

Bilboko Ingeniaritza Eskola

Industria Elektronikaren eta Automatikaren Ingeniaritzako Gradua  
Gradu Amaierako Lana  
2015 / 2016



eman ta  
zabal zazu



**(IBILGAILU AUTONOMO ERROBOTIZATUAREN KONTROLA)**

**4.DOKUMENTUA: KODEA**

**IKASLEAREN DATUAK**

IZENA: ENEKO

ABIZENAK: PUMAREJO FRADUA

SIN.:

DATA:

**ZUZENDARIAREN DATUAK**

IZENA: KOLDO

ABIZENAK: BASTERRETXEIA OYARZABAL

SAILA: TEKNOLOGIA ELEKTRONIKOA

SIN.:

DATA:

MODUA:

- JATORRIZKO  
A  
 KOPIA

# AURKIBIDEA

1	Programa nagusia .....	- 1 -
2	Azpi programak.....	- 2 -
2.1	Martxa .....	- 2 -
2.2	PID .....	- 3 -
2.3	Distantzia neurtu .....	- 4 -
2.4	Behaketa.....	- 5 -
2.5	Posizio1.....	- 6 -
2.6	Posizio2.....	- 7 -
2.7	Biratu .....	- 8 -
2.8	Bidebarri .....	- 9 -
2.9	Gelditu .....	- 9 -
3	Etenduren kudeaketa .....	- 10 -
3.1	Tenporizadoreak .....	- 10 -
3.2	IC Moduluak .....	- 10 -

## 1 Programa nagusia

Goi mailako kontrola, azpi programen bitartez nabigazioa eta hau garatzeko periferikoen kontrola.

```
#include<p24FJ128GA010.h>
#include <math.h>

char a,b,hasieratu,prest,eskuma,eskerra,biraketa,kont,itzaroen;
int erradioa,laginketa,itzaroten,i,p,oztopo,muga,luze,hurrunegi,neurturea,aurre_irtera,aurre_irtera2;
float arrapaketa,ms,denbora,denbora1,denbora2,denbora3,denbora4,denbora5,maiztasuna2,konsigna,
float kap1,kap2,dembora,periodo,errorea,errorea2,errorebatura2,errorebatura,P,I,P2,I2,norantza,irtera,irter
signed int x_h,x_l,z_h,z_l,y_h,y_l,b_angelua,angelua,buka_posizio,nago,distantzia;
signed long X,Y,Z;
double posizio,BC,BC_angelua,arg;
```

```
int main (void)
{
//Osziladore
OSCCON=0x7700;//Barne osziladorearekin
CLKDIV = 0x0000;//8MHz
//zenbait hazieratze
Dis_helmuga=400;
Dis_egina=0;
eskerra=0;
eskuma=0;
biraketa=0;
//martza pultsadore
_TRISD13 = 1;// PORTD-ko 8.pina (IC1)
PORTDbits.RD13=0;// balio altua
while((PORTDbits.RD13=0))
{
}
//HASIERATU
posizio0();
norantza=posizio;
martza(); //azpiprogramaexekutatu
//Begizta nagusia
while (1)
{
while(Dis_helmuga>=Dis_egina)
{
if(posizio<norantza)
{
konsigna2=102;
konsigna=98;
}
if(posizio>norantza)
{
konsigna2=98;
konsigna=102;
}
if(posizio=norantza)
{
konsigna2=100;
konsigna=100;
}
if(laginketa==1) //laginketa denbora
{
laginketa=0;
_T1IE=0;
distantzianeurtu();
posizio2();
}
}
}

if(oztopo==1)
{
gelditu();
A=Dis_helmuga-Dis_egina;

behaketa();
biratu();
saihesbide1=disluzea/2;
saihesbide=0;
posizio2();

posizio=norantza;
martza();

konsigna=50;

konsigna2=50;
while (saihesbide<=saihesbide1)
{
posizio2();
if(posizio<norantza)
{
konsigna2=102;
konsigna=98;
}
if(posizio>norantza)
{
konsigna2=98;
konsigna=102;
}
if(posizio=norantza)
{
konsigna2=100;
konsigna=100;
}
}
gelditu();
bidebarri();
biratu();
gelditu();
posizio2();
norantza=posizio;
martza();
}
_T1IE=1;
gelditu();
}
}
```

