Treball de Fi de Màster

Màster Universitari en Enginyeria Industrial

Disseny, programació i implementació d'un robot de dibuix amb Arduino

Annexos

Autor: Josep Maria Martí i Elias
Director: Manel Velasco Garcia

Convocatòria: Juny 2017

Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona

Índex

Apèndix A

Programes creats

A.1 Python

A.1.1 RobotMoveBT.py

```
import math, serial, time

arduino=serial.Serial('COM4', 9600)
time.sleep(0.1)

StepsVolta=1600
VelocitatMax=200
x0 = 0.0
y0 = 0.0
#posicio[2];
pasdreta=0; #posicio absoluta del motor dret en pasos
pasesquerra=0; #posicio absoluta del motor dret en pasos
DiamRoda=51.9; #diametre de la roda en mm
RadiRoda=DiamRoda/2.0; #radi de la roda en mm
d=122.0/2.0; #distancia en mm entre el centre de l'eix (punt del boli) i les
                                        rodes
distDreta=61.8
distEsquerra=61.9
angle=0.0; #angle entre punt de la circumferencia en radians
arc=0.0; #arc de circumferencia que s'ha de recorre en mm
pas= math.pi *DiamRoda/StepsVolta; #mm recorreguts per pas del motor
direccio0=math.pi/2; #angle inicial del robot a 90 graus
tempsLectura=0.01; #temps per llegir l'Arduino


def G00(x1,y1):
  global x0
  global y0
```

```
    global pasdreta
    global pasesquerra
    global distancia
    global pas
    global direccio0
    dx = x1 - x0;
    dy = y1 - y0;
    distancia = math.sqrt(dx*dx + dy*dy);
    steps=round(distancia/pas);
    if (dx != 0):
      direccio = math.atan(dy/dx);
      if (dx < 0 and dy >= 0):
        direccio = direccio + math.pi;
      elif (dx < 0 and dy < 0):
        direccio = direccio - math.pi
      apuntar(direccio);
      print (direccio)
    else:
      if (dy > 0):
        direccio = math.pi/2.0;
      elif (dy<0):
        direccio = -math.pi/2.0;
      else:
        direccio=direccio0
      apuntar(direccio);
    direccio0=direccio
    pasdreta=pasdreta+steps;
    pasesquerra=pasesquerra+steps;
    x0=x1;
    y0=y1;
    text='0'+','+str(pasdreta)+','+str(pasesquerra)+','
    arduino.write(text)
    robot=1
    while robot==1:
      if arduino.inWaiting()>0:
        st=arduino.readline().strip()
        time.sleep(tempsLectura)
        if st=='Ready':
          robot=0
    return

  def G01(x1,y1):
    global x0
    global y0
    global pasdreta
    global pasesquerra
    global distancia
    global pas
    dx = x1 - x0;
```

```
    dy = y1 - y0;
    distancia = math.sqrt(dx*dx + dy*dy);
    steps=round(distancia/pas);
    pasdreta=pasdreta+steps;
    pasesquerra=pasesquerra+steps;
    x0=x1;
    y0=y1;
    text='1'+','+str(pasdreta)+','+str(pasesquerra)+','
    arduino.write(text)
    robot=1
    while robot==1:
      if arduino.inWaiting()>0:
        st=arduino.readline().strip()
        time.sleep(tempsLectura)
        if st=='Ready':
          robot=0
    return

def G02(x1,y1,xC,yC,direccio1):
    global x0
    global y0
    global pasdreta
    global pasesquerra
    global distancia
    global pas
    global direccio0
    RadiGirC=math.sqrt(xC*xC+yC*yC)
    xC=xC+x0;
    yC=yC+y0;
    x0=x0-xC;
    y0=y0-yC;
    x1=x1-xC;
    y1=y1-yC;
    angle0=math.atan2(y0,x0)
    angle1=math.atan2(y1,x1)
    angle=angle0-angle1
    if angle<0:
      angle=2*math.pi+angle
    distanciaD = angle * (RadiGirC-distDreta);
    distanciaE = angle * (RadiGirC+distEsquerra);
    stepsD=round(distanciaD/pas);
    stepsE=round(distanciaE/pas);
    pasdreta=pasdreta+stepsD;
    pasesquerra=pasesquerra+stepsE;
    x0=x1+xC;
    y0=y1+yC;
    direccio0=direccio1;
    text='1'+','+str(pasdreta)+','+str(pasesquerra)+','
    arduino.write(text)
```

```
    robot=1
    while robot==1:
      if arduino.inWaiting()>0:
        st=arduino.readline().strip()
        time.sleep(tempsLectura)
        if st=='Ready':
          robot=0
    return

def G03(x1,y1,xC,yC,direccio1):
    global x0
    global y0
    global pasdreta
    global pasesquerra
    global distancia
    global pas
    global direccio0
    RadiGirC=math.sqrt(xC*xC+yC*yC)
    xC=xC+x0;
    yC=yC+y0;
    x0=x0-xC;
    y0=y0-yC;
    x1=x1-xC;
    y1=y1-yC;
    angle0=math.atan2(y0,x0)
    angle1=math.atan2(y1,x1)
    angle=angle1-angle0
    if angle<0:
      angle=2*math.pi+angle
    distanciaD = angle * (RadiGirC+distDreta);
    distanciaE = angle * (RadiGirC-distEsquerra);
    stepsD=round(distanciaD/pas);
    stepsE=round(distanciaE/pas);
    pasdreta=pasdreta+stepsD;
    pasesquerra=pasesquerra+stepsE;
    x0=x1+xC;
    y0=y1+yC;
    direccio0=direccio1;
    text='1'+','+str(pasdreta)+','+str(pasesquerra)+','
    arduino.write(text)
    robot=1
    while robot==1:
      if arduino.inWaiting()>0:
        st=arduino.readline().strip()
        time.sleep(tempsLectura)
        if st=='Ready':
          robot=0
    return
```

```
def apuntar(direccio1):
  global pasdreta
  global pasesquerra
  global distancia
  global pas
  global direccio0
  gir=direccio0-direccio1;
  absgir=abs(gir);
  if (absgir > math.pi):
    if (gir<0):
      gir= 2.0 * math.pi + gir;
    else:
      gir= -2.0 * math.pi + gir;
  distancia=gir*d;
  steps = distancia / pas
  pasdreta=pasdreta-steps;
  pasesquerra=pasesquerra+steps;
  direccio0=direccio1;
  text='0'+','+str(pasdreta)+','+str(pasesquerra)+','
  arduino.write(text)
  robot=1
  while robot==1:
    if arduino.inWaiting()>0:
      st=arduino.readline().strip()
      time.sleep(tempsLectura)
      if st=='Ready':
        robot=0
  return


def GCodeFile(oldfile, newfile):
  oldfile=open(oldfile, 'r')
  newfile=open(newfile, 'w')
  oldfile=oldfile.readlines()
  for linia in oldfile:
    linia=linia.strip()
    linia=linia.split()
    if linia!=[]:
      if linia[0]=='G00' and linia[1][0]=='X':
        x=float(linia[1][1:])
        y=float(linia[2][1:])
        G00(x,y)
        newfile.write('G00('+linia[1][1:]+', '+linia[2][1:]+');\n')

      elif linia[0]=='G01' and linia[1][0]=='X':
        x=float(linia[1][1:])
        y=float(linia[2][1:])
        G01(x,y)
        newfile.write('G01('+linia[1][1:]+', '+linia[2][1:]+');\n')
```

```
      elif linia[0]=='G01' and linia[1][0]=='A':
        angle=float(linia[1][1:])
        angle=angle-((2.0*math.pi)*(angle//(2.0*math.pi)))
        apuntar(angle)
        newfile.write('apuntar('+linia[1][1:]+');\n')

      elif linia[0]=='G02':
        x1=float(linia[1][1:])
        y1=float(linia[2][1:])
        xc=float(linia[4][1:])
        yc=float(linia[5][1:])
        angle=float(linia[-1][1:])
        angle=angle-((2.0*math.pi)*(angle//(2.0*math.pi)))
        G02(x1,y1,xc,yc,angle)
        newfile.write('G02('+linia[1][1:]+', '+linia[2][1:]+', '+linia[4][1:]+
                                    ', '+linia[5][1:]+', '+linia[-1][1:]+')
                                    ;\n')

      elif linia[0]=='G03':
        x1=float(linia[1][1:])
        y1=float(linia[2][1:])
        xc=float(linia[4][1:])
        yc=float(linia[5][1:])
        angle=float(linia[-1][1:])
        angle=angle-((2.0*math.pi)*(angle//(2.0*math.pi)))
        G03(x1,y1,xc,yc,angle)
        newfile.write('G03('+linia[1][1:]+', '+linia[2][1:]+', '+linia[4][1:]+
                                    ', '+linia[5][1:]+', '+linia[-1][1:]+')
                                    ;\n')

      else:
        pass
    else:
      pass
  newfile.close()


def GCode(fitxer):
  fitxer=open(fitxer, 'r')
  fitxer=fitxer.readlines()
  for linia in fitxer:
    linia=linia.strip()
    linia=linia.split()
    if linia!=[]:
      if linia[0]=='G00' and linia[1][0]=='X':
        x=float(linia[1][1:])
        y=float(linia[2][1:])
        G00(x,y)
```

```
        elif linia[0]=='G01' and linia[1][0]=='X':
          x=float(linia[1][1:])
          y=float(linia[2][1:])
          G01(x,y)


        elif linia[0]=='G01' and linia[1][0]=='A':
          angle=float(linia[1][1:])
          angle=angle-((2.0*math.pi)*(angle//(2.0*math.pi)))
          apuntar(angle)


        elif linia[0]=='G02':
          x1=float(linia[1][1:])
          y1=float(linia[2][1:])
          xc=float(linia[4][1:])
          yc=float(linia[5][1:])
          angle=float(linia[-1][1:])
          angle=angle-((2.0*math.pi)*(angle//(2.0*math.pi)))
          G02(x1,y1,xc,yc,angle)


        elif linia[0]=='G03':
          x1=float(linia[1][1:])
          y1=float(linia[2][1:])
          xc=float(linia[4][1:])
          yc=float(linia[5][1:])
          angle=float(linia[-1][1:])
          angle=angle-((2.0*math.pi)*(angle//(2.0*math.pi)))
          G03(x1,y1,xc,yc,angle)


        else:
          pass
      else:
        pass
```

