



INDUSTRIA ELEKTRONIKOAREN ETA AUTOMATIKAREN INGENERITZAKO GRADUA

GRADU AMAIERAKO LANA

2015 / 2016

IKUSMEN ARTIFIZIALAREN APLIKAZIOA: PIEZA-SAILKATZAILE ROBOTIKOA

Eranskinak

IKASLEAREN DATUAK

IZENA Jon
 ABIZENAK Bilbao Etxebarria
 NAN zk 45751947W

Sinadura

DATA 2016-06-16

ZUZENDARIAREN DATUAK

IZENA Raquel
 ABIZENAK Martinez Rodriguez
 SAILA Sistemen Ingeneritza eta Automatika

Sinadura

DATA 2016-06-16

AURKIBIDEA

1. PROGRAMAZIOA

1.1. MATLAB-EN ERABILITAKO FUNTZIOAK...2

1.2. ROBOCELL.....23

1. PROGRAMAZIOA

1.1 MATLAB-en erabilitako funtzioak

- KAMERA KONEKTATU:

```
vid=videoinput('winvideo',1,'YUY2_640x480');
erresoluzioa=get(vid,'VideoResolution');
altuera=erresoluzioa(1);
zabalera=erresoluzioa(2);
ZenBanda=get(vid,'NumberofBands');
hImage=image(zeros(zabalera,altuera,ZenBanda),'Parent',handles.Bideo);
preview(vid,hImage);
```

- ARDUINO KONEKTATU:

```
[ a ] = Arduino( );
```

Funtzio hau erabiltzen da Arduinoa konektatzeko.

```
function [ a ] = Arduino( )
    a=arduino('COM3');
    a.configureDigitalPin(2,'output');
    a.configureDigitalPin(4,'output');
    a.configureDigitalPin(6,'output');
    a.configureDigitalPin(7,'output');
    a.configureDigitalPin(8,'output');
    a.configureDigitalPin(9,'output');
    a.configureDigitalPin(10,'output');
    a.configureDigitalPin(11,'input');
    a.configureDigitalPin(3,'output');
    a.configureDigitalPin(5,'output');
    %% balorea emateko
    a.writeDigitalPin(2,1);
    a.writeDigitalPin(4,1);
    a.writeDigitalPin(3,0);
    a.writeDigitalPin(6,1);
    a.writeDigitalPin(5,0);
    a.writeDigitalPin(7,1);
    a.writeDigitalPin(8,1);
    a.writeDigitalPin(9,1);
    a.writeDigitalPin(10,1);
end
```

- ARGAZKIA HARTZEKO:

```
Irudia=getsnapshot(vid);% argazkia atara
```

- PIEZAK DETKTATZEKO :

Funtzio hau erabiltzen da:

```
[ Pieza_kop,L ] = detektatu( Irudia );
```

```
function [ Pieza_kop,L ] = detektatu( Irudia )

Irudi_gris=rgb2gray(Irudia);%rgb tan dagoen irudia grisetara pasatzen da
[m,n]=size(Irudi_gris);%grisean dagoen irudiaren neurriak hartzen dira

Irudi_binario=ones(m,n);%irudi binarioari griseko neurriak jartzen jakoz
%Irudi binariora pasatzen da
umbral=72;

for i = 1:m
    for j = 1:n
        if (Irudi_gris(i,j)>umbral)
            Irudi_binario(i,j)=0;
        else
            Irudi_binario(i,j)=1;
        end
    end
end

Irudi_binario=bwareaopen(Irudi_binario,3000);%irudi binarioan partikula txikiak kentzen dira
[L, Pieza_kop]=bwlabel(Irudi_binario,8);%irudi binarioiko pixelak batzen dira, pieza kopurua emateko.

end
```

