

## LABORATORIO 3: Introducción a la Capa de Dispositivos usando CrowPi

### 1. OBJETIVO (S)

---

El objetivo del presente laboratorio es continuar con la exploración de las funcionalidades de la capa de dispositivos utilizando el kit de desarrollo CrowPi.

Al finalizar la práctica el estudiante estará en la capacidad de:

- Comprender los conceptos básicos de la capa de dispositivos de los sistemas IoT y su funcionamiento.
- Configurar la comunicación entre los distintos tipos de sensores y actuadores desplegados.
- Realizar Scripts que permitan la implementación de un sistema IoT en la CrowPi.

### 2. LECTURAS PREVIAS

---

- [CrowPi User Manual](#)
- [CrowPi Python Lesson](#)

### 3. INFORMACIÓN BÁSICA

---

- Durante el laboratorio deberá crear un sistema de asignación inteligente de puestos de trabajo utilizando tag RFId.
- Se recomienda leer la guía completamente antes de iniciar a resolver las actividades propuestas, con el objetivo de tener presente las actividades y los entregables a desarrollar.

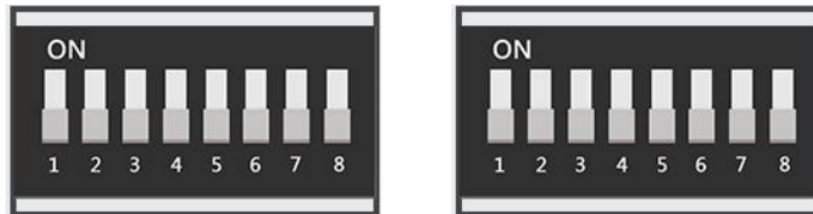
#### 3.1 Acerca de la CrowPi

CrowPi es un kit educativo de electrónica "todo en uno" ampliamente utilizado en la creación rápida de prototipos de ideas basadas en Raspberry Pi, o simplemente como un ordenador portátil. Los sensores y actuadores que se muestran en la Figura 1 coinciden con los números de la placa CrowPi mostrados en la Tabla 1.



## Conmutación entre los módulos

La CrowPi contiene 2 "interruptores" (Figura 2) cada uno de estos tiene 8 pines, para un total de 16 pines – los cuales están identificados con el número 25 en la Figura 1. Estos interruptores permiten cambiar entre el uso de sensores y módulos, esto debido a que la RaspberryPi sólo soporta un número limitado de pines GPIO; así que a través de la utilización de estos interruptores se puede ampliar la capacidad. El uso de estos interruptores es bastante fácil y será necesario en algunas de las lecciones anteriores.



*Figura 2. Interruptores (25)*

Cuando se requiere utilizar uno de estos dispositivos, se deberá realizar un ajuste en los interruptores para un correcto funcionamiento. Así mismo, se recomienda que después de realizada la práctica, o si no estás usando ese sensor, se deben colocar los interruptores en su posición original ya que otros dispositivos pueden usar esos mismos pines GPIO.

Los siguientes dispositivos requieren el uso de interruptores y el cambio de los mismos:

- Matriz de botones (22): se deben activar todos los pines del interruptor izquierdo
- Botón independiente (23): se deben activar todos los pines del interruptor izquierdo
- Sensor de vibración (8): se debe activar el primer pin del interruptor derecho
- Sensor de inclinación (17): se debe activar el segundo pin del interruptor derecho
- Motor paso a paso (16): se deben activar los pines 3, 4, 5, y 6 del interruptor derecho
- Servos (14 - 15): se deben activar los pines 7, y 8 del interruptor derecho

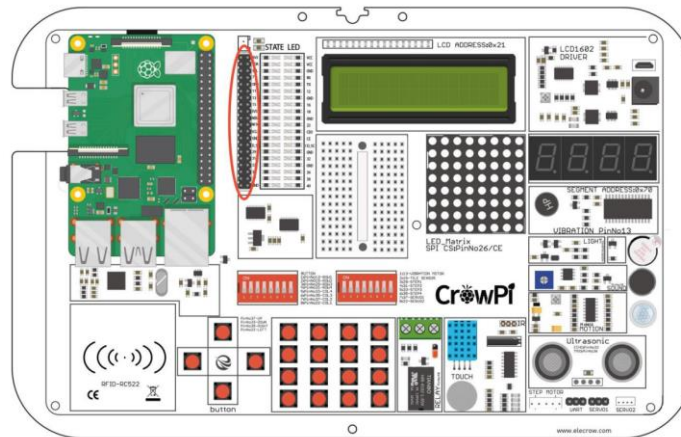
## Lenguaje de Programación

Teniendo en cuenta que la CrowPi cuenta con placas RaspberryPi tiene soporte para ejecutar scripts en Python.