## 2 Azpi programak

### 2.1 Martxa

Motorren kontrola. Motorrak abiatzeko PWM modualuak, input capture moduluak eta modulu hauei eta laginketa maiztasunari erlazionatutako tenporizadoreak abiaratzea.

```

void martxa(void)
{
// PORTUAK
_TRISD8 = 1; // PORTD-ko 8.pina (IC1) sarrera bezala
_TRISD0 = 0; // PORTD-ko 0.pina (OC1) irteera bezala
_TRISD1 = 0; // PORTD-ko 1.pina (OC2) irteera bezala
_TRISD6=0;
_TRISA9=0;
PORTDbits.RD6=0; // balioaltua
PORTAbits.RA9=1; // balioaltua

//Balio ezberdinen hasieraketa

TMR2=0; TMR3=0; TMR1=0;
erradioa=4;
denbora1=0;
denbora2=0;
konsigna=100;
konsigna2=100;
errorea=0;
errorebatura=0;
errorea2=0;
errorebatura2=0;
a=0;

//PWM
OC1CON = 0x0000; // Output compare itzali
OC1R = 35; // comparaketa erregistroa hasieratu
OC1RS = 35; // bigarren konparaketa erregistroa hasieratu
OC1CON=0x000e; //timer3
OC2CON = 0x0000; // Output compare itzali/ OC2R = 35;
OC2R = 35; // comparaketaerregistroa hasieratu
OC2RS = 35; // bigarren konparaketa erregistroa hasieratu
OC2CON=0x000e; //timer 3 PWM biak berdina izateko
//TIMER 3-ren gaitzea etakonfigurazioa
PR3 = 313; //PWM 200 Hz
T3CON = 0x8020; // 16 biteko kontaketa, barne erlojua Fosc/2, 1:64 preskalatua
//TIMER 1-ren gaitzea eta konfigurazioa (laginketa eta distantziaren neurketa)
TMR1 = 0;
PR1=6250; //laginketa denbora 0.1s
T1CON = 0x8020; //preskale 1:64, 16 biteko kontagailua
IEC0bits.T1IE= 1; // etenduragaitu
IFS0bits.T1IF = 0;
//IC1 moduluaren gaitzea etakonfigurazioa
IC1CON = 0x0000; //Reset
IC1CON = 0x0083; // Kaptura ertz gorakorrero, interruptzioa kaptura bakoitzeko, timer2
_IC1IE = 1; //Interruptzioakgaitu
_IC1IF = 0;
//IC4 moduluaren gaitzea etakonfigurazioa
IC4CON = 0x0000; //Reset
IC4CON = 0x0083; // Kaptura ertz gorakorrero, interruptzioa kaptura bakoitzeko, timer2
_IC4IE = 1; //Interruptzioakgaitu
_IC4IF = 0;
// IC1 moduluaren timerraren konfigurazioa (TIMER2)
TMR2 = 0; // hasieratu
T2CON = 0x8030; // Gaitu, eskalatua 1:256, 16 biteko kontagailua