- KOLOREA DETKTATZEKO

```
[ Zentro_Gorri, Zentro_Berde, Zentro_Urdin, Zenb_Gorri, Zenb_Urdin, Zenb_Berde ] = Kolorea( Irudia );
```

```
function [ Zentro_Gorri, Zentro_Berde, Zentro_Urdin, Zenb_Gorri, Zenb_Urdin, Zenb_Berde ] = Kolorea( Irudia )

gorri=imsubtract(Irudia(:,:,1),rgb2gray(Irudia));%irudi grisei konponente gorria kentzen zaio, gorria hartzeko.
gorri=medfilt2(gorri,[15,15]);% filtro bat erabiltzen dugu
gorri=im2bw(gorri,0.26);% griseko irudia binariotara pasatzen dugu
gorri=bwareaopen(gorri,300);%300 baino txikiagoko pixelak kendu

berde=imsubtract(Irudia(:,:,2),rgb2gray(Irudia));
berde=medfilt2(berde,[15,15]);
berde=im2bw(berde,0.03);
berde=bwareaopen(berde,30000);

urdin=imsubtract(Irudia(:,:,3),rgb2gray(Irudia));
urdin=medfilt2(urdin,[9,9]);
urdin=im2bw(urdin,0.18);
urdin=bwareaopen(urdin,500);
%koloretako zenbet pieza dauden jakingo da
[LGorri, Zenb_Gorri]=bwlabel(gorri,8);
[LBerde, Zenb_Berde]=bwlabel(berde,8);
[LUrdin, Zenb_Urdin]=bwlabel(urdin,8);
```

```
    if Zenb_Gorri>0
        Stats_Gorri=regionprops(LGorri,'Centroid');% propietateak berenganatu
    ]
        for ObjectGorri=1:length(Stats_Gorri)
            Zentro_Gorri=Stats_Gorri(ObjectGorri).Centroid;% zentroidean balioak
        end
        Zentro_Gorri=[Stats_Gorri.Centroid];
    else
        Zentro_Gorri=[];
    end
    if Zenb_Berde>0
        Stats_Berde=regionprops(LBerde,'Centroid');
    ]
        for ObjectBerde=1:length(Stats_Berde)
            Zentro_Berde=Stats_Berde(ObjectBerde).Centroid;
        end
        Zentro_Berde=[Stats_Berde.Centroid];
    else
        Zentro_Berde=[];
    end

    if Zenb_Urdin>0
        Stats_Urdin=regionprops(LUrdin,'Centroid');

        for ObjectUrdin=1:length(Stats_Urdin)
            Zentro_Urdin=Stats_Urdin(ObjectUrdin).Centroid;
        end
        Zentro_Urdin=[Stats_Urdin.Centroid];
    else
        Zentro_Urdin=[];
    end
end
```

• POSIZIOA JAKITEKO (koloretan)

```
function [ Posizio_Gorri , Posizio_Berde , Posizio_Urdin, Pg , Pb , Pu ] = Posizioa_aurkitu( Zenb_Gorri , Zentro_Gorri , Zenb_Berde , Zentro_Berde , Zenb_Urdi
% aldagaiak
x=0;
g=0;
z=0;
Pg=[0 0 0 0 0 0 0 0 0];
Pb=[0 0 0 0 0 0 0 0 0];
Pu=[0 0 0 0 0 0 0 0 0];

% Barneko taula
if kol1==1
    x=1;
end
if kol1==2
    g=1;
end
if kol1==3
    z=1;
end
if kol2==1
    x=2;
end
if kol2==2
    g=2;
end
if kol2==3
    z=2;
end

% Gorri
if Zentro_Gorri>0

    j=1;

]    for i=1:2:Zenb_Gorri*2

        if (Zentro_Gorri(i) < 213.3333333) && (Zentro_Gorri(i+1)<160)

            Posizio_Gorri(j)=1;
            Pg(1)=x;

        end

        if (Zentro_Gorri(i)<213.333333) && (160 < Zentro_Gorri(i+1))&& (Zentro_Gorri(i+1)< 320)

            Posizio_Gorri(j)=2;
            Pg(2)=x;

        end

        if (Zentro_Gorri(i)<213.33333) && (Zentro_Gorri(i+1)>320)

            Posizio_Gorri(j)=3;
            Pg(3)=x;

        end

    end
```

```
if (213.333333<Zentro_Gorri(i))&&(Zentro_Gorri(i)<426.66666) && (Zentro_Gorri(i+1)<160)

    Posizio_Gorri(j)=4;
    Pg(4)=x;

end

if (213.3333<Zentro_Gorri(i))&&(Zentro_Gorri(i)<426.66666) && (160<Zentro_Gorri(i+1))&&(Zentro_Gorri(i+1)<320)

    Posizio_Gorri(j)=5;
    Pg(5)=x;

end

if (213.333<Zentro_Gorri(i))&&(Zentro_Gorri(i)<426.66666) && (Zentro_Gorri(i+1)>320)

    Posizio_Gorri(j)=6;
    Pg(6)=x;

end

if (Zentro_Gorri(i)>426.66666) && (Zentro_Gorri(i+1)< 160)

    Posizio_Gorri(j)=7;
    Pg(7)=x;

end

if (Zentro_Gorri(i)>426.66666) && (160<Zentro_Gorri(i+1))&&(Zentro_Gorri(i+1)<320)

    Posizio_Gorri(j)=8;
    Pg(8)=x;

end

if (Zentro_Gorri(i)>426.66666) && (Zentro_Gorri(i+1)>320)

    Posizio_Gorri(j)=9;
    Pg(9)=x;

end

j=j+1;

end

else
    Posizio_Gorri=[];
end
```