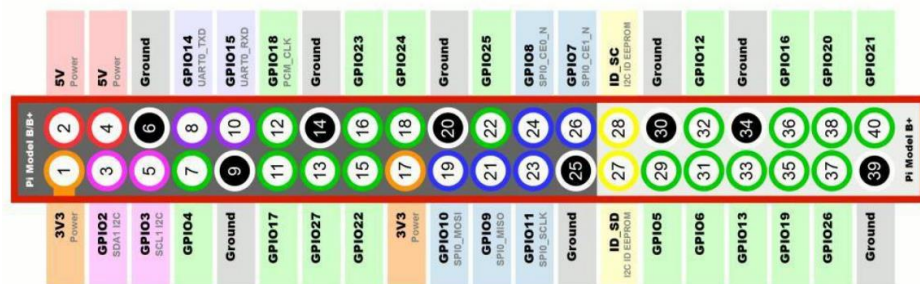
## Un vistazo a los fundamentos de la utilización de GPIO

Los pines GPIO (General-purpose input/output) son un pin **genérico** en un chip, cuyo comportamiento (incluyendo si es un pin de entrada o salida) se puede controlar (programar) por el usuario en tiempo de ejecución. En otras palabras, los pines GPIO no tienen un propósito específico dedicado, pueden ser configurados como pines de entrada o pines de salida y su propósito general depende de lo que se quiera lograr.

- Ejemplo de pin de entrada: un botón se consideraría de entrada, porque se hace clic en él.
- Ejemplo para el pin de salida: un buzzer se consideraría de salida, porque recibe una señal para vibrar.



*Figura 3. Ubicación de los Pines GPIO*



*Figura 4. Esquemas de GPIO en una Raspberry Pi*

## 4. PROCEDIMIENTO

---

<sup>1</sup> <https://pypi.org/project/RPi.GPIO/>

## 4.1. Configuración de la lectura de Tarjetas RFID

Como parte del programa de digitalización de laboratorios de la Universidad de los Cerros, se pretende llevar a cabo un piloto de prueba de estaciones de trabajo en la cual se guarde toda la información correspondiente de la persona que desea ingresar en el laboratorio. Para esto usted posee los siguientes códigos encargados de realizar escritura y lectura en las tarjetas de los potenciales usuarios.

```
1 import RPi.GPIO as GPIO
2 import MFRC522
3 import signal
4 from mfrc522 import SimpleMFRC522
5
6 continue_reading = True
7
8
9 def end_read(signal, frame):
10     global continue_reading
11     print("Ctrl+C captured, ending read.")
12     continue_reading = False
13     GPIO.cleanup()
14
15 signal.signal(signal.SIGINT, end_read)
16 # create the reader object
17 MIFAREReader = MFRC522.MFRC522()
18 reader = SimpleMFRC522()
19
20 def griteRegistro():
21     print("=====\n Agregar registro\n=====")
22     nombres = input('Ingrese el texto a escribir\n')
23
24     print("Coloque el chip RFID sobre el lector")
25
26     reader.write(text)
27     print("Escrito")
28     menu = "Menú de opciones\n\t1. Leer registros.\n\tIngrese la opción que desea ejecutar (ej: 1)\n"
29
30 def showTerminal():
31     while True:
32         option = input(menu)
33         if option == "1":
34             print("Leyendo registro")
35             readRFID()
36         elif option == "2":
37             griteRegistro()
38         else:
39             print("Opción no válida.")
40     showTerminal()
```

*Figura 5. Código de escritura con RFID*

```
1 import RPi.GPIO as GPIO
2 import MFRC522
3 import signal
4 from mfrc522 import SimpleMFRC522
5
6 reader = SimpleMFRC522()
7
8 continue_reading = True
9
10
11 def end_read(signal, frame):
12     global continue_reading
13     print("Ctrl+C captured, ending read.")
14     continue_reading = False
15     GPIO.cleanup()
16
17 signal.signal(signal.SIGINT, end_read)
18 # create the reader object
19
20 def readRFID():
21     print("Acerque la tarjeta al lector")
22     id, text = reader.read()
23     print(text)
24
25
26 menu = "Menú de opciones\n\t1. Leer registros.\n\tIngrese la opción que desea ejecutar (ej: 4)\n"
27
28 def showTerminal():
29     while True:
30         option = input(menu)
31         if option == "1":
32             print("Leyendo registro")
33             readRFID()
34         else:
35             print("Opción no válida.")
36     showTerminal()
```

*Figura 6. Código de lectura con RFID*

## Trabajo Propuesto

Con la finalidad de crear un MVP del sistema se propone que se utilice los kits Crowpi disponibles en el laboratorio de Comunicaciones de la universidad. El MVP debe permitir las siguientes funcionalidades:

1. Implementar la lectura de tarjetas RFID dentro de la Crowpi.
2. Implementar la escritura de las tarjetas RFID dentro de la Crowpi.
3. Guardar información vital de la persona que posee estas tarjetas (nombres, apellidos, cargo, código, edad).

## 4.2. Configuración de la escritura en el Módulo LED

Una vez realizado la lectura y escrituras de las tarjetas para los usuarios, usted decide utilizar la matriz 8x8 led que se encuentra disponible dentro de la Crowpi. Con esta matriz de led usted debe darles la bienvenida a los usuarios.

