# SVEUČILIŠTE U SPLITU

**FAKULTET ELEKTROTEHNIKE, STROJARSTVA I BRODOGRADNJE**

**SEMINARSKI RAD**

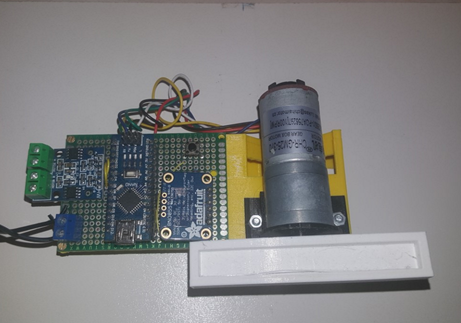
**ZAMJENA ON-OFF REGULATORA POLOŽAJA OSOVINE DC MOTORA PID REGULATROM**

**IVAN SIČAJA**

Split, rujan 2021.

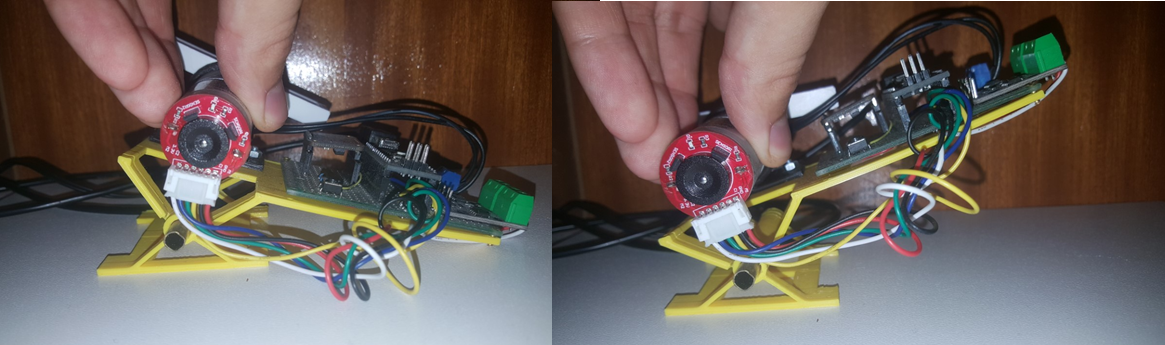
# UVOD

Sustav koji želimo regulirati prikazan je na sljedećoj slici.



Slika 1.1 Prikaz sustava na kojem se vrši testiranje

Ukratko sustav se sastoji od DC motora koji posjeduje dvije izlazne osovine. Na jednu izlaznu osovinu (osovinu prijenosa DC motora) montirana je horizontalna ploča kompaktnih dimenzija, čiji položaj nastojimo održavati u horizontalnoj ravnini neovisno o stupnju zakrivljenosti cjelokupnog sustava oko osi koja prolazi kroz središte izlaznih osovina DC motora. Na drugu izlaznu osovinu DC motora montiran je enkoder pomoću kojeg imamo uvid u trenutni položaj horizontalne ploče koja je montirana na izlaznu prijenosa motora. Mozak cijelog sustava je mikrokontroler Arduino Nano pomoću kojeg preko kontrolera motora upravljamo DC motorom, očitavamo trenutni položaj izlazne osovine prijenosa DC motora kako i nagib cjelokupnog sustava u odnosu na osi koja prolazi kroz središte izlaznih osovina DC motora. Nagib očitavamo pomoću žiroskopa Adafruit BNO055 uređaja. Cilj zadatka je realizirati što je moguće bolji PID regulator koji bi montiranu horizontalnu ploču sustava nastojao držati u početnom (horizontalnom) položaju unatoč rotaciji cjelokupnog sustava u smjeru koji je predviđen samim sklopovljem.