Beste koloreekin berdina da.

• FORMAK JAKITEKO

```
function [Zirkulo,Karratu,Triangelu,Zentro_Zirkulo,Zentro_Karratu,Zentro_Triangelu ] = formak( Pieza_kop,L )

if Pieza_kop>0
    i=1;
    while i<=Pieza_kop
        Stats_Pieza=regionprops(L,'Centroid','Extent');
        Centro=[Stats_Pieza.Centroid];
        Extent=[Stats_Pieza.Extent];
        i=i+1;
    end
end
```

Honek forma horien zentroak jakiteko egiten da.

```
while a<=Pieza_kop
    Zirkulo(a)=0;
    Karratu(a)=0;
    Triangelu(a)=0;

    if (0.51<Extent(a)) && (Extent(a)<0.8)
        Zirkulo(a)=1;
        Zentro_Zirkulo(e)=Centro(b);
        Zentro_Zirkulo(e+1)=Centro(b+1);
        e=e+2;
        Karratu(a)=0;
        Triangelu(a)=0;
    end

    if (0.8<Extent(a)) && (Extent(a)<1.1)
        Zirkulo(a)=0;
        Zentro_Karratu(c)=Centro(b);
        Zentro_Karratu(c+1)=Centro(b+1);
        c=c+2;
        Karratu(a)=1;
        Triangelu(a)=0;
    end

    if (0.2<Extent(a)) && (Extent(a)<0.51)
        Zirkulo(a)=0;
        Karratu(a)=0;
        Triangelu(a)=1;
        Zentro_Triangelu(d)=Centro(b);
        Zentro_Triangelu(d+1)=Centro(b+1);
        d=d+2;
    end
end
```



```
                end
                b=b+2;
                a=a+1;
            end
        else
            Zirkulo=0;
            Karratu=0;
            Triangelu=0;
            Zentro_Zirkulo=0;
            Zentro_Karratu=0;
            Zentro_Triangelu=0;
        end
    end
end
```

