```python
#importem la funció sleep de la llibreria time
from time import sleep
#Importem la llibreria MyCobot que ens permet comunicar amb el robot
from pymycobot.mycobot import MyCobot
#Importem la llibreria d'entrades i sortides de la raspberry
import RPi.GPIO as GPIO
#Importem la llibreria de comunicació serie per a poder-nos comunicar amb l'ordinador
import serial

#iniciem l'objecte per el qual ens comunicarrem amb l'ordinador
ser = serial.Serial('/dev/ttyUSB0', 9600)
#iniciem l'objecte per al qual ens comunicarem amb el robot
mc = MyCobot('/dev/ttyAMA0',1000000)
#Establim que les entrades i sortides estaran codificades amb el nombre del processador en lloc
#del nombre pintat a la placa de la raspberry ja que no coincideixen
GPIO.setmode(GPIO.BCM)

#Definim una constant per referir-nos al pin que farem servir per la entrada del botó start
PINSTART = 23
#Definim una constant per referir-nos al pin que farem servir per la entrada del botó stop
PINSTOP = 1
#Definim una constant per referir-nos al pin que farem serfir per a enviar la senyal de sortida al relé
PINSORTIDA = 19

#Iniciem les entrades i sortides com a tals
GPIO.setup(PINSTART, GPIO.IN)
GPIO.setup(PINSTOP,GPIO.IN)
GPIO.setup(PINSORTIDA, GPIO.OUT)

#Establim una variable que farem servir en cada moment per saber si el robot s'està movent o no
movent = False
#Definim una variable que ens farà de buffer per a guardar l'ultim valor llegit en la entrada sèrie
info = b'b'
#Creem dues variables per guardar les dades que arribin des de l'ordinador
fila = 0
columna = 0
#creem una llista amb totes les posicions en que es pot deixar una peça
posicions = [[39.46, -58.53, -80.06, -54.84, -43.85, 53.08],[47.28, -64.68, -67.41, -50.53, -30.41, 41.83],[58.71, -53.87, -80.06, -43.15, -37.0, 35.24],[27.86, -67.41, -55.19, -52.82, -51.59, 35.24],[38.05, -67.14, -55.01, -51.85, -45.26, 34.98],[51.32, -67.5, -55.28, -53.61, -51.06, 35.06],[27.33, -67.5, -55.01, -45.87, -73.3, 34.89],[34.18, -67.5, -55.01, -44.03, -72.94, 34.89],[45.08, -67.58, -55.19, -45.7, -76.28, 34.89]]
#Creem una llista amb totes les posicions en les que es pot recollir una peça de forma de creu
recollircreu = [[-17.49, -39.02, -142.11, 10.45, -98.08, 22.58],[-11.42, -36.47, -127.35, -12.12, -96.32, 43.15],[-11.86, -51.41, -94.48, -33.75, -94.74, 42.97],[-9.49, -61.69, -66.44, -50.18, -94.48, 42.97]]
#Creem una llista més amb totes les posicions que es pot recollir les peçes amb forma de cercle
recollircercle = [[123.39, -56.68, -86.39, -42.97, -52.91, 49.57], [110.39, -54.93, -86.48, -42.01, -62.92, 43.85], [95.44, -50.53, -86.83, -36.21, -78.92, 35.06], [81.38, -60.38, -79.62, -19.33, -88.41, 15.73]]
#Definim una posició inicial amb el robot completament estirat
INICI = [0,0,0,0,0,0]
#Fem una posició intermitja des de la qual es pugui arribar a totes les posicions de la graella
INTERSORITR = [38.49, -26.8, -55.28, -94.13, -83.05, 33.13]
#Definim una constant per a la velocitat
VELOCITAT = 15
```

```
49 #Definim una posició des de la que es pugui arribar a totes les posicions utilitzades per a
   agafar les peces en forma de creu
50 CREUINTER = [5.71, 50.62, -126.47, 14.94, -81.12, -72.42]
51 #Definim una posició des de la que es pugui arribar a totes les posicions utilitzades per a
   agafar les peces en forma de cercle
52 CERCLEINTER = [99.22, -44.73, -33.66, -106.78, -68.64, 38.4]
53 #Tenim una variable estat en el que es guardarà el procés actual del robot
54 estat = 0
55 #Finalment s'engega el robot en cas que estigués parat
56 mc.power_on()
57 #s'engega un bucle amb la màquina d'estats corresponent
58 while True:
59     #En cas que el robot s'estigui movent s'activa la sortida, si ho ho fa es desactiva
60     if movent:
61         GPIO.output(PINSORTIDA, 1)
62     else:
63         GPIO.output(PINSORTIDA, 0)
64     #Es mostra l'estat per pantalla, ja que permet un major control del que passa en tot
   moment
65     print(estat)
66     #En cas que es premi el pulsador d'estop es torna a iniciar el robot
67     if not GPIO.input(PINSTOP)and estat != 0:
68         #Es para allà on sigui
69         mc.pause()
70         #Es passa a l'estat inicial