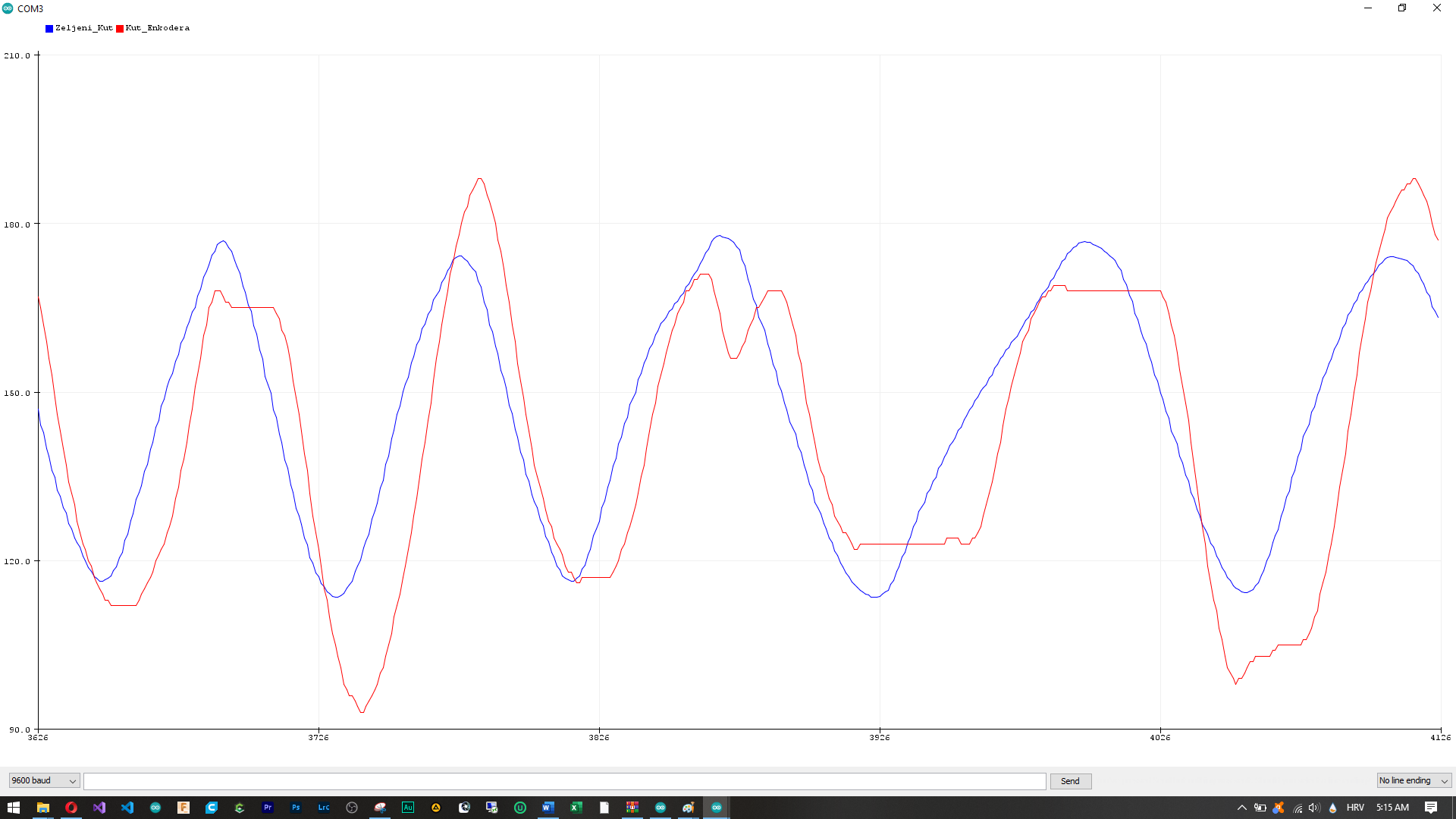


Slika 1.2. Prikaz smjera rotiranja sustava

# POSTUPAK ZAMJENE ON-OFF REGULATORA PID REGULATOROM

Kao osnovni regulator u navedeni sustav dolazi integriran ON-OFF regulator čija je karakteristika u odnosu na položaj izlazne osovine prijenosa DC motora prikazana na sljedećoj slici. Frekvencija osciliranja sustava je približno 1Hz.