```

## 2.2 PID

PID balio berria kalkulatzeko programa, laginketa maiztasunarekin batera exekutatzen da abiadura balioa hartu eta OCxRS erregistroan kargatu beharreko balioaren kalkulua.

```
void PID(void)
{
    errorea=(konsigna-abiadura)/256; //errorea bilatu eta normalizatu/256
    errorea2=(konsigna2-abiadura2)/256; //errorea bilatu eta normalizatu/256
    errorebatura=errorebatura+errorea;
    errorebatura2=errorebatura2+errorea2;
    P=errorea*0.049;
    P2=errorea2*0.039;
    I=errorebatura*0.060;
    I2=errorebatura2*0.058;
    irtera = P+I;
    irtera2= P2+I2;

    if (irtera>1)
    {
        irtera=1;
    }
    if (irtera<0)
    {
        irtera=0.1;
    }
    if (irtera2>1)
    {
        irtera2=1;
    }
    if (irtera2<0)
    {
        irtera2=0.1;
    }

    OC1RS=irtera*256;
    OC2RS=irtera2*256;

}
```

### 2.3 Distantzia neurtu

Ultra soinu sentsorearen kudeatzailea. Sentsorearen martxan jartzeko seinalea sortu eta eta harrapaketak egiteko IC2 modulua eta tenporizadorea kudeatzen du.

```
void distantzianeurtu(void)
{
//PORTUAK
TRISDbits.TRISD9=1;//RD9, IC2 pina sarrera bezala
TRISDbits.TRISD2=0;//RD2 pina irteera bezala

//IC2 modulua gaitzea etakonfigurazioa
IC2CON = 0x0000; //Reset
IC2CON=0x0081;//TMR2, inetrupzioa kaptura bakoitzeko,kaptura ertz gorakor zein berakor
_IC2IE = 1; //IC2 interruptzioagaitu
_IC2IP = 1; //level 1priority
_IC2IF = 0; //badaezpada banderatzoa jaitzi

//Balio ezberdinen hasieraketa
kap1=0;
kap2=0;
denbora=0;
i=0;
maiztasuna=0;
oztopo=0;
hurrunegi=0;
neurtuta=0;
kont=0; TMR2=0;
T2CON=0x8030;
PORTDbits.RD2=1;

while(kont<3)
{
kont=kont+1;
} PORTDbits.RD2=0;
luze=TMR2;
muga=luze+368;

while ((neurtuta==0)&&(hurrunegi==0))
{

if(TMR2>muga)
{
hurrunegi=1;
distantzia=400;
}
//itzaro
}
}
```

## 2.4 Behaketa

Serbomotorearen kontrola. 0° eta 180° bitartean bost posizio hartu eta bakoitzean 2.3 Distantzia\_neurtu programa exekutatzen du. Distantzia luzeena eta angeluaren erregistroa. PWM seinalearen kontrola.

```

void behaketa (void)
{
//TMR5 konfigurazio
TMR5=0;
T5CON=0x8030; //Fosc/2 ,1:8
PR5 = 10000;// PWM 3mskoT
_T5IE=1;
_T5IF=0;

//PWM gaitzea eta TMR3
OC5CON = 0x0000; // OC4 modulua reseteatu
OC5R = 100; // Registroahasieratu
OC5RS=100;//Bigarren konparaketa erregistroa hasieratu
OC5CON=0x000e;//konfigurazioa, timer3,
TMR3=0;
T3CON=0x8010; //Fosc/2,1:8
PR3=1500; // PWM 3ms
itzaron=0;
while(itzaron==0)//TMR5
{
}
disluzea=0;
distantzia=0;
p=0;
angelua=0;
while(angelua<=180)
{
distantzianeurtu();
TMR3=0;
T3CON=0x8010; //Fosc/2,1:8
PR3=1500; // PWM 3ms
if(distantzia>disluzea)
{
disluzea=distantzia;
norabide=angelua;
}
angelua=angelua +45;
if (angelua<=180)
{
OC5RS=OC5RS+215;
}
itzaron=0;
while(itzaron==0) //Birtuarteitzaron
{
}
itzaron=0;
}
OC5RS=528;
while(itzaron==0)//TMR1
{
}
itzaron=0;
OC5CON=0x0000;
T5CON=0x0000;
}

