnu provjeru programskog koda te ga upload - ati na mikrokontroler. Nakon povezivanja unutar aplikacijskog prikazivača moguće je vidjeti izlazne podatke senzora. Da bi podaci i sama mjerenja bila točna, potrebno je kalibrirati senzor. Postupak kalibriranja opisan je unutar samog programskog koda putem komentara.



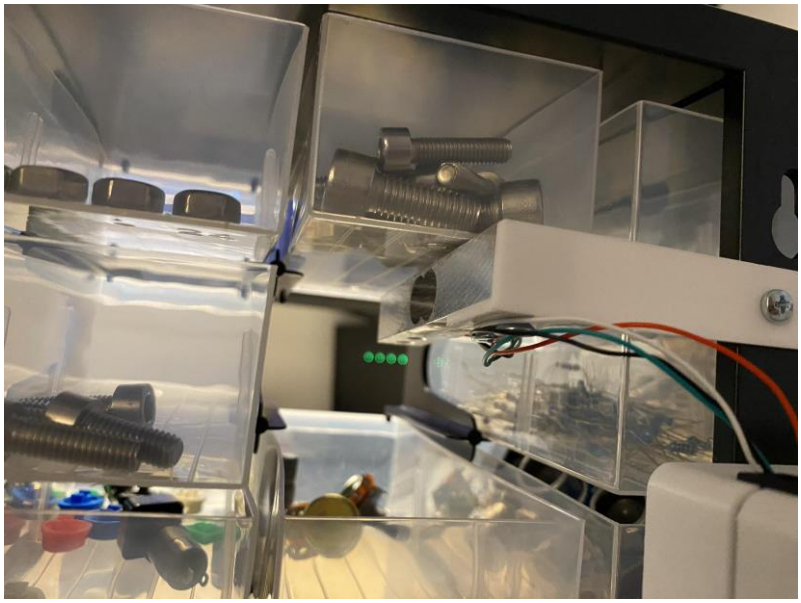
Slika 8.1 Prikaz procesa kalibracije senzora – 1.dio

Slika prikazuje potrebne korake za ispravnu kalibraciju senzora. Prvo je potrebno postaviti tenzometarsku traku na ravnu površinu. Također bitno je pobrinuti se da na područje gdje je mjerna traka nema djelovanja nikakvih vanjskih sila, odnosno da je područje mirno. Zatim, potrebno je skinuti bilo kakvu masu s trake te postali „t“ u serijski pokaznik za početak kalibracijskog procesa. Senzor zatim postavlja vrijednost unutar memorijskog registra i postavlja traku u nulto stanje. Potom je potrebno postaviti poznatu masu predmeta na mjernu traku.

```
Pokretanje...
Učitavanje izvršeno
***
Početak inicijalizacije senzora :
Postaviti tenzometarsku traku na ravnu površinu.
Maknuti bilo kakvu masu sa tenzometarske trake.
Poslati 't' preko serijskog sučelja za postavljanje mjerne pogreške mjernog objekta .
Mjerenje mase neopterećenog objekta izvršeno
Postaviti poznatu masu na tenzometarsku traku.
Potom, unijesti poznatu masu preko serijskog sučelja.
Mjerena masa je : 10.00
Nova kalibracijska vrijednost je postavljena: 1.40, koristi to za kalibracijsku vrijednost unutar sketch zapisa programa.
Spremi vrijednost unutar EEPROM memorije 0? y/n
Vrijednost 1.40Spremi u EEPROM memoriju: 0
Kalibracija sustava završena
***
Za ponovnu kalibraciju unesi 'r' .
Za manualnu kalibraciju unesi 'c'.
***
Vrijednost tenzometarske trake: 10.00
```