A.1.2 Draw.py

```
import turtle
import math
import RobotMoveBT

moviments=[]

def cercle(t,R,angle):
  t.circle(R,angle)

def recta(t,x,y):
  t.pendown()
  t.goto(x,y)

def rectaup(t,x,y):
  t.penup()
  t.goto(x,y)

def dibuixApuntar(t,angle):
  t.setheading(angle)
  ordre=['apuntar',angle]
  moviments.append(ordre)

def dibuixG00(t,x1,y1):
  t.penup()
  (x0,y0)=t.pos()
  dx = x1 - x0;
  dy = y1 - y0
  if (dx != 0):
    direccio = math.atan(dy/dx);
    if (dx < 0 and dy >= 0):
      direccio = direccio + math.pi;
    elif (dx < 0 and dy < 0):
      direccio = direccio - math.pi
    t.setheading(direccio);
  else:
    if (dy > 0):
      direccio = math.pi/2.0;
    elif (dy<0):
      direccio = -math.pi/2.0;
    else:
      direccio=t.heading()
    t.setheading(direccio);
  t.goto(x1,y1)
  ordre=['G00',x1,y1]
  moviments.append(ordre)
  return
```

```
def dibuixG01(t,x1,y1):
  t.pendown()
  (x0,y0)=t.pos()
  dx = x1 - x0;
  dy = y1 - y0
  if (dx != 0):
    direccio = math.atan(dy/dx);
    if (dx < 0 and dy >= 0):
      direccio = direccio + math.pi;
    elif (dx < 0 and dy < 0):
      direccio = direccio - math.pi
    t.setheading(direccio);
  else:
    if (dy > 0):
      direccio = math.pi/2.0;
    elif (dy<0):
      direccio = -math.pi/2.0;
    else:
      direccio=t.heading()
    t.setheading(direccio);
  t.goto(x1,y1)
  ordre=['apuntar',direccio]
  moviments.append(ordre)
  ordre=['G01',x1,y1]
  moviments.append(ordre)
  return


def dibuixG02(t,x1,y1,xC,yC,angle):
  t.pendown()
  radi=math.sqrt(xC*xC + yC*yC)
  (x0,y0)=t.pos()
  xC=xC+x0;
  yC=yC+y0;
  x0=x0-xC;
  y0=y0-yC;
  x1=x1-xC;
  y1=y1-yC;
  angle0=math.atan2(y0,x0)
  angle1=math.atan2(y1,x1)
  angle=angle0-angle1
  if angle<0:
    angle=2*math.pi+angle
  t.circle(-radi,angle)
  ordre=['G02',x1,y1,xC,yC,t.heading()]
  moviments.append(ordre)
  return

def dibuixG03(t,x1,y1,xC,yC,angle):
```

```
    t.pendown()
    radi=math.sqrt(xC*xC + yC*yC)
    (x0,y0)=t.pos()
    xC=xC+x0;
    yC=yC+y0;
    x0=x0-xC;
    y0=y0-yC;
    x1=x1-xC;
    y1=y1-yC;
    angle0=math.atan2(y0,x0)
    angle1=math.atan2(y1,x1)
    angle=angle1-angle0
    if angle<0:
      angle=2*math.pi+angle
    t.circle(radi,angle)
    ordre=['G03',x1,y1,xC,yC,t.heading()]
    moviments.append(ordre)
    return

def afegir(t,val):
    moviments.append(val)
    return

def desfer(t):
    t.undo()
    if moviments[-1][0]=='G01':
      del moviments[-1]
      del moviments[-1]
    else:
      del moviments[-1]
    return


def previsualitzaFitxer(t,fitxer):
    fitxer=open(fitxer, 'r')
    fitxer=fitxer.readlines()
    for linia in fitxer:
      linia=linia.strip()
      linia=linia.split()
      if linia!=[]:
        if linia[0]=='G00' and linia[1][0]=='X':
          x=float(linia[1][1:])
          y=float(linia[2][1:])
          dibuixG00(t,x,y)


        elif linia[0]=='G01' and linia[1][0]=='X':
          x=float(linia[1][1:])
          y=float(linia[2][1:])
```

```
          dibuixG01(t,x,y)


        elif linia[0]=='G01' and linia[1][0]=='A':
          angle=float(linia[1][1:])
          angle=angle-((2.0*math.pi)*(angle//(2.0*math.pi)))
          dibuixApuntar(t,angle)


        elif linia[0]=='G02':
          x1=float(linia[1][1:])
          y1=float(linia[2][1:])
          xc=float(linia[4][1:])
          yc=float(linia[5][1:])
          angle=float(linia[-1][1:])
          angle=angle-((2.0*math.pi)*(angle//(2.0*math.pi)))
          dibuixG02(t,x1,y1,xc,yc,angle)


        elif linia[0]=='G03':
          x1=float(linia[1][1:])
          y1=float(linia[2][1:])
          xc=float(linia[4][1:])
          yc=float(linia[5][1:])
          angle=float(linia[-1][1:])
          angle=angle-((2.0*math.pi)*(angle//(2.0*math.pi)))
          dibuixG03(t,x1,y1,xc,yc,angle)


        else:
          pass
      else:
        pass
    return

def reset():
    t.reset()
    return

def representaLlist():
    for i in moviments:
      if i[0]=='G00':
        dibuixG00(i[1],i[2])
      elif i[0]=='G01':
        dibuixG01(i[1],i[2])
      elif i[0]=='G02':
        dibuixG02(i[1],i[2],i[3],i[4],i[5])
      elif i[0]=='G03':
        dibuixG03(i[1],i[2],i[3],i[4],i[5])
```

```
    elif i[0]=='apuntar':
      dibuixApuntar(i[1])
    else:
      pass
  return

def dibuixaLlista():
  import RobotMoveBT
  for i in moviments:
    if i[0]=='G00':
      RobotMoveBT.G00(i[1],i[2])
    elif i[0]=='G01':
      RobotMoveBT.G01(i[1],i[2])
    elif i[0]=='G02':
      RobotMoveBT.G02(i[1],i[2],i[3],i[4],i[5])
    elif i[0]=='G03':
      RobotMoveBT.G03(i[1],i[2],i[3],i[4],i[5])
    elif i[0]=='apuntar':
      RobotMoveBT.apuntar(i[1])
    else:
      pass
  return
```