```

## 2.5 Posizio1

Konpas digitalaren konfigurazioa.

```

void posizio1 (void)
{
  b_angelua=0;
  //I2C konfigurazioa
  I2C1BRG=0x13; // Fcy=4MFscl=100khz
  I2C1CON=0x8200;
  _MI2C1IE=1; //etenduragaitu
  _MI2C1IF=0; // banderatzozajaitzi

  while(hasieratu!=10)
  {
    hasieratu=hasieratu+1;
  }
  //////////////////////////////////////
  //I2C modua
  _SEN=1; // START
  while(I2C1CONbits.SEN==1)//ITZARON
  {
  }
  I2C1TRN=0x3C; //IDATZI
  while(_TBF==1)// ESKLABUAREN ERANTZUNARI ITZARON
  {
  }
  while(_TRSTAT==1)
  {
  }
  while(_TRSTAT==1)
  {
  }
  I2C1TRN=0x00; //ERREGISTROA
  while(_TBF==1)// ESKLABUAREN ERANTZUNARI ITZARON
  {
  }
  while(_TRSTAT==1)
  {
  }
  I2C1TRN=0x78; // AERREGISTROA
  while(_TBF==1)// ESKLABUAREN ERANTZUNARI ITZARON
  {
  }
  while(_TRSTAT==1)
  {
  }
  I2C1TRN=0x20; // BERREGISTROA
  while(_TBF==1)// ESKLABUAREN ERANTZUNARI ITZARON
  {
  }
  while(_TRSTAT==1)
  {
  }
  I2C1TRN=0x00; // ModuERREGISTROA
  while(_TBF==1)// ESKLABUAREN ERANTZUNARI ITZARON
  {
  }
  while(_TRSTAT==1)
  {
  }
  _PEN=1; //STOP
  while(I2C1CONbits.PEN==1)//ITZARON
  {
  }
}

```



## 2.6 Posizio2

Konpas digitalaren kontrola. I2C bidezko komunikazio protokoloa burutu, konpas eta mikrokontrolagailuaren artean. Ardatz cartesianetako datuak jaso eta bektore bihurtu.

```

void posizio2 (void)
{
  _SEN=1; // START
  while(I2C1CONbits.SEN==1)//ITZARON
  {
  }
  I2C1TRN=0x3C; //IDATZI
  while(_TBF==1)//ESKLABUAREN ERANTZUNARI ITZARON
  {
  }
  while(_TRSTAT==1)
  {
  }
  I2C1TRN=0x03; //ERREGISTROA
  while(_TBF==1)//ESKLABUARI ITZARON
  {
  }
  while(_TRSTAT==1)
  {
  }
  _RSEN=1; // START
  while(I2C1CONbits.RSEN==1)//ITZARON
  {
  }
  I2C1TRN=0x3D; //IDATZI
  while(_TBF==1)//ESKLABUARI ITZARON
  {
  }
  while(_TRSTAT==1)
  {
  }
  _RCEN=1; // jasotze moduagaitu
  while(I2C1CONbits.RCEN==1)
  {
  }
  while(_RBF==0)
  {
  }
  x_h=I2C1RCV;
  _ACKDT = 0; //01 DATUAK BIDALTZEN
  _ACKEN = 1;
  while(_ACKEN==1)
  {
  }
  _RCEN=1; // jasotze moduagaitu
  while(I2C1CONbits.RCEN==1)
  {
  }
  while(_RBF==0)
  {
  }
  x_l=I2C1RCV;
  _ACKDT = 0; //01 DATUAK BIDALTZEN JARRAITU
  _ACKEN = 1;
  while(_ACKEN==1)
  {
  }
  _RCEN=1; // jasotze moduagaitu
  while(I2C1CONbits.RCEN==1)
  {
  }
  while(_RBF==0)
  {
  }
  z_h=I2C1RCV;
  _ACKDT = 0; //01 DATUAK BIDALTZEN
  JARRAITU 00 EZ BIDALI GEHIAGO
  _ACKEN = 1;
  while(_ACKEN==1){
  }
  _RCEN=1; // jasotze moduagaitu
  while(I2C1CONbits.RCEN==1)
  {
  }
  while(_RBF==0)
  {
  }
  z_l=I2C1RCV;
  _ACKDT = 0; //01 DATUAK BIDALTZEN
  JARRAITU 00 EZ BIDALI GEHIAGO
  _ACKEN = 1;
  while(_ACKEN==1)
  {
  }
  _RCEN=1; // jasotze moduagaitu
  while(I2C1CONbits.RCEN==1)
  {
  }
  while(_RBF==0)
  {
  }
  y_h=I2C1RCV;
  _ACKDT = 0; //01 DATUAK BIDALTZEN
  JARRAITU 00 EZ BIDALI GEHIAGO
  _ACKEN = 1;
  while(_ACKEN==1)
  {
  }
  _RCEN=1; // jasotze moduagaitu
  while(I2C1CONbits.RCEN==1)
  {
  }
  while(_RBF==0)
  {
  }
  y_l=I2C1RCV;
  _ACKDT = 1; //01 DATUAK BIDALTZEN
  JARRAITU 11 EZ BIDALI GEHIAGO
  _ACKEN = 1;
  while(_ACKEN==1)
  {
  }
  X = x_h * 256 +x_l;
  Y = y_h * 256 +y_l;
  X=X-119;// offset Y=Y+218;// offset
  posizio= atan2((double)Y,(double)X)* (180 /
  3.14159265) ; // angle in degrees

  if (posizio<0)
  {
  posizio=posizio+360;
  }
  nago=posizio;

  _PEN=1; //STOP
  while(I2C1CONbits.PEN==1)//ITZARON
  {
  }
}