Iz izlazne karakteristike sustava vidljivo je da izlazna osovina DC motora ne prati krivulju željenog kuta na adekvatan način , odnosno vidimo da regulator sustava ima jako velike tolerancije na željeni položaj krivulje te ukoliko se nalazi unutar tolerancija zaustavit će vrtnju izlazne osovine DC motora što će rezultirati pogreškom. Stoga je prvi parametar regulatora koji želimo poboljšati upravo parametar preciznosti. Preciznost sustava povećavamo na način da toleranciju trenutno i željenog položaja s 15° snizimo na 0° što nam omogućava sama izvedba PID regulatora što je prednost u odnosu na ON-OFF regulator ovog sustava, kojem ako smanjimo toleranciju ispod 15° dolazi do izrazito povećane mogućnosti da sustav postane nestabilan.



Slika 2.1 Prikaz odziva sustava reguliranog ON-OFF regulatorom

Uz malu preciznost odziv ON-OFF regulatora karakterizira relativno velik prebačaj kao i određene oscilacije. Za frekvencije osciliranja sustava koje su manje od 1 Hz odziv sustava postaje vrlo trom.



Slika 2.2 Prikaz odziva sustava reguliranog ON-OFF regulatorom pri frekvenciji od približno 0.3 Hz

Budući da ON-OFF regulator za naš sustav koji karakterizira izrazita nelinearnost vrlo i velika mrtva zona DC motora potrebno je korištenje naprednijeg regulatora odnosno regulatora pomoću kojeg možemo imati više utjecaja na oblik krivulje odziva izlazne osovine DC motora. Za naš slučaj smo se odlučili za PID (eng. Proportional Integral Derivative) regulator. Kao što sam naziv kaže ovaj regulator se sastoji od proporcionalnog, integralnog i derivativnog dijela čije ćemo utjecaje ukratko opisati u nastavku.