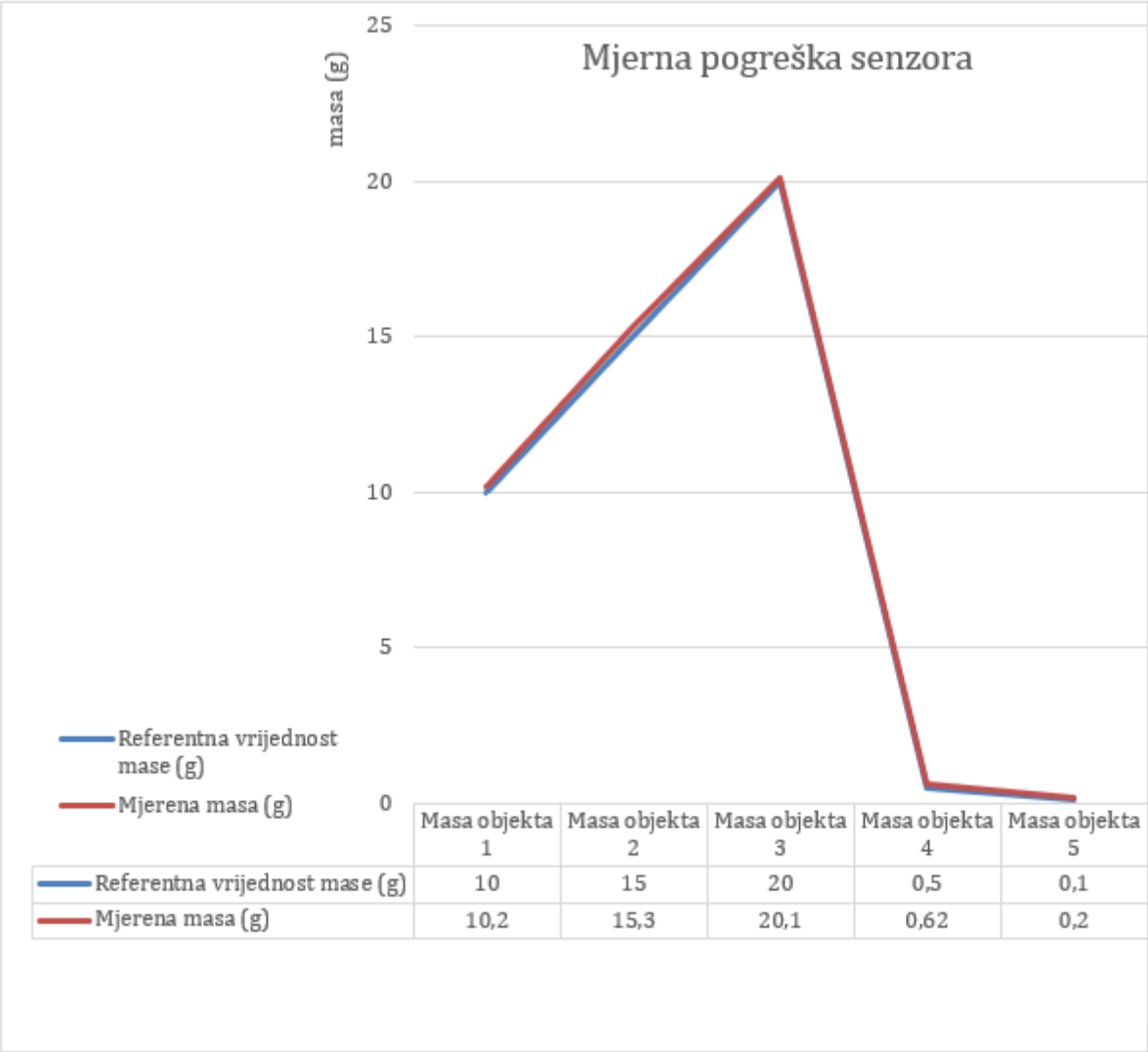
Slika 8.2 Prikaz procesa kalibracije senzora - 2. dio

Postavljanjem poznate mase na mjernu traku stvara se nova kalibracijska vrijednost senzora. Odnosno senzor postavlja off - set vrijednost. Ako dođe do prekida signala tijekom kalibracije ili preskakanja bitova unutar memorije, moguće je ponoviti postupak ili pokrenuti manualnu kalibraciju. Također bitno je napomenuti da se postavlja i dodatna off - set vrijednost kod pokazne makete zbog postavljanja mjerne trake ispod organizatora te je potrebno i oduzeti masu istog za točno mjerenje. Nakon što je senzor uspješno kalibriran može početi testiranje. Testiranje se izvodi postavljanjem mjernog objekta unutar spremnika koji je fiksiran na tenzometarsku traku.