• POSIZIOA JAKITEKO (formetan)

```
function [Pz,Pk,Pt,Posizio_Zirkulo,Posizio_Karratu,Posizio_Triangelu] = Posizioa_surkitui(Zirkulo,Karratu,Triangelu,Zentro_Zirkulo,Zentro_Karratu,Zentro_Triangelu)

% aldagaiak
x=0;
g=0;
z=0;
Pz=[0,0,0,0,0,0,0,0,0,0];
Pk=[0,0,0,0,0,0,0,0,0,0];
Pt=[0,0,0,0,0,0,0,0,0,0];
zir=sum(Zirkulo);
kar=sum(Karratu);
tri=sum(Triangelu);

% Barneko taula
if form1==1
    x=1;
end
if form1==2
    g=1;
end
if form1==3
    z=1;
end
if form2==1
    x=2;
end
if form2==2
    g=2;
end
if form2==3
    z=2;
end
end
```

```
% zirkulo
if zir>0

    j=1;

    for i=1:2:zir*2

        if (Zentro_Zirkulo(i) < 213.3333333) && (Zentro_Zirkulo(i+1)<160)

            Posizio_Zirkulo(j)=1;
            Pz(1)=x;

        end

        if (Zentro_Zirkulo(i)<213.333333) && (160 < Zentro_Zirkulo(i+1))&& (Zentro_Zirkulo(i+1)< 320)

            Posizio_Zirkulo(j)=2;
            Pz(2)=x;

        end

        if (Zentro_Zirkulo(i)<213.33333) && (Zentro_Zirkulo(i+1)>320)

            Posizio_Zirkulo(j)=3;
            Pz(3)=x;

        end

        if (213.33333<Zentro_Zirkulo(i))&&(Zentro_Zirkulo(i)<426.66666) && (Zentro_Zirkulo(i+1)<160)

            Posizio_Zirkulo(j)=4;
            Pz(4)=x;

        end

        if (213.3333<Zentro_Zirkulo(i))&&(Zentro_Zirkulo(i)<426.66666) && (160<Zentro_Zirkulo(i+1))&&(Zentro_Zirkulo(i+1)<320)

            Posizio_Zirkulo(j)=5;
            Pz(5)=x;

        end

        if (213.333<Zentro_Zirkulo(i))&&(Zentro_Zirkulo(i)<426.66666) && (Zentro_Zirkulo(i+1)>320)

            Posizio_Zirkulo(j)=6;
            Pz(6)=x;

        end

        if (Zentro_Zirkulo(i)>426.66666) && (Zentro_Zirkulo(i+1)< 160)

            Posizio_Zirkulo(j)=7;
            Pz(7)=x;

        end

        if (Zentro_Zirkulo(i)>426.66666) && (160<Zentro_Zirkulo(i+1))&&(Zentro_Zirkulo(i+1)<320)

            Posizio_Zirkulo(j)=8;
            Pz(8)=x;

        end

    end

end
```

```

        if (Zentro_Zirkulo(i)>426.66666) && (Zentro_Zirkulo(i+1)>320)

            Posizio_Zirkulo(j)=9;
            Pz(9)=x;

        end

        j=j+1;

    end

else
    Posizio_Zirkulo=[];
    Pz=[0,0,0,0,0,0,0,0,0,0];
end

```

- BARNE TAULA

Posizioen laguntzarekin barne taula bat egiten da mahi jokoa simulatzeko.

```

Tabla=[0,0,0; 0,0,0; 0,0,0];
A=[Pg(1),Pg(4),Pg(7); Pg(2),Pg(5),Pg(8); Pg(3),Pg(6),Pg(9)];
B=[Pu(1),Pu(4),Pu(7); Pu(2),Pu(5),Pu(8); Pu(3),Pu(6),Pu(9)];
C=[Pb(1),Pb(4),Pb(7); Pb(2),Pb(5),Pb(8); Pb(3),Pb(6),Pb(9)];
Tabla=A+B+C;

```

- ERROBOTAREN AUKERA

```

function [Fila, Kolumna ]= Makina(Tabla,jug)

%Proba batzuk egiten dira. Proba batek ondo ematen badu nehikoa da.
if jug==2

    calcula_posicion=0;
    while calcula_posicion==0
        proba=1;

        %lehenego beti ikusten du erdiko posizioa libre dagoen.
        if proba==1
            if Tabla(2,2)==0
                Fila=2;
                Kolumna=2;
                proba=0;
                calcula_posicion=1;
            else
                proba=2;
            end
        end
    end
end