A.1.3 AppTFM.py

```
import math

import Tkinter as tk
import ttk

import turtle
import Draw
import RobotMoveBT

LARGE_FONT= ("Verdana", 12)


class TFM(tk.Tk):

  def __init__(self, *args, **kwargs):

    tk.Tk.__init__(self, *args, **kwargs)

    tk.Tk.wm_title(self, "TFM Josep Marti")


    container = tk.Frame(self)
    container.pack(side="top", fill="both", expand = True)
    container.grid_rowconfigure(0, weight=1)
    container.grid_columnconfigure(0, weight=1)

    self.frames = {}

    for F in (StartPage, Inkscape, Manual):

      frame = F(container, self)

      self.frames[F] = frame

      frame.grid(row=0, column=0, sticky="nsew")

    self.show_frame(StartPage)

  def show_frame(self, cont):

    frame = self.frames[cont]
    frame.tkraise()


class StartPage(tk.Frame):

  def __init__(self, parent, controller):
```

```
    tk.Frame.__init__(self,parent)
    label = tk.Label(self, text="Com vols dibuixar?", font=LARGE_FONT)
    label.pack(pady=10,padx=10)

    button = ttk.Button(self, text="Importar archiu GCode creat per Inkscape",
    command=lambda: controller.show_frame(Inkscape))
    button.pack(pady=10,padx=10)

    button2 = ttk.Button(self, text="Realitzar un dibuix pas a pas",
    command=lambda: controller.show_frame(Manual))
    button2.pack(pady=5,padx=5)



class Inkscape(tk.Frame):

  def __init__(self, parent, controller):
  nomfitxer=tk.StringVar(None)

  tk.Frame.__init__(self, parent)


  label = tk.Label(self, text="Importar archiu des de Inkscape", font=
                                    LARGE_FONT).pack(pady=10,padx=10)

  label2=tk.Label(self, text='Nom del fitxer:').pack()

  fitxer=tk.Entry(self, textvariable=nomfitxer).pack(pady=10,padx=10)

  button1 = ttk.Button(self, text="Dibuixar",command=lambda:Dibuixa(self)).
                                    pack(pady=10,padx=10)

  button2 = ttk.Button(self, text="Previsualitzar",command=lambda:Preview(self
                                    )).pack(pady=10,padx=10)

  button3 = ttk.Button(self, text="Tornar a l\'inici",command=lambda: back(
                                    self)).pack(pady=10,padx=10)

  canvas = tk.Canvas(self, width=500, height=500)
  canvas.pack(pady=10,padx=10)

  turtle1 = turtle.RawTurtle(canvas)
  turtle1.shape("turtle")
  turtle1.setheading(90)
  turtle1.radians()

  def Dibuixa(self):
    nom=nomfitxer.get()
```

```
      RobotMoveBT.GCode(nom)

    def Preview(self):
      nom=nomfitxer.get()
      Draw.previsualitzaFitxer(turtle1,nom)

    def back(self):
      turtle1.reset()
      turtle1.setheading(math.pi/2.0)
      controller.show_frame(StartPage)



class Manual(tk.Frame):

    def __init__(self, parent, controller):

      tk.Frame.__init__(self, parent)

      label0=tk.Label(self, text='           ').grid(row=0,column=0)

      label = tk.Label(self, text="Pas a pas", font=LARGE_FONT).grid(row=1,
                                          column=2)

      label1=tk.Label(self, text='           ').grid(row=3,column=0)


      ordre=tk.StringVar()
      ordre.set(None)

      G00X=tk.DoubleVar()
      G00Y=tk.DoubleVar()
      G01X=tk.DoubleVar()
      G01Y=tk.DoubleVar()
      G02X=tk.DoubleVar()
      G02Y=tk.DoubleVar()
      G02I=tk.DoubleVar()
      G02J=tk.DoubleVar()
      G02A=tk.DoubleVar()
      G03X=tk.DoubleVar()
      G03Y=tk.DoubleVar()
      G03I=tk.DoubleVar()
      G03J=tk.DoubleVar()
      G03A=tk.DoubleVar()
      RA=tk.DoubleVar()

      checkBox2Graus=tk.BooleanVar()
      checkBox3Graus=tk.BooleanVar()
      checkBoxGraus=tk.BooleanVar()
```

```
#G00
radioG=tk.Radiobutton(self, text='Linia recta de posicionament (G00): ',
                                    value='G00', variable= ordre).grid(row=3
                                    ,column=0)
labelG00X=tk.Label(self, text='X =').grid(row=3,column=2)
entryG00X=tk.Entry(self, textvariable= G00X, width=8).grid(row=3,column=3)
labelG00Y=tk.Label(self, text='Y =').grid(row=3,column=5)
entryG00Y=tk.Entry(self, textvariable=G00Y, width=8).grid(row=3,column=6)



#G01
radioG=tk.Radiobutton(self,text='Linia recta (G01): ', value='G01',
                                    variable=ordre).grid(row=5,column=0)
labelG01X=tk.Label(self, text='X =').grid(row=5,column=2)
entryG01X=tk.Entry(self, textvariable=G01X, width=8).grid(row=5,column=3)
labelG01Y=tk.Label(self, text='Y =').grid(row=5,column=5)
entryG01Y=tk.Entry(self, textvariable=G01Y, width=8).grid(row=5,column=6)



#G02