Slika 8.3 Prikaz procesa mjerenja mase na pokaznoj maketi

- Za prvo mjerenje se koristi čelični vijak promjera M4, dužine 15 mm. Stvarna masa vijka iznosi 10 g. Mjerena masa prikazana na serijskom sučelju je 10.2 g. Uzevši u obzir mjernu pogrešku od 2 % iz mjerenja je vidljivo da je senzor ispravno kalibriran te da prikazuje točne vrijednosti predmeta.
- Za drugo mjerenje se koristi čelični vijak promjera M6, dužine 15 mm. Stvarna masa vijka iznosi 15 g. Mjerena masa prikazana na serijskom sučelju je 15.3 g. Uzevši u obzir mjernu pogrešku od 2 % iz mjerenja je vidljivo da je senzor ispravno kalibriran te da prikazuje točne vrijednosti predmeta.
- Za treće mjerenje se koristi čelični vijak promjera M8, dužine 15 mm. Stvarna masa vijka iznosi 20 g. Mjerena masa prikazana na serijskom sučelju je 20.1 g. Uzevši u obzir mjernu pogrešku od 0.5 % iz mjerenja je vidljivo da je senzor ispravno kalibriran te da prikazuje točne vrijednosti predmeta.
- Za četvrto mjerenje se koristi THT otpornik od 20 kΩ, dimenzija Φ 10/40 mm. Stvarna masa otpornika iznosi 0.5 g. Mjerena masa prikazana na serijskom sučelju je 0.62 g. Uzevši u obzir mjernu pogrešku od 24 % iz mjerenja je vidljivo da je senzor ispravno kalibriran te da prikazuje točne vrijednosti predmeta.
- Za peto mjerenje se koristi SMD otpornik od 20 kΩ, dimenzija Φ 10/40 mm. Stvarna masa otpornika iznosi 0.1 g. Mjerena masa prikazana na serijskom sučelju je 0 - 0.2 g. Uzevši u obzir mjernu pogrešku od 100 % iz mjerenja je vidljivo da senzor prikazuje netočne vrijednosti predmeta.



	Masa objekta 1	Masa objekta 2	Masa objekta 3	Masa objekta 4	Masa objekta 5
Referentna vrijednost mase (g)	10	15	20	0,5	0,1
Mjerena masa (g)	10,2	15,3	20,1	0,62	0,2

Slika 8.4 Grafički prikaz mjerenih predmeta

Iz prethodne tablice mjerenja prikazana je usporedba referentne vrijednosti mase i mjerene mase. Referentna vrijednost mase dobivena je mjerenjem predmeta kuhinjskom vagom, dok je mjerena masa rezultat mjerenja tenzometarskom trakom. Iz prethodnih mjerenja moguće je zaključiti da mogućnost upotrebe ove vrste senzora vrijedi samo za mjerenje predmeta čija nazivna masa ne prelazi masu mjerne pogreške. Odlično je za mjerenje krupnijih komponenata te različitih sitnih predmeta veće mase, dok mjerenje SMD elektroničkih komponenta nije preporučeno zbog njihove male mase. Mjerenjem SMD komponenti nemoguće je odrediti prvobitnu svrhu senzora te odrediti ukupan broj komponenti unutar spremnika uz dobiveni jediničnu masu samog predmeta.

