

Universidad de las Fuerzas Armadas “ESPE”

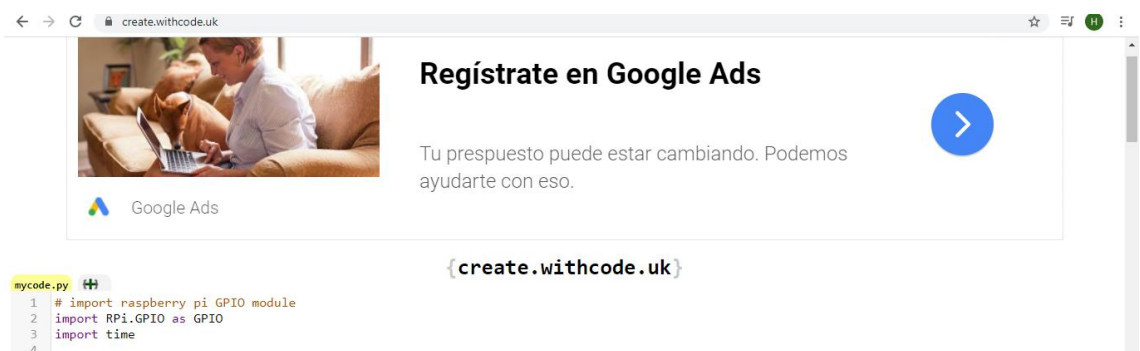
Tema: Ejercicios empleando Python y simulado en una Raspberry Pi.

Manual de usuario

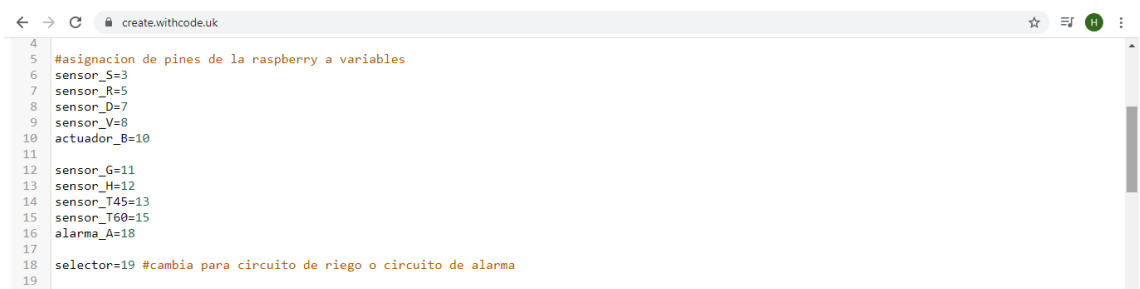
Pasos a seguir en la utilización:

Este proyecto fue creado para ayudar a comprender las principales herramientas que proporciona utilizando las librerías que son para simular dispositivos.

1. Nos dirigimos hacia la página <https://create.withcode.uk/> para crear nuestro código.
2. Después de importamos las librerías “RPi.GPIO as GPIO” y “time”.



3. Como paso siguiente vamos a realizar la asignación de pines que es la parte más importante del primer enunciado como del segundo.



4. Como paso siguiente vamos a configurar esos pines más para saber si queremos que esos sean de entrada o salida ya que con anterioridad ya asignamos los pines con sus respectivas variables.

```
22 # configuracion de pines de entrada de la raspberry
23 ##### ENUNCIADO 1 #####
24 GPIO.setup(sensor_S, GPIO.IN) #S: Señal que indica si la tierra está seca
25 #Tierra seca: S=1; Tierra húmeda: S=0
26
27 GPIO.setup(sensor_R, GPIO.IN) #R: Señal que indica si hay restricciones en el riego (es verano)
28 #Hay restricciones:R=1 No hay restricciones: R=0
29
30 GPIO.setup(sensor_D, GPIO.IN) #D: Señal que indica si es de día o de noche
31 #Día: D=1; Noche: D=0
32
33 GPIO.setup(sensor_V, GPIO.IN) #V: Señal que indica si el depósito de agua está vacío
34 #Vacío: V=1; Hay agua: V=0
35
36 GPIO.setup(actuador_B, GPIO.OUT) #B: que accionará la bomba para regar
37 # B=1; Bomba apagada B=0
38
39
40 ##### ENUNCIADO 2 #####
41 GPIO.setup(sensor_G, GPIO.IN) #G: señal de GAS
42 #G: vale '1' si se detecta GAS resultante de la combustión.
43
44 GPIO.setup(sensor_H, GPIO.IN) #H: señal de HUMO
45 #H: vale '1' si se detecta HUMO
46
47 GPIO.setup(sensor_T45, GPIO.IN) #T45: sensor temperatura T45
48 #T45: vale '1' si la temperatura es superior a 45°C
49
50 GPIO.setup(sensor_T60, GPIO.IN) #T60: sensor temperatura T60
51 #T60: vale '1' si la temperatura es superior a 60°C
52
53 GPIO.setup(alarma_A, GPIO.OUT) #A: señal de alarma
54 #A: se activará a nivel alto
55
```