```

```
%bigarren proban begiratzeko du irabazi ahal duen, horrela  
%ez bada begiratzeko du aurkariak irabazi dezaken.
```

```
if proba==2  
    Fila=0;  
    Kolumna=0;  
    if Tabla(2,2)==jug  
        if Tabla(1,2)==jug  
            filapr=3;  
            columnapr=2;  
            if Tabla(filapr,columnapr)==0  
                Fila=3;  
                Kolumna=2;  
            end  
        end  
    end  
    if Tabla(3,2)==jug  
        filapr=1;  
        columnapr=2;  
        if Tabla(filapr,columnapr)==0  
            Fila=filapr;  
            Kolumna=columnapr;  
        end  
    end  
    if Tabla(2,1)==jug  
        filapr=2;  
        columnapr=3;  
        if Tabla(filapr,columnapr)==0  
            Fila=filapr;  
            Kolumna=columnapr;  
        end  
    end  
end
```

```
if Tabla(2,3)==jug
    filapr=2;
    columnapr=1;
    if Tabla(filapr,columnapr)==0
        Fila=filapr;
        Kolumna=columnapr;
    end
end
if Tabla(1,1)==jug
    filapr=3;
    columnapr=3;
    if Tabla(filapr,columnapr)==0
        Fila=filapr;
        Kolumna=columnapr;
    end
end
if Tabla(3,3)==jug
    filapr=1;
    columnapr=1;
    if Tabla(filapr,columnapr)==0
        Fila=filapr;
        Kolumna=columnapr;
    end
end
```

```
if Tabla(3,1)==jug
    filapr=1;
    columnapr=3;
    if Tabla(filapr,columnapr)==0
        Fila=filapr;
        Kolumna=columnapr;
    end
end
if Tabla(1,3)==jug
    filapr=3;
    columnapr=1;
    if Tabla(filapr,columnapr)==0
        Fila=filapr;
        Kolumna=columnapr;
    end
end
end
if Tabla(1,1)==jug
    if Tabla(1,2)==jug
        filapr=1;
        columnapr=3;
        if Tabla(filapr,columnapr)==0
            Fila=filapr;
            Kolumna=columnapr;
        end
    end
end
```

```
if Tabla(1,3)==jug
    filapr=1;
    columnapr=2;
    if Tabla(filapr,columnapr)==0
        Fila=filapr;
        Kolumna=columnapr;
    end
end
if Tabla(2,1)==jug
    filapr=3;
    columnapr=1;
    if Tabla(filapr,columnapr)==0
        Fila=filapr;
        Kolumna=columnapr;
    end
end
if Tabla(3,1)==jug
    filapr=2;
    columnapr=1;
    if Tabla(filapr,columnapr)==0
        Fila=filapr;
        Kolumna=columnapr;
    end
end
end
end
```

```
if Tabla(3,3)==jug
    if Tabla(3,2)==jug
        filapr=3;
        columnapr=1;
        if Tabla(filapr,columnapr)==0
            Fila=filapr;
            Kolumna=columnapr;
        end
    end
end
if Tabla(3,1)==jug
    filapr=3;
    columnapr=2;
    if Tabla(filapr,columnapr)==0
        Fila=filapr;
        Kolumna=columnapr;
    end
end
if Tabla(2,3)==jug
    filapr=1;
    columnapr=3;
    if Tabla(filapr,columnapr)==0
        Fila=filapr;
        Kolumna=columnapr;
    end
end
end
```



```
    if Tabla(1,3)==jug
        filapr=2;
        columnapr=3;
        if Tabla(filapr,columnapr)==0
            Fila=filapr;
            Kolumna=columnapr;
        end
    end
end
if Tabla(1,3)==jug
    if Tabla(1,2)==jug
        filapr=1;
        columnapr=1;
        if Tabla(filapr,columnapr)==0
            Fila=filapr;
            Kolumna=columnapr;
        end
    end
end
if Tabla(2,3)==jug
    filapr=3;
    columnapr=3;
    if Tabla(filapr,columnapr)==0
        Fila=filapr;
        Kolumna=columnapr;
    end
end
end
end
```

```
    if Tabla(2,3)==jug
        filapr=3;
        columnapr=3;
        if Tabla(filapr,columnapr)==0
            Fila=filapr;
            Kolumna=columnapr;
        end
    end
end
if Tabla(3,1)==jug
    if Tabla(2,1)==jug
        filapr=1;
        columnapr=1;
        if Tabla(filapr,columnapr)==0
            Fila=filapr;
            Kolumna=columnapr;
        end
    end
end
if Tabla(3,2)==jug
    filapr=3;
    columnapr=3;
    if Tabla(filapr,columnapr)==0
        Fila=filapr;
        Kolumna=columnapr;
    end
end
end
end
```

```
%  
if Fila~=0  
    proba=0;  
    calcula_posicion=1;  
    jug=2;  
else  
    if jug==2  
        jug=1;  
        proba=2;  
    else  
        jug=2;  
        proba=3;  
    end  
end  
end
```

```
%  
if Fila ~=0  
    if Tabla(Fila,Kolumna)~=0  
        proba=3;  
        calcula_posicion=0;  
    end  
end
```

```
end  
  
%aleatorioki posizioak aukeratzen ditu huts bat topatu arte  
if proba==3  
if Fila==0  
    Fila=2;  
    Kolumna=2;  
end  
  
while Tabla(Fila,Kolumna)~=0  
    Fila = (fix(rand(1)*2.99))+1;  
    Kolumna = (fix(rand(1)*2.99))+1;  
end  
    calcula_posicion=1;  
    proba=4;  
end  
end  
end
```