Para esto usted dispone del siguiente código:

```
1  # Import all the modules
2  import re
3  import time
4  from luma.led_matrix.device import max7219
5  from luma.core.interface.serial import spi, noop
6  from luma.core.render import canvas
7  from luma.core.virtual import viewport
8  from luma.core.legacy import text, show_message
9  from luma.core.legacy.font import proportional, CP437_FONT, TINY_FONT, SINCLAIR_FONT, LCD_FONT
10
11
12  def main(cascaded, block_orientation, rotate):
13
14      # create matrix device
15      serial = spi(port=0, device=1, gpio=noop())
16      device = max7219(serial, cascaded=cascaded or 1, block_orientation=block_orientation, rotate=rotate or 1)
17      # debugging purpose
18      print("[~] Matrix initialized")
19
20      # print hello world on the matrix display
21      msg = "Y que no me digan en la esquina 'EL VENAO , EL VENAO' que eso a mi me mortifica"
22      # debugging purpose
23      print("[~] Printing: %s" % msg)
24      show_message(device, msg, fill="red", font=proportional(CP437_FONT), scroll_delay=0.1)
25
26
27  if __name__ == "__main__":
28
29      # cascaded = Number of cascaded MAX7219 LED matrices, default=1
30      # block_orientation = choices 0, 90, -90, Corrects block orientation when wired vertically, default=0
31      # rotate = choices 0, 1, 2, 3, Rotate display 0=0°, 1=90°, 2=180°, 3=270°, default=0
32
33      try:
34          main(cascaded=0, block_orientation=180, rotate=0)
35      except KeyboardInterrupt:
36          pass
```

*Figura 7. Código de lectura con RFID*

## Trabajo Propuesto

Para realizar esto usted debe realizar las siguientes operaciones:

1. Implementar el código en la Crowpi.
2. Integrar el código para que dé la bienvenida al usuario con los datos que se encuentran en el lector RFID.

## 5. Reto

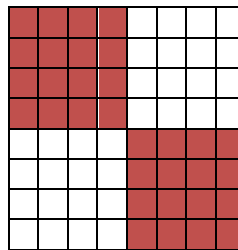
---

Como extensión al trabajo desarrollado en la parte anterior del reto se propone que el grupo muestre en la matriz de LEDS el puesto de trabajo asignado al empleado que acaba de ingresar a la oficina.

Las reglas para la asignación del puesto de trabajo son las siguientes:

1. Los puestos de trabajo iniciaran su numeración en 1.
2. Al ingresar un empleado se le asignará el primer puesto libre que se encuentre en la oficina; por ejemplo, al ingresar el primer empleado se le asignará el primer puesto.
3. Al salir un empleado se liberará el puesto de trabajo asignado.

El número de puestos de trabajo a asignar para este MVP es cuatro (4), por lo cual debe segmentar la matriz de leds en cuatro (4) cuadrantes para diferencia cada puesto. Por ejemplo, si los puestos uno (1) y (4) están asignados se debe encender la matriz de leds de la siguiente manera:



## 6. Preguntas

---

A continuación, encontrará una serie de preguntas que le permitirán aplicar, afianzar y profundizar los conocimientos adquiridos durante las sesiones de clase dentro del desarrollo de este laboratorio.

1. Tomando como base la implementación del sistema IoT realizada ¿El caso cumple con el ciclo de valor de la información?, en caso de qué considere lo contrario ¿Qué le haría falta para cumplir con este ciclo?
2. Teniendo en cuenta la implementación del sistema IoT realizada, identifique los posibles riesgos de seguridad que presenta el sistema.
3. A partir de los datos recolectados por el sistema IoT ¿Qué tipo de análisis se podría realizar para la organización? Explique brevemente cada análisis.



## 6. Entregables

---

El entregable para este laboratorio es:

1. Un archivo .zip con los scripts de Python realizados. El nombre del archivo debe tener formato *<grupo>\_Lab3.zip*
2. Un archivo .pdf donde se documente los cambios realizados para realizar el reto propuesto y donde se respondan las preguntas propuestas. El nombre del archivo debe tener formato *<grupo>\_reporte\_Lab3.pdf*

Estos archivos deben ser entregados utilizando el enlace habilitado en Bloque Neón para tal fin.

## 7. EVALUACIÓN

---

La evaluación de este laboratorio será distribuida de la siguiente manera:

Entregable	Valor
Tutorial	30 %
Reto	40 %
Reporte	30 %
<b>Total</b>	100 %

### HISTORIAL DE REVISIONES

#### Registro de Cambios

Fecha	Autor(es)	Versión	Referencia de Cambios
01/09/2022	Ramón Arias Carlos Lozano	0.1	Versión Inicial

#### Revisores

Nombre	Autor(es)	Institución
Carlos Lozano	Profesor Asistente Departamento de Ingeniería de Sistemas y Computación	Universidad de los Andes