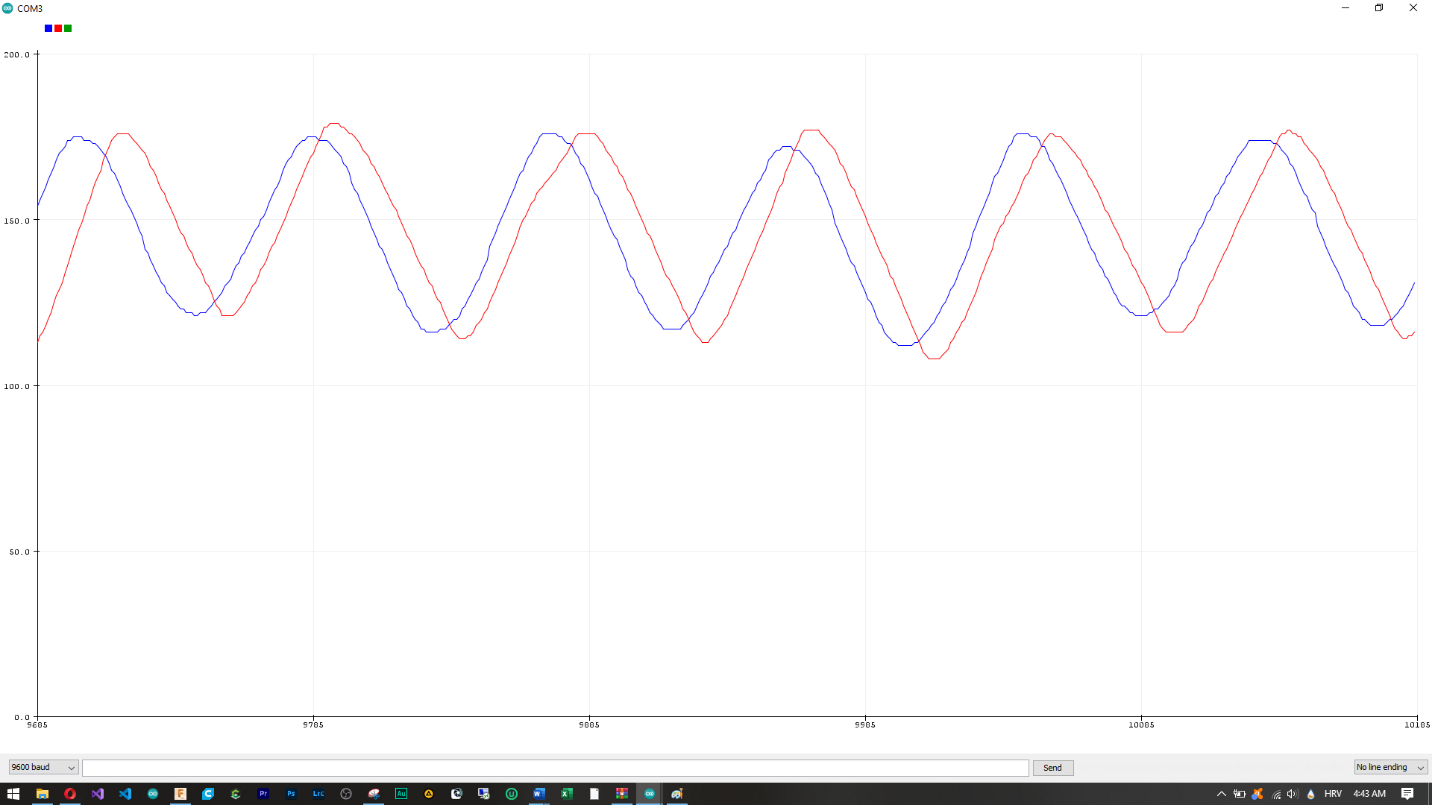
**Proporcionalni dio** – je direktno vezan s odstupanjem trenutne od željene vrijednosti, uvelike utječe na vrijeme odziva sustava odnosno povećanjem P parametra regulatora ubrzujemo odziv sustava ali također postoji mogućnost da prevelikom vrijednošću P parametra stvorimo neželjeno veliki prebačaj izlaza sustava u odnosu na željeni položaj, stoga je pomoću analize odziva sustava odabrati adekvatnu vrijednost P parametra, kako sustav ne bi bio trom ili nestabilan. Za analizu sustava za ovaj projekt koristit ćemo „Serial Plotter“ mogućnost Arduino IDE programa. P parametar ima mal učinak na vrijeme smirivanja sustava

**Intgralni dio** – regulatora omogućava poboljšanje preciznosti sustava ukoliko je izlaz sustava vrlo blizu željenom izlazu, što ga u ovom području čini dominantnim u odnosu na P i D parametre sustava čije je djelovanje više vezano za područja u kojima izlaz sustava više odstupa od željenog odziva. Integralni dio regulatora utječe na brzinu odziva sustava što može imati i negativne posljedice kako što je generiranje prebačaja sustava

**Derivativni dio** – vezan je za vrijeme promjene pogreške, što PD regulator čini robusnijim u odnosu na P regulator. Temeljna uloga derivativnog dijela da smanjuje prebačaj i vrijeme smirivanja izlaza sustava. Nedostatak ovog parametra PID regulatora je taj što ima određeni utjecaj na vrijeme porasta sustava koje uglavnom malo povećava što rezultira konačnim sporijim odzivom sustava. Odabirom adekvatne vrijednost D parametra regulatora smanjujemo šumove izlazne krivulje sustava.