71         estat = 0
72         #Es deiaxa de moure
73         movent = False
74         #En cas que es pugui escriure a través del canal serie
75         if ser.writable:
76             #S'escriu el byte a, d'atur
77             ser.write(b'a')
78     #Estat inicial
79     if estat ==0:
80         #Posem el display de color vermell
81         mc.set_color(255,0,0)
82         #En el cas que es premi el pin de stert
83         if GPIO.input(PINSTART):
84             #Saltem a l'estat 1
85             estat = 1
86             #Enviem la posició inicial al robot
87             mc.send_angles(INICI, VELOCITAT)
88             #Indiquem que el robot s'està movent
89             movent = True
90             #Esperem un petit moment per evitar errors de lectura en la posició del robot.
91             #Procés que es repeteix més endavant amb la mateixa finalitat
92             sleep(0.1)
93     #Estat 1
94     elif estat == 1:
95         #Obrim la pinça del robot
96         mc.set_pwm_output(23,50,15)
97         #Esperem un moment
98         sleep(0.1)
99         #Comporovem si el robot ha arribat a la posició desitjada
100        if mc.is_in_position(INICI):
101            #En cas que hi hagi arribat indiquem que ja no es mou
102            movent = False
103            #Passem al següent estat
104            estat = 2
105            #Enviem el byte m a través del canal serie per indicar a l'ordinador que pot
   començar la transmissió
106            ser.write(b'm')
```

```
107     elif estat == 2:
108         #Canviem el color del display a verd
109         mc.set_color(0,255,0)
110         #En el cas que hi hagi alguna cosa per llegir en el canal serie
111         if ser.inWaiting():
112             #Es llegeix el primer byte que arribés i es guarda a la variable info
113             info = ser.read()
114             #comprova la informació per si coincideix amb s
115             if info == 's':
116                 #en tal cas torna a enviar la senyal de marxa, de tal manera que s'engegui
el programa que s'engegui primer
117                 #el programa funciona de manera adequada
118                 ser.write(b'm')
119             #En el cas que la informació sigui f, el seguent nombre serà la fila de destí i
es llegeix en l'estat 3
120             if info == 'f':
121                 estat = 3
122             #En el cas que la informació siguic, el seguent nombre serà la columna de destí
i es llegeix en l'estat 4
123             elif info == 'c':
124                 estat = 4
125     elif estat == 3:
126         #Espera a que hi hagi alguna cosa per llegir si no hi ha res
127         if ser.inWaiting():
128             #Es llegeix un byte
129             info = ser.read()
130             #Es converteix a enter i es posa a la variable fila
131             fila = int(info)
132             #Es torna a l'estat anterior
133             estat = 2
134     elif estat == 4:
135         #Espera a que hi hagi alguna cosa per llegir
136         if ser.inWaiting():
137             #Llegeix un byte
138             info = ser.read()
139             #Transforma el byte a un nombre enter i el guarda a la variable columna
140             columna = int(info)
141             #Es passa al següent estat
142             estat = 5
143     elif estat == 5:
144         #Espera a que hi hagi alguna cosa per llegir si no hi ha res
145         if ser.inWaiting():
146             #Es llegeix un byte
147             info = ser.read()
148             #En el cas que el byte fos 9 es tornaria a l'estat inicial ja que significaria
que hi ha hagut algun error
149             if int(info) == 9:
150                 estat = 0
151             #En el cas que el nombre sigui parell es va a l'estat 6
152             elif int(info)%2 == 0:
153                 estat = 6
154             #En cas que sigui senar a l'estat 7
155             else: estat = 7
156     elif estat == 6:
157         #S'envien les posicions d'aproximar-se a les peces en forma de creu al robot
158         mc.send_angles(CREUINTER,VELOCITAT)
159         #s'sindica que el robot s'està movent
160         movent = True