```

## 2.7 Biratu

Biraketak kudeatzeko programa. Biraketa zentzua eta motorren kontrola. Bukaerako posizioaren kalkulua eta konpas digitalarekin komunikazioa, posizio2 programa exekutatu.

```

void biratu (void)
{
  if(biraketa==0)
  {
    norabide=norabide;
    b_angelua=90-norabide;
    buka_posizio = posizio + b_angelua;
  }
  if(biraketa==1)
  {
    if(eskuma==1)
    {
      b_angelua=biraketa2;
      buka_posizio=posizio+ b_angelua;
      eskuma=0;
    }
    if(eskerra==1)
    {
      b_angelua=-biraketa2;
      buka_posizio=posizio+b_angelua;
      eskerra=0;
    }
    biraketa=0;
  }

  if(buka_posizio>360)
  {
    buka_posizio=buka_posizio-360;
  }
  if(buka_posizio<0)
  {
    buka_posizio=buka_posizio+360;
  }
  //TIMER 3-ren gaitzea etakonfigurazioa
  PR3 = 313; //PWM 200 Hz
  T3CON = 0x8020;// 16 biteko kontaketa, barne erlojua Fosc/2, 1:64 preskalatua

  if(b_angelua<0)
  {
    konsigna=50;
    konsigna2=50;
    PORTDbits.RD6=1;//DIR2
    PORTAbits.RA9=1;//DIR1
    eskuma=1;
  }
}

```



```


if(b_angelua>0)
{
  konsigna=50;
  konsigna2=50;
  PORTDbits.RD6=0;//DIR2
  PORTAbits.RA9=0;//DIR1
  eskerra=1;
}
_T1IE=1; //PID laginketagaitzea
while((nago<buka_posizio-1)||
(nago>buka_posizio+1))
{
  posizio2();
}
gelditu();
biraketa=1;
Dis_egina=0;
PORTDbits.RD3=0;//DIR2
PORTAbits.RA9=1;//DIR1
}

```



## 2.8 Bidebarri

Oztopoak ekiditeko eta helmuga puntura irizteko biraketa angeluen kudeaketa.