radioG=tk.Radiobutton(self,text='Arc en sentit horari (G02): ', value='G02
                                    ', variable=ordre).grid(row=7,column=0)
labelG02X=tk.Label(self, text='X =').grid(row=7,column=2)
entryG02X=tk.Entry(self, textvariable=G02X, width=8).grid(row=7,column=3)
labelG02Y=tk.Label(self, text='Y =').grid(row=7,column=5)
entryG02Y=tk.Entry(self, textvariable=G02Y, width=8).grid(row=7,column=6)
labelG02I=tk.Label(self, text='I =').grid(row=7,column=8)
entryG02I=tk.Entry(self, textvariable=G02I, width=8).grid(row=7,column=9)
labelG02J=tk.Label(self, text='J =').grid(row=7,column=11)
entryG02J=tk.Entry(self, textvariable=G02J, width=8).grid(row=7,column=12)
labelG02A=tk.Label(self, text='Orientacio =').grid(row=7,column=14)
entryG02A=tk.Entry(self, textvariable=G02A, width=8).grid(row=7,column=15)



#G03
radioG=tk.Radiobutton(self,text='Arc en sentit horari (G03): ', value='G03
                                    ', variable=ordre).grid(row=9,column=0)
labelG03X=tk.Label(self, text='X =').grid(row=9,column=2)
```

```
        entryG03X=tk.Entry(self, textvariable=G03X, width=8).grid(row=9,column=3)
        labelG03Y=tk.Label(self, text='Y =').grid(row=9,column=5)
        entryG03Y=tk.Entry(self, textvariable=G03Y, width=8).grid(row=9,column=6)
        labelG03I=tk.Label(self, text='I =').grid(row=9,column=8)
        entryG03I=tk.Entry(self, textvariable=G03I, width=8).grid(row=9,column=9)
        labelG03J=tk.Label(self, text='J =').grid(row=9,column=11)
        entryG03J=tk.Entry(self, textvariable=G03J, width=8).grid(row=9,column=12)
        labelG03A=tk.Label(self, text='Orientacio =').grid(row=9,column=14)
        entryG03A=tk.Entry(self, textvariable=G03A, width=8).grid(row=9,column=15)




        #Rotacio
        radioG=tk.Radiobutton(self,text='Rotacio: ', value='apuntar', variable=
                                        ordre).grid(row=11,column=0)
        labelRA=tk.Label(self, text='Orientacio =').grid(row=11,column=2)
        entryRA=tk.Entry(self, textvariable=RA, width=8).grid(row=11,column=3)

        checkBox1=tk.Checkbutton(self, variable=checkBoxGraus, text="Orientacio en
                                         graus").grid(row=11, column=5)



        #Botons
        buttonDibuixar=tk.Button(self, text='Dibuixar', fg='blue', command=lambda:
                                        Dibuixar(self)).grid(row=20,column=7)
        buttonVeure=tk.Button(self, text='Veure', fg='blue', command=lambda: Veure
                                        (self)).grid(row=20,column=3)
        buttonUndo=tk.Button(self, text='Desfer', fg='blue', command=lambda: undo(
                                        self)).grid(row=20,column=5)
        buttonTorna=tk.Button(self, text='Tornar a l\'inici', fg='blue', command=
                                        lambda:back(self)).grid(row=20,column=9)

        canvas = tk.Canvas(self, width=500, height=500)
        canvas.grid(row=25,column=1,columnspan=12)

        turtle1 = turtle.RawTurtle(canvas)
        turtle1.shape("turtle")
        turtle1.setheading(90)
        turtle1.radians()

        label1=tk.Label(self, text='          ').grid(row=11,column=5)




        def Dibuixar(self):
```

```
    Draw.dibuixaLlista()


  def Veure(self):
    print ordre.get()
    tria=ordre.get()
    if tria == 'G00':
      x=G00X.get()
      y=G00Y.get()
      Draw.dibuixG00(turtle1,x,y)

    elif tria == 'G01':
      x=G01X.get()
      y=G01Y.get()
      Draw.dibuixG01(turtle1,x,y)

    elif tria == 'G02':
      x=G02X.get()
      y=G02Y.get()
      i=G02I.get()
      j=G02J.get()
      angle=G02A.get()
      graus=checkBox1.get()
      if graus == True:
        angle= (angle*math.pi)/180
      print (x, y, i , j, angle)
      Draw.dibuixG02(turtle1,x,y,i,j,angle)

    elif tria == 'G03':
      x=G03X.get()
      y=G03Y.get()
      i=G03I.get()
      j=G03J.get()
      angle=G03A.get()
      graus=checkBox1.get()
      if graus == True:
        angle= (angle*math.pi)/180
      Draw.dibuixG03(turtle1,x,y,i,j,angle)

    elif tria == 'apuntar':
      angle=RA.get()
      graus=checkBox1.get()
      if graus == True:
        angle= (angle*math.pi)/180
      else:
        pass
      Draw.dibuixapuntar(turtle1,angle)

    else:
```

```
            pass

        def undo(self):
            Draw.desfer(turtle1)


        def back(self):
            turtle1.reset()
            turtle1.setheading(math.pi/2.0)
            controller.show_frame(StartPage)



app = TFM()
app.mainloop()
```