9. Pokazna maketa

Završna faza ovog projekta je povezivanje elektronike unutar pokazne makete. Koristeći se 3D printanim dijelovima, sastavljeno je kućište za elektroniku. Pokazna maketa pokazuje pravi primjer i svrhu ovog rada. Postavljanjem tenzometarske trake na središnji dio potporne stjenke omogućuje se ispravno mjerenje mase neke komponente ili sitnog djela. Mjerenjem promjene otpora na tenzometarskoj traci uzrokovanim tlačnim naprezanjem stijenke omogućeno je praćenje same promjene napona unutar petlje. Daljnjom obradom analognog signala ADS 1256 sklop povećava rezoluciju mjerenja te omogućuje prikazati veće oscilacije napona unutar mosta. Izlazna vrijednost signala se zatim prikazuje u rasponu od 0 do 2^{24} vrijednosti, što omogućuje veću preciznost mjerenja. Senzor potom šalje mjereni signal u većoj rezoluciji Arduino mikrokontroleru koji s pomoću formule za izračunavanje mase, konstantno prati promjenu iste na senzoru. Promjena mase prikazuje se preko „Serial Monitor“ prikaznika. Cilj makete je prikazati pojednostavljeni prikaz mjerenja elemenata u industrijskom postrojenju.

Slika 9.1 Prikaz pokazne makete

10. Zaključak

Razvoj ovakve vrste elektroničkog podsustava za mjerenje mase vrlo je učinkovito i točno. Sama izvedba mjerenja na vrlo jednostavan način, omogućuje korisniku konstantni pristup elementima unutar organizatora. Također takva primjena sustava preporučljiva je za korištenje u industrijske svrhe. Jednostavnost same izvedbe kućišta i elektronike dodatno ušteđuje na vremenu i novčanim resursima. Kućište je vrlo elegantno montirano na dijelove organizatora što sprječava mogućnost bilo kakvog ometanja elektronike sa samim mehaničkim dijelovima istog.
Naravno, postoje razna moguća poboljšanja koja se mogu primijeniti na ovaj rad, a to su:

- Promjena dizajna elektroničke pločice tako da se sam senzor i mikrokontroler zajedno implementiraju.
- Izmjena tenzometarske trake od 120 Ω s trakom većeg otpora.
- Implementacija IoT platforme za komunikaciju više senzora unutar Cloud platforme.
- Odvajanje samog mjernog objekta od senzora, te izvedba komunikacije putem RF prijamnika i predajnika.
- Promjena same formule mikrokontrolera, kako bi sustav na temelju jedinične mase mogao odrediti i količinu elemenata unutar mjerne kutije. Time bi korisnik dodatno mogao vršiti kontrolu nad brojem elemenata unutar organizatora. Također bi se i time pojednostavio proces inventure unutar nekog postrojenja.

Mane uređaja za mjerenje mase:

- Senzor ne može precizno mjeriti predmete čija je masa manja od 1 g.
- Senzor koristi promjenu otpora djelovanjem tlačnog naprezanja trake kao mjernu veličinu. Što znači da je potrebno osigurati mirnu i ravnu površinu gdje će se takav isti senzor koristiti.

Prednosti uređaja za mjerenje mase:

- Koristan za mjerenje većih masa.
- Jednostavnost izvedbe senzora.
- Mogućnost skaliranja senzora za industrijske upotrebe.
- Velika brzina mjerenja.

Prikaz ovakvog sustava definitivno ima više pozitivnih strana nego negativnih u smislu jednostavnosti korištenja te same brzine mjerenja. No financijska isplativost je u većem slučaju održiva ako se sustavi primjenjuju unutar većih industrijskih postrojenja gdje jako važno vrijeme i brzina rada pogona.