Budući da smo analizom sustava dobili mrtvu zonu odziva sustava koja seže do vrijednosti PWM signala PWM=100, odnosno mrtva zona odziva sustava je PWM=49 ako se izlazna osovina već giba, stoga ukupni PID upravljački signal sustava kojeg predstavljamo PWM signalom postavljamo na minimalnu vrijednost od PWM=50 kako bismo prebrodili mrtvu zonu DC motora dok preostali raspon upravljačkog signala od PWM>50 i PWM< 255 generiramo pomoću PID regulatora. Kao konstante odnosno ukupne vrijednosti P I i D dijela u prvom koraku ugađamo da budu u rasponu od oko 205 (MAX PWM- MIN PWM=255-50=205), te određene konstante Kp, Ki i Kd ovisno o prikazima utjecaja pojedine konstante na odziv sustava na Serial Ploteru, modificiramo na željeni način.

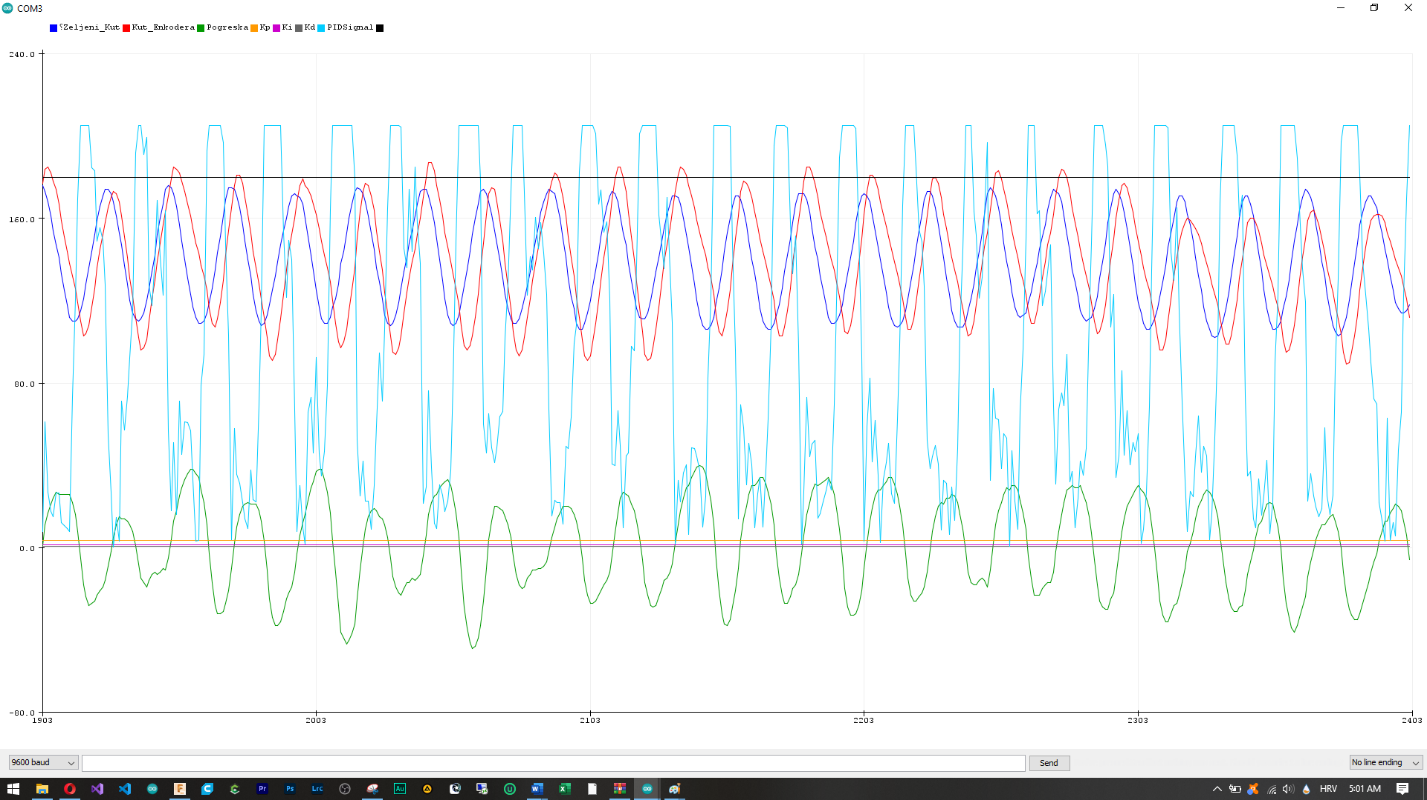
Prikaz odziva sustava reguliranog PID regulatorom u odnosu na željeni položaj izlazne osovine DC motora dan je u nastavku.