A.2 Arduino

A.2.1 Robot.ino

```
#include <AccelStepper.h>
#include <MultiStepper.h>
#include <Servo.h>
#include <SoftwareSerial.h>

SoftwareSerial bluetooth(10,11);

Servo boli;
int up=0;
int down=50;

AccelStepper RodaDreta(1,9,8);
AccelStepper RodaEsquerra(1,13,12);
MultiStepper Robot;
int VelocitatMax=500;
long posicio[2];
int pasdreta=0; //posicio absoluta del motor dret en pasos
int pasesquerra=0; //posicio absoluta del motor dret en pasos

int text[3];
int cnt=0;
boolean Rebut = false;
```

```
void setup() {
  bluetooth.begin(9600);


  pinMode(2, OUTPUT); //microstepping off
  pinMode(3,OUTPUT);
  digitalWrite(2,HIGH);
  digitalWrite(3,HIGH);
  pinMode(4, OUTPUT);
  pinMode(5,OUTPUT);
  digitalWrite(4,HIGH);
  digitalWrite(5,HIGH);
  boli.attach(6);
  boli.write(0);

  Robot.addStepper(RodaDreta);
  Robot.addStepper(RodaEsquerra);

  RodaDreta.setMaxSpeed(VelocitatMax);
  RodaEsquerra.setMaxSpeed(VelocitatMax);
}

void loop() {
  getSerialData();
  delay(1);
  processData();
}

void getSerialData(){
  if(bluetooth.available() > 0) {
    String x = bluetooth.readString();
    String buff="";

    for (int i=0; i<x.length();i++){
      String caracter="";
      caracter=caracter+x[i];
      if (caracter!=","){
        buff=buff+x[i];
      }
      else {
        int y=buff.toInt();
        text[cnt]=y;
        buff="";
        if (cnt<3){
          cnt+=1;
          }
```

```
        if (cnt==3){
          cnt=0;
        }
      }
    }
    Rebut=true;
    delay(1);
  }
}


void processData(){
  if (Rebut==true){
    if (text[0]==0 and boli.read()!=up){
      boli.write(up);
      delay(300);
    }
    else if (text[0]==1 and boli.read()!=down){
      boli.write(down);
      delay(300);
    }
    posicio[0]=-text[1];
    posicio[1]=text[2];
    Robot.moveTo(posicio);
    Robot.runSpeedToPosition();
    bluetooth.println("Ready");
    Rebut=false;
  }
}
```