- AUKERA ZUZENA

```
function [ zuzena ] = Komprobaketa( Fila, Kolumna, Tabla )  
  
    if Tabla(Fila,Kolumna)==0  
  
        zuzena=0;  
    else  
  
        zuzena=1;  
    end  
end
```

Tableru simulatuan begiratzten du posizio hori hutsik badago.

- IRABAZLERIK BADAGOEN JAKITEA

```
function [amaitu]=Amaitu(Tabla, kontagailua, jug, amaitu)  
  
if amaitu==0  
    %BEGIRATZEN DU PIEZA JARRI DUENAK IRABAZI BADU EDO EZ.  
    if Tabla(2,2)==jug  
        if Tabla(1,2)==jug  
            if Tabla(3,2)==jug  
                amaitu=jug;  
            end  
        end  
    end  
    if Tabla(2,1)==jug  
        if Tabla(2,3)==jug  
            amaitu=jug;  
        end  
    end  
    if Tabla(1,1)==jug  
        if Tabla(3,3)==jug  
            amaitu=jug;  
        end  
    end  
    if Tabla(3,1)==jug  
        if Tabla(1,3)==jug  
            amaitu=jug;  
        end  
    end  
end  
if Tabla(1,1)==jug  
    if Tabla(1,2)==jug  
        if Tabla(1,3)==jug  
            amaitu=jug;  
        end  
    end  
    if Tabla(2,1)==jug  
        if Tabla(3,1)==jug  
            amaitu=jug;  
        end  
    end  
end
```

```
        end
    end
    if Tabla(3,3)==jug
        if Tabla(3,2)==jug
            if Tabla(3,1)==jug
                amaitu=jug;
            end
        end
        if Tabla(2,3)==jug
            if Tabla(1,3)==jug
                amaitu=jug;
            end
        end
    end
    end

    %TABLERUA BETETZEN DENEAN
    if kontagailua>=9 & amaitu==0
        amaitu=3;
    end
else
    amaitu=0;
end
end
```

• BIDALKETAK

```
function [a]=Bidalketa(Fila, Kolumna,a)

if Fila==1 && Kolumna==1
    a.writeDigitalPin(2,0);
    while a.readDigitalPin(11)==0

    end
    a.writeDigitalPin(2,1);
end

if Fila==2 && Kolumna==1
    a.writeDigitalPin(4,0);
    while a.readDigitalPin(11)==0

    end
    a.writeDigitalPin(4,1);
end

if Fila==3 && Kolumna==1
    a.writeDigitalPin(3,1);
    while a.readDigitalPin(11)==0

    end
    a.writeDigitalPin(3,0);
end

if Fila==1 && Kolumna==2
    a.writeDigitalPin(6,0);
    while a.readDigitalPin(11)==0

    end
    a.writeDigitalPin(6,1);
end
```

```
if Fila==2 && Kolumna==2
    a.writeDigitalPin(5,1);
    while a.readDigitalPin(11)==0

        end
a.writeDigitalPin(5,0);
end
if Fila==3 && Kolumna==2
    a.writeDigitalPin(7,0);
    while a.readDigitalPin(11)==0

        end
a.writeDigitalPin(7,1);
end
if Fila==1 && Kolumna==3
    a.writeDigitalPin(8,0);
    while a.readDigitalPin(11)==0

        end
a.writeDigitalPin(8,1);
end
if Fila==2 && Kolumna==3
    a.writeDigitalPin(9,0);
    while a.readDigitalPin(11)==0

        end
a.writeDigitalPin(9,1);
end
if Fila==3 && Kolumna==3
    a.writeDigitalPin(10,0);
    while a.readDigitalPin(11)==0

        end
a.writeDigitalPin(10,1);
end
```