Slika 2.3 Prikaz odziva sustava reguliranog PID regulatorom

Iz izlazne karakteristike vidljivo je prebačaj sustava uvelike umanjen u odnosu na ON-OFF regulator, kao i nepoželjne oscilacije dok je sama preciznost sustava uvelike povećana.

Prikaz odziva sustava, konstanti te ukupnog PID upravljačkog signala dan je na sljedećoj slici.



Slika 2.4 Prikaz odziva sustava reguliranog PID regulatorom s parametrima koji utječu na generiranje upravljačkog signala sustava

Način realizacije PID regulatora na Arduino platformi vrlo je jednostavan i vidljiv je u priloženom kodu, kao i detaljna analiza odziva sustava.

# ZAKLJUČAK

PID regulator u odnosu na ON-OFF regulator omogućuje realiziranje robusnijeg upravljačkog signala baziranog na kombinaciji pogreške, promjene i akumulacije pogreške odziva sustava u odnosu na željni sustav što nam omogućava utjecaj na brzinu odziva sustava, povećanje preciznosti odziva sustava kao i smanjenje utjecaja šumova. Konačni iznosi kombinacija P, I i D djela regulatora treba biti unutar reda veličine potrebnog iznosa, što postižemo analizom utjecaja pojedine komponente PID regulatora na konačni odziv sustava.

# PROGRAMSKI KOD

//kod za regulaciju nagiba grede

#include <util/atomic.h> // For the ATOMIC\_BLOCK macro

#define PIND\_A 2

#define PIND\_B 3

#define ENC\_PIN\_A 2 //pin kanala A enkodera

#define ENC\_PIN\_B 3 //pin kanala B enkodera

#define READ\_ENC\_A bitRead(PIND, PIND\_A)

#define READ\_ENC\_B bitRead(PIND, PIND\_B)

//ukljucivanje bibilioteka potrebnih za rad sa senzorom BNO055

#include <Wire.h>

#include <Adafruit\_Sensor.h>

#include <Adafruit\_BNO055.h>

#include <utility/imumaths.h>

#define MOTORB\_1A 5

#define MOTORB\_1B 6

//stvaranje instance

Adafruit\_BNO055 bno = Adafruit\_BNO055(55);

float kut\_BNO;

volatile long encBrojac = 0;//###

long prevT = 0;

float eprev = 0;

float eintegral = 0;

//##################################################################

void encISR()

{

if (READ\_ENC\_A == HIGH)//kanal A L->H (rastuci brid)

READ\_ENC\_B == HIGH ? encBrojac++ : encBrojac--;

else //kanal A H->L (padajuci brid)

READ\_ENC\_B == HIGH ? encBrojac-- : encBrojac++;

}

void motorSmjer1(int setSpeed)

{

digitalWrite(MOTORB\_1A, LOW);

analogWrite(MOTORB\_1B, setSpeed);

}

void motorSmjer2(int setSpeed)

{

analogWrite(MOTORB\_1A, setSpeed);

digitalWrite(MOTORB\_1B, LOW);

}

void motorStop()

{

digitalWrite(MOTORB\_1A, LOW);

digitalWrite(MOTORB\_1B, LOW);

}

//##################################################################

void setup()

{

Serial.begin(9600);

//Serial.println("Zeljeni\_Kut,Kut\_Enkodera");