Apèndix B

Plànols de les peces

B.1 Xassís

B.2 Guia del retolador

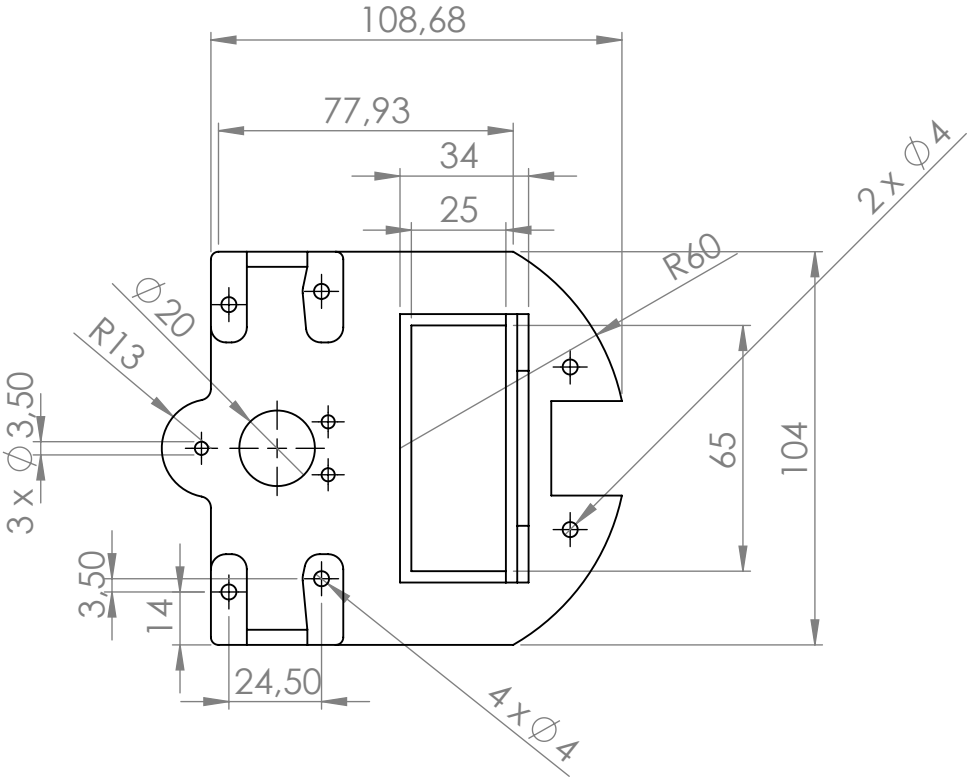
B.3 Mecanisme d'elevació del retolador

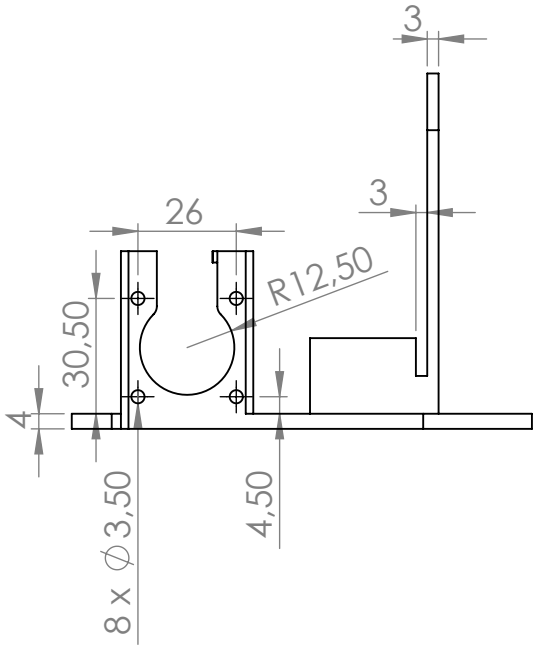
B.4 Roda boja davantera

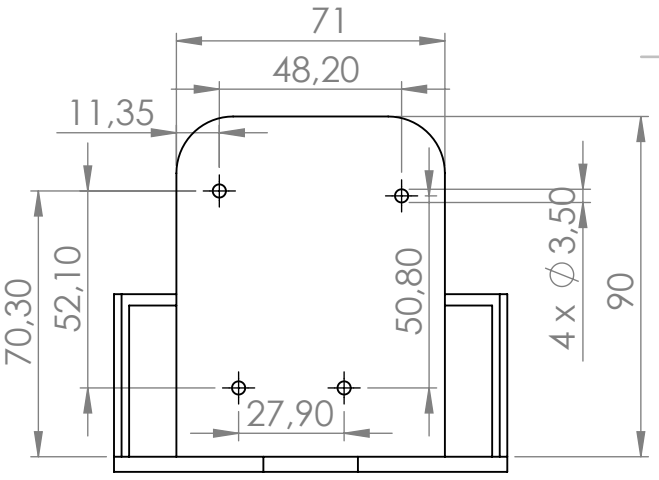
B.5 Roda motriu

B.6 Tubs auxiliars

B.1 Xassís

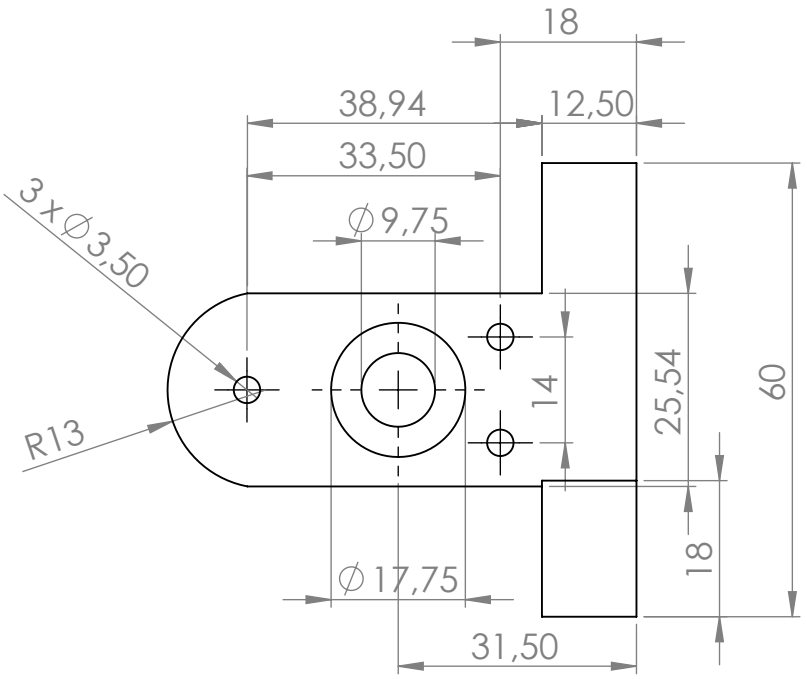


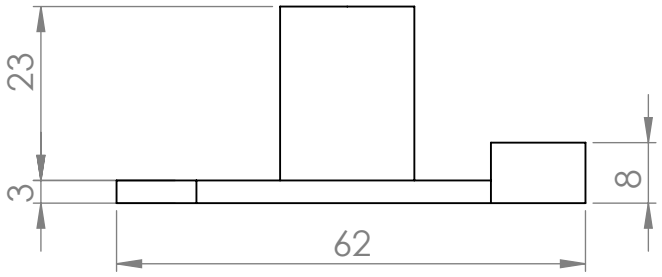




Nom:	Josep Martí
Títol:	Xassís