161         sleep(0.1)
162         #En cas que el robot hagi arribat a la posició desitjada
163         if mc.is_in_position(CREUINTER):
164             #S'indica que el robot ja no es mou
```

```
            movent = False
            #Es passa al seguent estat
            estat = 8
    elif estat == 7:
        #S'envien les posicions d'aproximar-se a les peces en forme de cercle al robot
        mc.send_angles(CERCLEINTER,VELOCITAT)
        #s'sindica que el robot s'està movent
        movent = True
        sleep(0.1)
        #En cas que el robot hagi arribat a la posició desitjada
        if mc.is_in_position(CERCLEINTER):
            #S'indica que el robot ja no es mou
            movent = False
            #Es passa al següent estat
            estat = 9
    elif estat == 8:
        #Es calcula quina peça s'ha d'agafar del carregador
        pos = int(info) / 2
        #s'envien els angles d'aquella peça
        mc.send_angles(recollircreu[pos],VELOCITAT)
        movent = True
        sleep(0.1)
        #Si el robot estigui en posició
        if mc.is_in_position(recollircreu[pos]):
            movent = False
            #Es passa a l'estat següent
            estat = 10
            #Es tanca la pinça
            mc.set_pwm_output(23,50,28)
    elif estat == 9:
        #Es calcula quina peça s'ha d'agafar del carregador
        pos = (int(info)-1)/2
        #s'envien els angles d'aquella peça
        mc.send_angles(recollircercle[pos],VELOCITAT)
        movent = True
        sleep(0.1)
        #En cas que el robot estigui en posició
        if mc.is_in_position(recollircercle[pos]):
            movent = False
            #Es passa al següent estat
            estat = 11
            #Es tanca la pinça
            mc.set_pwm_output(23,50,28)
    elif estat == 10:
        #S'envia el robot al punt d'aproximació una altra vegada
        mc.send_angles(CREUINTER,VELOCITAT)
        movent = True
        sleep(0.1)
        #Si és al punt d'aproximació es passa al següent estat
        if mc.is_in_position(CREUINTER):
            movent = False
            estat = 12
    elif estat == 11:
        #S'envia el robot al punt d'aproximació una altra vegada
        mc.send_angles(CERCLEINTER,VELOCITAT)
        movent = True
        sleep(0.1)
        #Si és al punt d'aproximació es passa al següent estat
        if mc.is_in_position(CERCLEINTER):
            movent = False
            estat = 12
```

```
226     elif estat == 12:
227         #S'envia el robot al punt d'aproximació de la graella
228         mc.send_angles(INTERSORITR, VELOCITAT)
229         movent = True
230         sleep(0.1)
231         # Si el robot és al punt d'aproximació de la graella
232         if mc.is_in_position(INTERSORITR):
233             movent = False
234             #Es passa al seguent estat
235             estat = 13
236     elif estat == 13:
237         #Es calcula la posició de la graella on es guarden els angles del destí
238         pos = fila * 3 + columna
239         #S'envien els angles corresponents
240         mc.send_angles(posicions[pos],VELOCITAT)
241         movent = True
242         sleep(0.1)
243         #En cas que el robot hagi arribat al seu destí
244         if mc.is_in_position(posicions[pos]):
245             movent = False
246             #Es passa al següent estat
247             estat = 14
248     elif estat == 14:
249         sleep(0.5)
250         #S'obre la pinça
251         mc.set_pwm_output(23,50,15)
252         estat = 15
253 
254     elif estat == 15:
255         #S'envien els angles d'aproximació a la graella altra vegada
256         mc.send_angles(INTERSORITR, VELOCITAT)
257         movent = True
258         sleep(0.1)
259         #En el cas que el robot hagi assolit la posició d'aproximació
260         if mc.is_in_position(INTERSORITR):
261             movent = False
262             #Es salta al següent estat
263             estat = 16
264     elif estat == 16:
265         #S'envia el robot a l'estat 1 amb la posició inicial
266         estat = 1
267         mc.send_angles(INICI, VELOCITAT)
268         movent = True
269         sleep(0.1)
```