11. Literatura

[1]dipl. ing. Horvatić Miroslav: Procesna instrumentacija, Sveučilište Sjever u Varaždinu, Skripte za učenje

[2]https://www.festo.com/hr/hr/c/proizvodi/automatizacija-tvornice/senzori/induktivni-senzori-id_pim129/?page=0, dostupno 27.11.2023

[3]dr. sc. mag. ing. Hižak Jurica: Fizika 1, Sveučilište Sjever u Varaždinu, Skripte za učenje

[4] https://hr.wikipedia.org/wiki/Kidalica, dostupno 27.11.2023

[5]https://sfsb.unisb.hr/ksk/cvrstoca/web_cvrstoca/N_eksperiment/N_d.pdf, dostupno 22.11.2023

[6] https://www.ti.com/product/ADS1256, dostupno 27.11.2023

[7] https://www.jameco.com/Jameco/workshop/CircuitNotes/CN-Arduino-uno.html, dostupno 27.11.2023

[8] https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi/all, dostupno 27.11.2023

[9] https://github.com/adienakhmad/ADS1256, dostupno 27.11.2023

[10] https://www.altium.com/documentation/altium-designer/managing-design-changes-between-schematic-pcb-overview?version=18.1, dostupno 27.11.2023

Popis slika

Slika 2.1 Primjer induktivnog senzora [1]
Slika 2.2 Primjer univerzalne kidalice [4]
Slika 3.1 Dijagram rastezanja za niskougljični čelik [5]
Slika 3.2 Ovisnost otpora od značajki mjerne trake [5]
Slika 3.3 Vrste elektrootporničkih mjernih traka [5]
Slika 3.4 Postavljanje mjerne trake na površinu objekta [5]
Slika 3.5 Wheatstoneov mjerni most [5]
Slika 3.6 Primjer montirane tenzometarske trake na mjerni objekt [5]
Slika 4.1 Tiskana pločica senzora
Slika 4.2 Softverski paket Altium Designer [10]
Slika 4.3 Napajanje senzora - AMS1117 - 3.3
Slika 4.4 Napajanje senzora - AMS1117 - 2.5
Slika 4.5 Decoupling kondenzatori operacijskog pojačala
Slika 4.6 Decoupling kondenzatori na digitalnom izvoru +3.3 V
Slika 4.7 Decoupling kondenzatori na analognom izvoru +5 V
Slika 4.8 Vanjski generator takta
Slika 4.9 Shema ulaznog konektora senzora
Slika 4.10 Shema Sensor Output/MCU konektora
Slika 4.11 Blok dijagram ADS 1256 sklopa [6]
Slika 4.12 Prikazni spoj SPI sučelja s mikrokontrolerom MSC 12xx [6]
Slika 5.1 Prikaz odjeljka za odabir komponenti preko on - line baze komponenti
Slika 5.2 Prikaz odlomka za stvaranje pravila prilikom dizajniranja PCB Layout - a
Slika 5.3 Shema senzora za mjerenje mase primjenom tenzometarske trake
Slika 5.4 Prikaz veza i otiska komponenti na prednjoj strani tiskane pločice
Slika 5.5 Prikaz veza na stražnjoj strani tiskane pločice
Slika 5.6 Prikaz sloja „Top Overlay“ na prednjoj strani tiskane pločice
Slika 5.7 Prikaz sloja „Soldermask - a“ na prednjoj strani tiskane pločice
Slika 5.8 Prikaz pozicija montažnih rupa i prespoja prednje i stražnje strane tiskane pločice
Slika 5.9 3D model završne verzije tiskane pločice
Slika 6.1 Završna izvedba tiskanih pločica
Slika 6.2 Prikaz postavljenih „header“ konektora na tiskanoj pločici
Slika 7.1 Prikaz Arduino Uno mikrokontrolera [7]