Serial.println("Zeljeni\_Kut,Kut\_Enkodera,Pogreska,P\_DIO,I\_DIO,D\_DIO,PIDSignal");

pinMode(ENC\_PIN\_A, INPUT\_PULLUP);

pinMode(ENC\_PIN\_B, INPUT\_PULLUP);

if (!bno.begin()) //ako je doslo do problema s povezivanjem s BNO055

{

Serial.print("Problem s povezivanjem sa senzorom BNO055!");

while (1);

}

bno.setExtCrystalUse(true);

attachInterrupt(digitalPinToInterrupt(ENC\_PIN\_A), encISR, CHANGE);

pinMode(MOTORB\_1A, OUTPUT);

pinMode(MOTORB\_1B, OUTPUT);

digitalWrite(MOTORB\_1A, LOW);

digitalWrite(MOTORB\_1B, LOW);

}

void loop() {

//dohvati polozaj

sensors\_event\_t eventT;

bno.getEvent(&eventT);

kut\_BNO = eventT.orientation.z;

kut\_BNO < 0 ? kut\_BNO += 360 : kut\_BNO;

int target = kut\_BNO;

//int target = 120;

// tu ide algoritam regulatora

// PID constants

float kp = 3.5;

float kd = 0.5;

float ki = 1.5;

// time difference

long currT = micros();

float deltaT = ((float) (currT - prevT)) / ( 1.0e6 );

prevT = currT;

// Read the position in an atomic block to avoid a potential

// misread if the interrupt coincides with this code running

// see: https://www.arduino.cc/reference/en/language/variables/variable-scope-qualifiers/volatile/

int kut = 0;

ATOMIC\_BLOCK(ATOMIC\_RESTORESTATE) {

kut = encBrojac / (4.6);

}

// error

int e = kut - target;

// derivative

float dedt = (e - eprev) / (deltaT);

// integral

eintegral = eintegral + e \* deltaT;

// control signal

float u = kp \* e + kd \* dedt + ki \* eintegral;

/\*

Serial.print("PID TOTAL BEFOR MAP: ");

Serial.print(PID\_total);

Serial.print(",");

Serial.print(PID\_p);

Serial.print(",");

Serial.print(PID\_i);

Serial.print(",");

Serial.print(PID\_d);

Serial.print(".");

\*/

/\*

Serial.print(" PID TOTAL AFTER MAP: ");

Serial.print(PID\_total);

\*/

if (u < 0) {

u = u\*(-1);

}

if (u > 205) {

u = 205;

}

// store previous error

eprev = e;

//gotov algoritam regulatora

if ((kut - target) > 0)

motorSmjer1(u + 50);

if ((kut - target) < -0)

motorSmjer2(u + 50);

if (abs(kut - target) == 1)

motorStop();

Serial.print(target);

Serial.print(" ");

Serial.print(kut);

Serial.print(" ");

Serial.print(e);

Serial.print(" ");

Serial.print(kp \* e);

Serial.print(" ");

Serial.print(kd \* dedt);

Serial.print(" ");

Serial.print(kd \* dedt);

Serial.print(" ");

Serial.print(u);

Serial.print(" ");

Serial.println(180);

//ispis stanja

//Serial.print(millis());

//Serial.print(" ");

//Serial.print(encBrojac);

/\*

Serial.println(" ");

Serial.print("Kut ziroskopa po z osi: ");

Serial.println(kut\_BNO);

Serial.print("Kut enkodera: ");

Serial.println(kut);

Serial.print("RAZLIKA KUTEVA Ž-E: ");

Serial.println(kut\_BNO-kut);

\*/

/\*

Serial.print(" PID TOTAL: ");

Serial.print(PID\_total);

Serial.print(" P: ");

Serial.print(PID\_p);

Serial.print(" D: ");

Serial.print(PID\_d);

Serial.print(" I: ");

Serial.println(PID\_i);

\*/

//delay(1000);

// digitalWrite(MOTORB\_1A, LOW);

// analogWrite(MOTORB\_1B, 160);

}