1.2 ROBOCELL

Programa nagusia:

1	Set Variable SALTO = 1
2	Set Variable KONT = 0
3	ZINTA:
4	Set Variable KONT = KONT + 1
5	Reset Timer
6	If KONT == 10 Jump to AMAIERA
7	Start Conveyor Axis 7 at Speed 20 (%) in Plus Direction
8	DETEKTATU:
9	Set Variable TIM to Timer
10	If TIM > 400 Jump to GELDITU
11	If Input 8 On Jump to GELDITU
12	Jump to DETEKTATU
13	GELDITU:
14	Stop Conveyor Axis 7
15	HASIERA:
16	If Input 1 On Call Subroutine POSIZIOA1
17	If Input 2 On Call Subroutine POSIZIOA2
18	If Input 3 On Call Subroutine POSIZIOA4
19	If Input 4 On Call Subroutine POSIZIOA6
20	If Input 5 On Call Subroutine POSIZIOA7
21	If Input 6 On Call Subroutine POSIZIOA8
22	If Input 7 On Call Subroutine POSIZIOA9
23	Set Variable P3 to Analog Input 2
24	If P3 > 1 Jump to POSIZIOA3
25	Set Variable P5 to Analog Input 3
26	If P5 > 1 Jump to POSIZIOA5
27	Set Variable TIM to Timer
28	If TIM > 500 Jump to AMAIERA
29	If SALTO > KONT Jump to ZINTA
30	Jump to HASIERA
31	POSIZIOA3:
32	Call Subroutine POSIZIOA3
33	Jump to ZINTA
34	POSIZIOA5:
35	Call Subroutine POSIZIOA5
36	Jump to ZINTA
36	Jump to ZINTA
37	AMAIERA:
38	Set Variable SALTO = 1
39	Set Variable KONT = 0
40	Set Variable TIM to Timer
41	If TIM > 2000 Jump to FIN
42	If Input 8 On Jump to PAUSE
43	Jump to AMAIERA
44	PAUSE:
45	Wait 10 (10ths of seconds)
46	Jump to ZINTA
47	FIN:
48	End

Subrutinak:

49	Set Subroutine POSIZIOA1
50	Open Gripper
51	Go to Position 24 Speed 50 (%)
52	Go to Position 23 Speed 15 (%)
53	Close Gripper
54	Go to Position 1 Speed 50 (%)
55	Go to Position 2 Speed 50 (%)
56	Go to Position 3 Speed 20 (%)
57	Go to Position 4 Speed 10 (%)
58	Jaw 38 (mm)
59	Go to Position 3 Speed 10 (%)
60	Go to Position 2 Speed 20 (%)
61	Go to Position 1 Speed 50 (%)
62	Turn On Output 3
63	Turn Off Output 3
64	Set Variable SALTO = SALTO + 1
65	Return from Subroutine
66	Set Subroutine POSIZIOA2
67	Open Gripper
68	Go to Position 24 Speed 50 (%)
69	Go to Position 23 Speed 10 (%)
70	Close Gripper
71	Go to Position 2 Speed 50 (%)
72	Go to Position 5 Speed 20 (%)
73	Go to Position 6 Speed 10 (%)
74	Jaw 38 (mm)
75	Go to Position 5 Speed 10 (%)
76	Go to Position 2 Speed 20 (%)
77	Go to Position 1 Speed 50 (%)
78	Turn On Output 3
79	Turn Off Output 3
80	Set Variable SALTO = SALTO + 1
81	Return from Subroutine

82	Set Subroutine POSIZIOA3
83	Open Gripper
84	Go to Position 24 Speed 50 (%)
85	Go to Position 23 Speed 10 (%)
86	Close Gripper
87	Go to Position 2 Speed 50 (%)
88	Go to Position 7 Speed 20 (%)
89	Go to Position 8 Speed 10 (%)
90	Jaw 38 (mm)
91	Go to Position 7 Speed 10 (%)
92	Go to Position 2 Speed 20 (%)
93	Go to Position 1 Speed 50 (%)
94	Turn On Output 3
95	Turn Off Output 3
96	Set Variable SALTO = SALTO + 1
97	Return from Subroutine
98	Set Subroutine POSIZIOA4
99	Open Gripper
100	Go to Position 24 Speed 50 (%)
101	Go to Position 23 Speed 10 (%)
102	Close Gripper
103	Go to Position 1 Speed 50 (%)
104	Go to Position 21 Speed 30 (%)
105	Go to Position 9 Speed 20 (%)
106	Go to Position 10 Speed 10 (%)
107	Jaw 38 (mm)
108	Go to Position 9 Speed 10 (%)
109	Go to Position 21 Speed 20 (%)
110	Go to Position 1 Speed 50 (%)
111	Turn On Output 3
112	Turn Off Output 3
113	Set Variable SALTO = SALTO + 1
114	Return from Subroutine