Número de dibuix:	1/6
	A4
ESCALA:1:2	

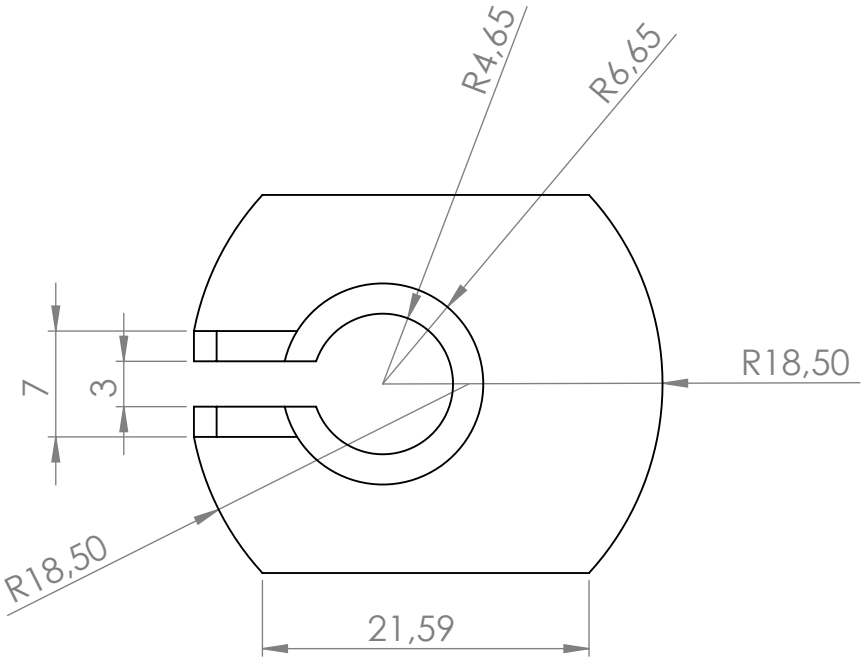
B.2 Guia del retolador

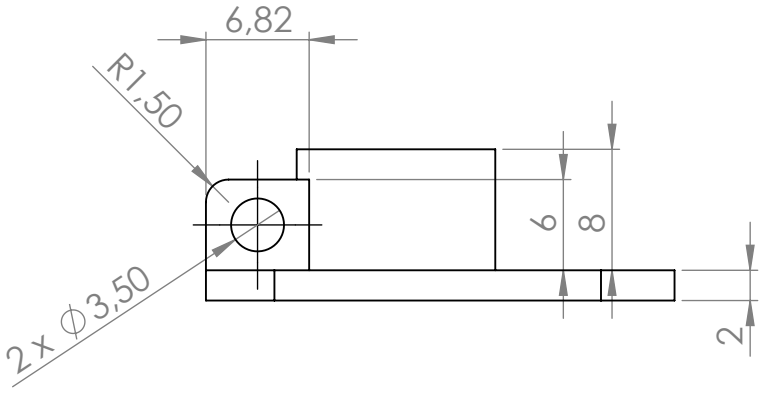




Nom:	Josep Martí
Títol:	Guia
Número de dibuix:	2/6
	A4
ESCALA1:1	

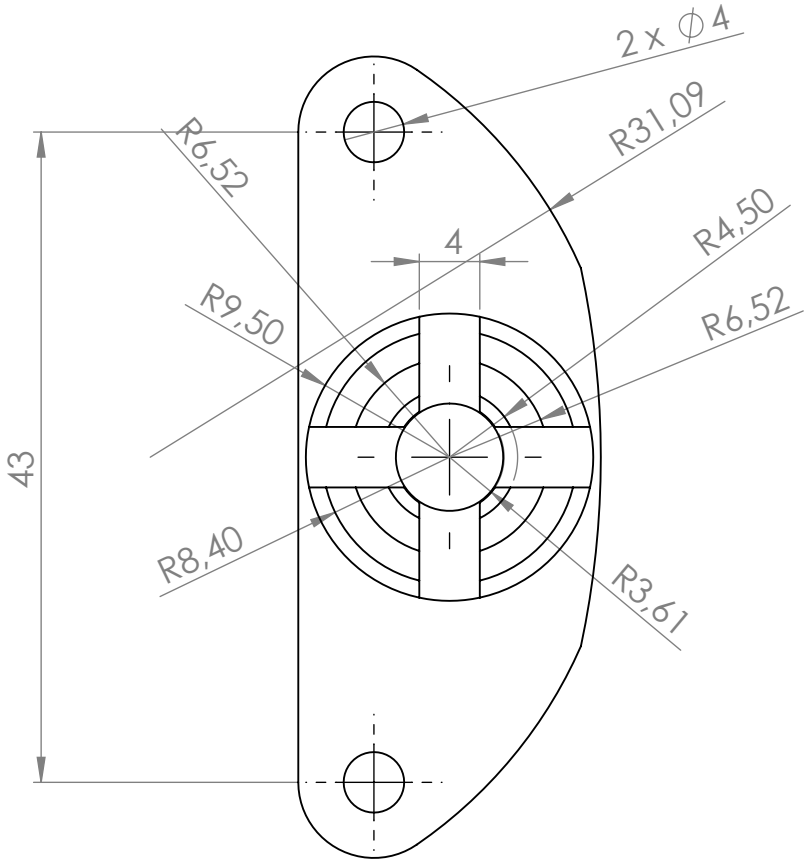
B.3 Mecanisme d'elevació del retolador

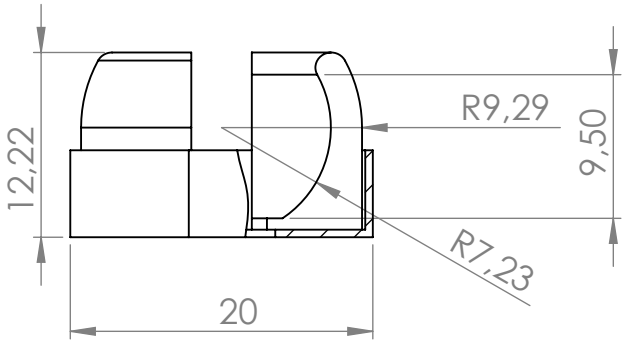




Nom:	Josep Martí
Títol:	Mecanisme d'elevació
Número de dibuix:	3/6
	A4
ESCALA 2:1	