<pre> void bidebarri(void) {   B=saihesbide1;    if ((b_angelua==45)  (b_angelua== -45))   {     AB= (b_angelua*3.1416)/180;//radianes     C2= (A*A)+(B*B)-(2*A*B*cos(AB));     C=sqrt(C2);     Dis_helmuga=C;     Dis_egina=0;      BC=((A*A)-(B*B)-(C*C))/(-2*B*C);     BC_angelua=acos(BC);     BC_angelua=(BC_angelua*180)/3.1416;     biraketa2=180-BC_angelua;     biraketa=1;   } } </pre>	 <pre> if ((b_angelua==90)  (b_angelua== -90)) {   C2= (A*A)+(B*B);   C=sqrt(C2);   arg=A/B;   BC_angelua=atan(arg);   BC_angelua=(BC_angelua*180)/3.1416;   biraketa2=180-BC_angelua;   biraketa=1; } } </pre>
---	--

## 2.9 Gelditu

Motorren gelditze kontrola eta PID balioen reseta.

```

void gelditu (void)
{
  konsigna=0;
  konsigna2=0;
  P=0;
  I=0;
  errorea=0;
  errorebatura=0;
  irtera=0;
  aurre_irtera=0;
  P2=0;
  I2=0;
  errorea2=0;
  errorebatura2=0;
  irtera2=0;
  aurre_irtera2=0;
}

```

### 3 Etenduren kudeaketa

#### 3.1 Tenporizadoreak

Tenporidazoreen kontaketa burutua Berrasieratzea eta flag-en sorketa.

```

void attribute ((interrupt,no_auto_psv)) _T2Interrupt(void)
{
  hurrunegi=1;
  oztopo=0; //ez dagooztoporik
  _T2IF=0; kap2=IC2BUF;
}
//*****

void attribute ((interrupt,no_auto_psv)) _T1Interrupt(void)
{
  IFS0bits.T1IF=0;
  laginketa=1;
  TMR1=0;
  PID();
}
//*****

void attribute ((interrupt,no_auto_psv)) _T5Interrupt(void)
{
  _T5IF=0;
  itzaron=1;
  TMR5=0;
}

```

#### 3.2 IC Moduluak

IC1 eta IC4 kodetzaileen pultsuen kudeaketa. Pultsuak jaso, abiadura eta desplazamendua kalkulatu.

```

void attribute ((interrupt,no_auto_psv)) _IC1Interrupt(void)
{
  if (a==0)
  {
    denbora1=IC1BUF;
  }
  if (a==1)
  {
    denbora2=IC1BUF;
    denbora = (denbora2-denbora1);
    maiztasuna=1/(denbora*0.000064);
    abiadura=((maiztasuna/53)/3)*60; //Abiadura rpm 3pultzu 5 3reductora
    IFS0bits.IC1IF = 0; //Flagajaitzi
  }
  denbora1=denbora2; a=1;
  Dis_egina=((0.128)+ Dis_egina); //interrupzio bakoitzean arkuak gehitu 2 pi r/3 zati 53
  saihsbide=0.128+saihsbide;
}

```

## IC2 modulua

Ultra soinu sentsorearen pultsuen kudeaketa. Pultsuen arteko denbora neurketa eta distantzia kalkulua.

```
void attribute ((interrupt,no_auto_psv)) _IC2Interrupt(void)
{
  if (i==0)
  {
    kap1=IC2BUF;
  }
  if (i==1)
  {
    kap2=IC2BUF;
    if(kap2<kap1)
    {
      kap2=kap2+65536;
    }
    denboraa=kap2-kap1;
    distantzia=(denboraa*0.000064*340*100)/2; //256/4x10e6
    distantzia=distantzia*1.05;//
    neurtuta=1;
    if (distantzia<=50)
    {
      oztopo=1; // oztopoa dago beaketansartu
    }
    else
    {
      oztopo=0; //ez dagooztoporik
    }
  }
  i=1;
  _IC2IF = 0;//banderatzoa jaitzi
  _IC2IE = 1; //etenduragaitu
}
```