115	Set Subroutine POSIZIOA5
116	Open Gripper
117	Go to Position 24 Speed 50 (%)
118	Go to Position 23 Speed 10 (%)
119	Close Gripper
120	Go to Position 1 Speed 50 (%)
121	Go to Position 21 Speed 50 (%)
122	Go to Position 11 Speed 25 (%)
123	Go to Position 12 Speed 10 (%)
124	Jaw 38 (mm)
125	Go to Position 11 Speed 10 (%)
126	Go to Position 21 Speed 25 (%)
127	Go to Position 1 Speed 50 (%)
128	Turn On Output 3
129	Turn Off Output 3
130	Set Variable SALTO = SALTO + 1
131	Return from Subroutine
132	Set Subroutine POSIZIOA6
133	Open Gripper
134	Go to Position 24 Speed 50 (%)
135	Go to Position 23 Speed 10 (%)
136	Close Gripper
137	Go to Position 1 Speed 50 (%)
138	Go to Position 21 Speed 25 (%)
139	Go to Position 13 Speed 20 (%)
140	Go to Position 14 Speed 10 (%)
141	Jaw 38 (mm)
142	Go to Position 13 Speed 10 (%)
143	Go to Position 21 Speed 25 (%)
144	Go to Position 1 Speed 50 (%)
145	Turn On Output 3
146	Turn Off Output 3
147	Set Variable SALTO = SALTO + 1
148	Return from Subroutine

149 Set Subroutine POSIZIOA7
150 Open Gripper
151 Go to Position 24 Speed 50 (%)
152 Go to Position 23 Speed 10 (%)
153 Close Gripper
154 Go to Position 1 Speed 50 (%)
155 Go to Position 22 Speed 50 (%)
156 Go to Position 15 Speed 25 (%)
157 Go to Position 16 Speed 10 (%)
158 Jaw 38 (mm)
159 Go to Position 15 Speed 10 (%)
160 Go to Position 22 Speed 25 (%)
161 Go to Position 1 Speed 50 (%)
162 Turn On Output 3
163 Turn Off Output 3
164 Set Variable SALTO = SALTO + 1
165 Return from Subroutine
166 Set Subroutine POSIZIOA8
167 Open Gripper
168 Go to Position 24 Speed 50 (%)
169 Go to Position 23 Speed 10 (%)
170 Close Gripper
171 Go to Position 1 Speed 50 (%)
172 Go to Position 22 Speed 50 (%)
173 Go to Position 17 Speed 25 (%)
174 Go to Position 18 Speed 10 (%)
175 Jaw 38 (mm)
176 Go to Position 17 Speed 10 (%)
177 Go to Position 22 Speed 25 (%)
178 Go to Position 1 Speed 50 (%)
179 Turn On Output 3
180 Turn Off Output 3
181 Set Variable SALTO = SALTO + 1
182 Return from Subroutine

183 Set Subroutine POSIZIOA9
184 Open Gripper
185 Go to Position 24 Speed 50 (%)
186 Go to Position 23 Speed 10 (%)
187 Close Gripper
188 Go to Position 1 Speed 50 (%)
189 Go to Position 22 Speed 50 (%)
190 Go to Position 19 Speed 25 (%)
191 Go to Position 20 Speed 10 (%)
192 Jaw 38 (mm)
193 Go to Position 19 Speed 10 (%)
194 Go to Position 22 Speed 25 (%)
195 Go to Position 1 Speed 50 (%)
196 Turn On Output 3
197 Turn Off Output 3
198 Set Variable SALTO = SALTO + 1
199 Return from Subroutine