Slika 7.2 Prikaz asinkronog serijskog prijenosa podataka [8]
Slika 7.3 Prikaz sinkronog serijskog prijenosa podataka [8]
Slika 7.4 Prikaz povezivanja kontrolera i periferije sinkronom vezom [8]
Slika 7.5 Prikaz čitanja i pisanja informacija SPI komunikacijom te logičko stanje CS linije [8]
Slika 7.6 Prikaz .h programskog djela koda – 1.dio [9]
Slika 7.7 Prikaz .h programskog djela koda – 2.dio [9]
Slika 7.8 Prikaz .h programskog djela koda – 3.dio [9]
Slika 7.9 Prikaz .c++ programskog djela koda – 1. dio [9]
Slika 7.10 Prikaz .c++ programskog djela koda – 2. dio [9]
Slika 7.11 Prikaz .c++ programskog djela koda – 3. dio [9]
Slika 7.12 Prikaz .c++ programskog djela koda – 4. dio [9]
Slika 7.13 Prikaz .c++ programskog djela koda – 5. dio [9]
Slika 7.14 Prikaz programskog koda mikrokontrolera – 1. dio [9]
Slika 7.15 Prikaz programskog koda mikrokontrolera – 2. dio [9]
Slika 7.16 Prikaz programskog koda mikrokontrolera – 3. dio [9]
Slika 7.17 Prikaz programskog koda mikrokontrolera – 4. dio [9]
Slika 7.18 Prikaz programskog koda mikrokontrolera – 5. dio [9]
Slika 7.19 Prikaz programskog koda mikrokontrolera – 6. dio
Slika 8.1 Prikaz procesa kalibracije senzora – 1.dio
Slika 8.2 Prikaz procesa kalibracije senzora – 2.dio
Slika 8.3 Prikaz procesa mjerenja mase na pokaznoj maketi
Slika 8.4 Grafički prikaz mjerenih predmeta
Slika 9.1 Prikaz pokazne makete

Sveučilište
Sjever

SVEUČILIŠTE
SJEVER

IZJAVA O AUTORSTVU
I
SUGLASNOST ZA JAVNU OBJAVU

Završni/diplomski rad isključivo je autorsko djelo studenta koji je isti izradio te student odgovara za istinitost, izvornost i ispravnost teksta rada. U radu se ne smiju koristiti dijelovi tuđih radova (knjiga, članaka, doktorskih disertacija, magistarskih radova, izvora s interneta, i drugih izvora) bez navođenja izvora i autora navedenih radova. Svi dijelovi tuđih radova moraju biti pravilno navedeni i citirani. Dijelovi tuđih radova koji nisu pravilno citirani, smatraju se plagijatom, odnosno nezakonitim prisvajanjem tuđeg znanstvenog ili stručnoga rada. Sukladno navedenom studenti su dužni potpisati izjavu o autorstvu rada.

Ja, ROKO RISEK *(ime i prezime)* pod punom moralnom, materijalnom i kaznenom odgovornošću, izjavljujem da sam isključivi autor/~~ica~~ završnog/~~diplomskog~~ (*obrisati nepotrebno*) rada pod naslovom MJERENJE MASE PRIMJENOM TENZOMETARSKE TRAKE *(upisati naslov)* te da u navedenom radu nisu na nedozvoljeni način (bez pravilnog citiranja) korišteni dijelovi tuđih radova.

Student/ica:
(upisati ime i prezime)

Rišk
(vlastoručni potpis)

Sukladno Zakonu o znanstvenoj djelatnost i visokom obrazovanju završne/diplomske radove sveučilišta su dužna trajno objaviti na javnoj internetskoj bazi sveučilišne knjižnice u sastavu sveučilišta te kopirati u javnu internetsku bazu završnih/diplomskih radova Nacionalne i sveučilišne knjižnice. Završni radovi istovrsnih umjetničkih studija koji se realiziraju kroz umjetnička ostvarenja objavljuju se na odgovarajući način.

Ja, ROKO RISEK *(ime i prezime)* neopozivo izjavljujem da sam suglasan/~~na~~ s javnom objavom završnog/~~diplomskog~~ (*obrisati nepotrebno*) rada pod naslovom MJERENJE MASE PRIMJENOM TENZOMETARSKE TRAKE *(upisati naslov)* čiji sam autor/~~ica~~.

Student/ica:
(upisati ime i prezime)

Rišk
(vlastoručni potpis)