5. Entonces ya acabando con los pines ya definimos nuestras clases y atributos en ambos enunciados ya que al final estos van complementarse uno con el otro, agregando que ya de una vez vamos a crear nuestro constructor para referir a nuestros atributos.

```
58
59
60 ##### ENUNCIADO 1
61 class riego:
62     #atributos
63     S=0
64     R=0
65     D=0
66     V=0
67
68     #constructor
69     def __init__(self,S,R,D,V):
70         self.S=S
71         self.R=R
72         self.D=D
73         self.V=V
74
```

```
111
112     elif modo==3:
113         print("BOMBA ENCENDIDA: sin restriccion")
114         print("V=1 Hay agua")
115         print("S=1 Tierra SECA")
116     elif modo==4:
117         print("BOMBA APAGADA: sin restriccion")
118         print("V=1 Hay agua")
119         print("S=1 Tierra HUMEDA")
120
121
122
123 ##### ENUNCIADO 2
124 class circuito:
125     #atributos
126     G=0
127     H=0
128     T45=0
129     T60=0
130
131     #constructor
132     def __init__(self,G,H,T45,T60):
133         self.G=G
134         self.H=H
135         self.T45=T45
136         self.T60=T60
137
```

6. Ahora definimos nuestras funciones y ya sean como nosotros queramos llamarle en este caso en ambos será la de operación y en cada uno de ellos entraremos en decisiones porque en los enunciados indican que se activará según estén estos valores ya que son 1 y 0.

```
← → ↺ create.withcode.uk ☆ ⌵ H ⋮
69 def __init__(self,S,R,D,V):
70     self.S=S
71     self.R=R
72     self.D=D
73     self.V=V
74
75 def operacion(self,S,R,D,V):
76
77     if V==1: #bomba vacia
78         GPIO.output(actuador_B, GPIO.LOW) #bomba apagada
79         modo=0
80
81     elif V==0: #hay agua
82
83         if R==1: #hay restricciones
84             if D==0: #es de noche
85                 GPIO.output(actuador_B, GPIO.HIGH) #bomba encendida
86                 modo=1
87             elif D==1: #es de dia
88                 GPIO.output(actuador_B, GPIO.LOW) #bomba apagada
89                 modo=2
90
91         elif R==0: #no hay restricciones
92             if S==1: #si la tierra esta seca
93                 GPIO.output(actuador_B, GPIO.HIGH) #bomba encendida
94                 modo=3
95             elif S==0: #si la tierra esta humeda
96                 GPIO.output(actuador_B, GPIO.LOW) #bomba apagada
97                 modo=4
98
99     return modo
100
```

7. Después creamos una función que va a imprimir el mansaje que nosotros deseamos que este se proyecte en pantalla y claro está que se mostrara según este almacenada en las variables.

```
← → ↺ create.withcode.uk ☆ ⌵ H ⋮
88     GPIO.output(actuador_B, GPIO.LOW) #bomba apagada
89     modo=2
90
91     elif R==0: #no hay restricciones
92         if S==1: #si la tierra esta seca
93             GPIO.output(actuador_B, GPIO.HIGH) #bomba encendida
94             modo=3
95         elif S==0: #si la tierra esta humeda
96             GPIO.output(actuador_B, GPIO.LOW) #bomba apagada
97             modo=4
98
99     return modo
100
101 def imprimir(self,modo):
102     if modo==0:
103         print("BOMBA APAGADA: No hay agua")
104     elif modo==1:
105         print("BOMBA ENCENDIDA: con restriccion")
106         print("V=1 Hay agua")
107         print("D=0 Es de NOCHE")
108     elif modo==2:
109         print("BOMBA APAGADA: con restriccion")
110         print("V=1 Hay agua")
111         print("D=0 Es de DIA")
112     elif modo==3:
113         print("BOMBA ENCENDIDA: sin restriccion")
114         print("V=1 Hay agua")
115         print("S=1 Tierra SECA")
116     elif modo==4:
117         print("BOMBA APAGADA: sin restriccion")
118         print("V=1 Hay agua")
119         print("S=1 Tierra HUMEDA")
120
```

8. Ya en nuestro programa principal utilizaremos ya nuestros objetos para poder llamar a cada uno de nuestras funciones según nosotros mismo las llamemos en el orden deseado.

```

176 #se instancian los objetos de las clases
177 manguera=riego(0,0,0,0)
178 incendio=circuito(0,0,0,0)
179
180 while (True):
181
182     if(GPIO.input(selector)==0):
183         mod=manguera.operacion(GPIO.input(sensor_S),GPIO.input(sensor_R),GPIO.input(sensor_D),GPIO.input(sensor_V))
184         manguera.imprimir(mod)
185         time.sleep(0.25)
186
187     elif(GPIO.input(selector)==1):
188         mod=incendio.operacion(GPIO.input(sensor_G),GPIO.input(sensor_H),GPIO.input(sensor_T45),GPIO.input(sensor_T60))
189         incendio.imprimir(mod)
190         time.sleep(0.25)

```

9. Y como paso final observaremos los valores que estén generando ya que no entraremos a manipular ninguno de los pines.