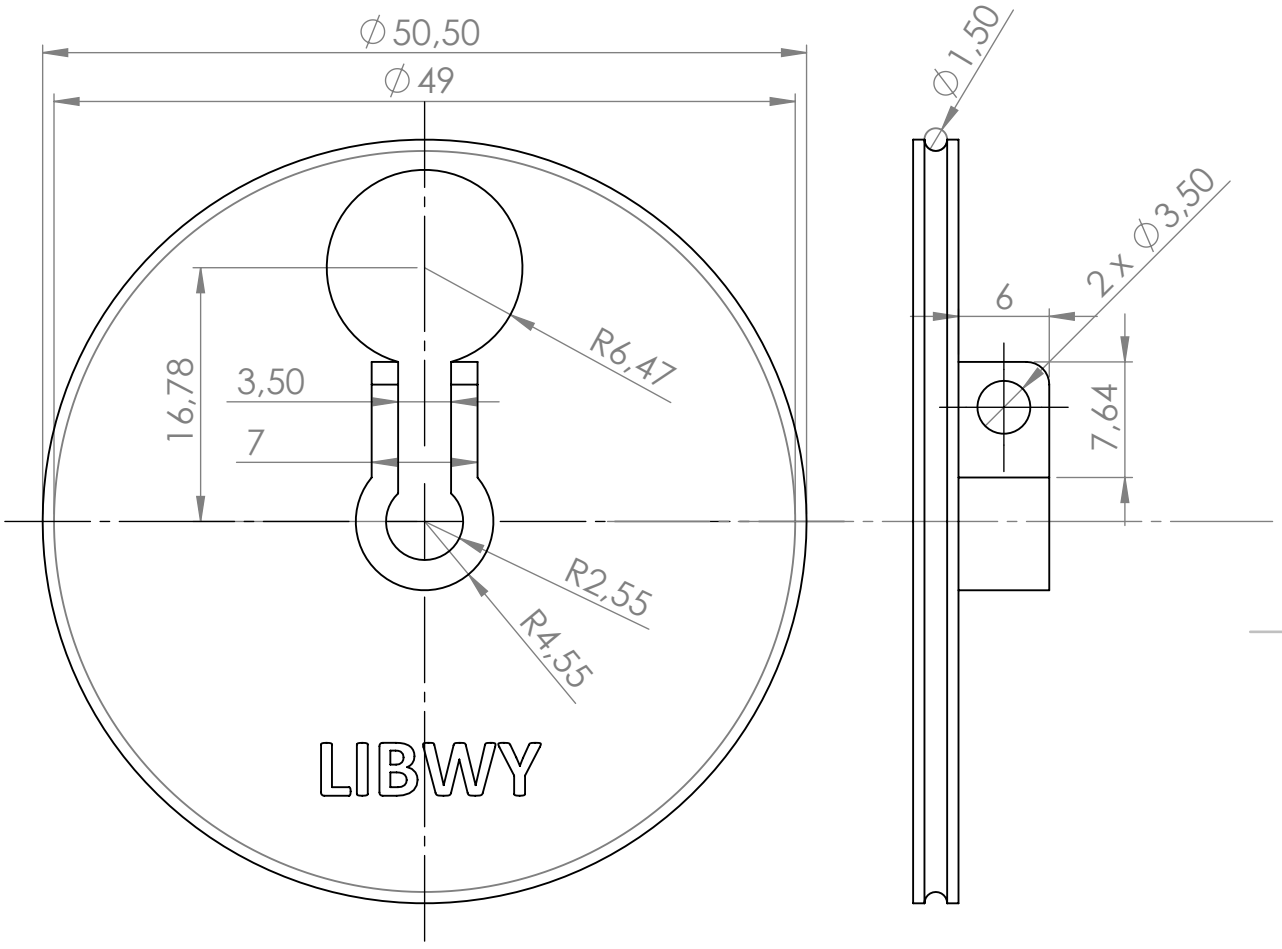
B.4 Roda boja davantera





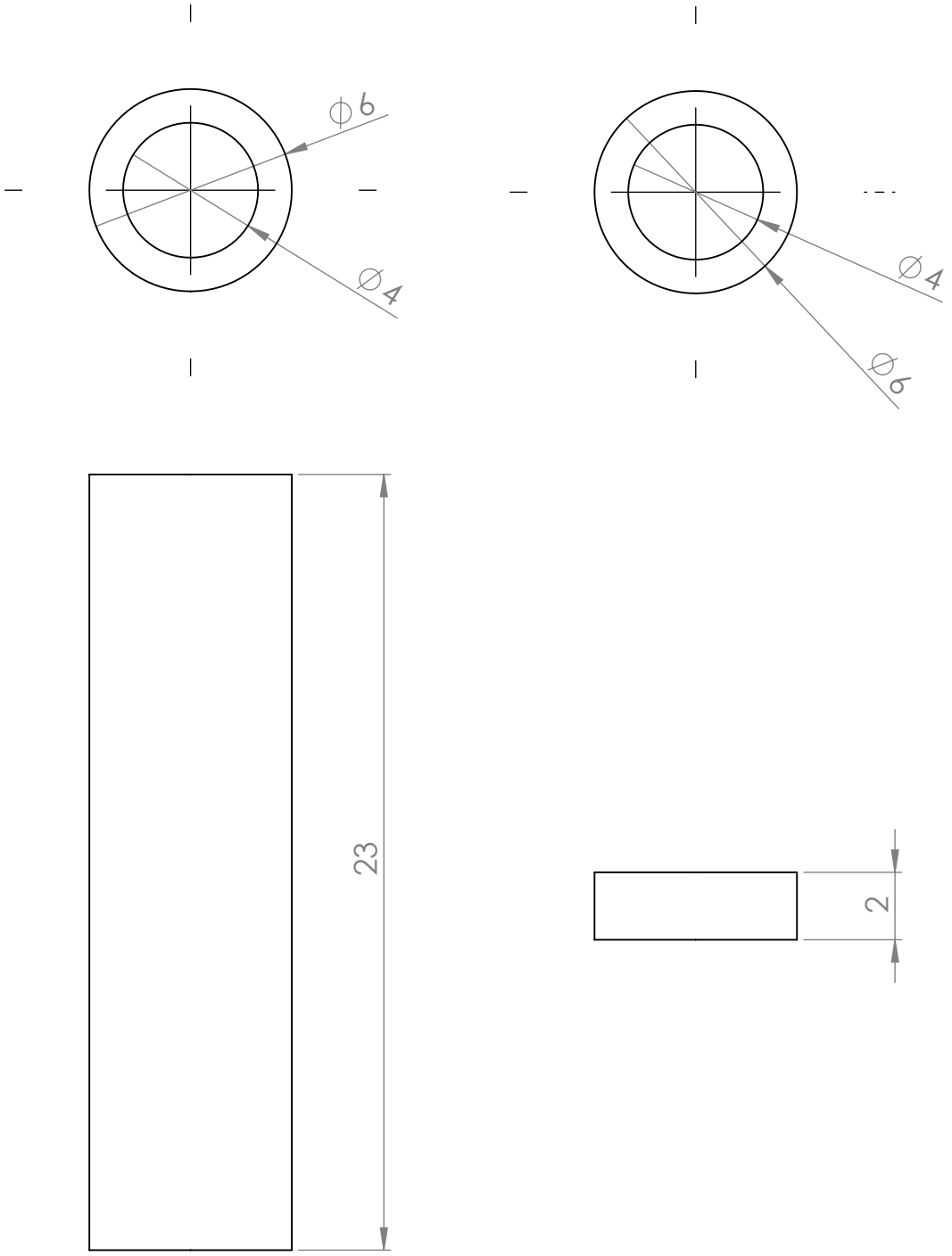
Nom:	Josep Martí
Títol:	Roda boja
Número de dibuix:	4/6
	A4
ESCALA 2:1	

B.5 Roda motriu



Nom:	Josep Martí
Títol:	Roda
Número de dibuix:	5/6
	A4
ESCALA 2:1	

B.6 Tubs auxiliars



Nom:	Josep Martí
Títol:	Tubs auxiliars
Número de dibuix:	6/6
	A4
ESCALA 5:1	