

Implementación de dos Programas en una Raspberry pi por medio del lenguaje de programación Python

Elian Toapanta, Iván Zambrano y Marlon Torres

Resumen - En este documento se presentan la utilización de los pines de la Raspberry Pi para poder usarlos en dos programas de lógica por medio del lenguaje de programación Python

Índice de Términos – Python, POO, OPP, lógica, GPIO, raspberry pi.

I. INTRODUCCIÓN

Node-RED es una herramienta de programación para conectar dispositivos de hardware, API y servicios en línea de formas nuevas e interesantes.

Python fue creado a finales de los ochenta por Guido van Rossum en el Centro para las Matemáticas y la Informática (CWI, Centrum Wiskunde & Informatica), en los Países Bajos, como un sucesor del lenguaje de programación ABC, capaz de manejar excepciones e interactuar con el sistema operativo Amoeba.

II. PYTHON

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. (Python, n.d.).

Python fue creado a finales de los ochenta por Guido van Rossum en el Centro para las Matemáticas y la Informática (CWI, Centrum Wiskunde & Informatica), en los Países Bajos, como un sucesor del lenguaje de programación ABC, capaz de manejar excepciones e

interactuar con el sistema operativo Amoeba. El nombre del lenguaje proviene de la afición de un grupo británico de comediantes conocidos como Monty Python (Python, n.d.).

III. RASPBERRY PI

Raspberry PI es una placa computadora (SBC) o Pc de placa única, de bajo precio, se podría expresar que es un ordenador de tamaño reducido, del orden de una tarjeta de crédito, desarrollado en el Reino Unido por la Fundación Raspberry PI (Universidad de Cambridge) en 2011, con el objetivo de estimular la enseñanza de la informática en las escuelas, aunque no empezó su comercialización hasta el año 2012.

A. Puerto GPIO de la RASPBERRY PI

La placa Raspberry Pi puede comunicarse con dispositivos externos a través del conector GPIO incorporado. En este conector hay pines de alimentación integrados (+5 y +3.3 V), tierra y pines de entrada / salidas capaces de implementar diferentes protocolos.

B. Los pines GPIO tienen funciones específicas:

BOARD	GPIO	GPIO	BOARD
01	3.3v DC Power		DC Power 5v 02
03	GPIO02 (SDA1 , I ² C)		DC Power 5v 04
05	GPIO03 (SCL1 , I ² C)		Ground 06
07	GPIO04 (GPIO_GCLK)		(TXD0) GPIO14 08
09	Ground		(RXD0) GPIO15 10
11	GPIO17 (GPIO_GEN0)		(GPIO_GEN1) GPIO18 12
13	GPIO27 (GPIO_GEN2)		Ground 14
15	GPIO22 (GPIO_GEN3)		(GPIO_GEN4) GPIO23 16
17	3.3v DC Power		(GPIO_GEN5) GPIO24 18
19	GPIO10 (SPI_MOSI)		Ground 20
21	GPIO09 (SPI_MISO)		(GPIO_GEN6) GPIO25 22
23	GPIO11 (SPI_CLK)		(SPI_CE0_N) GPIO08 24
25	Ground		(SPI_CE1_N) GPIO07 26
27	ID_SD (PC ID EEPROM)		(PC ID EEPROM) ID_SC 28
29	GPIO05		Ground 30
31	GPIO06		GPIO12 32
33	GPIO13		Ground 34
35	GPIO19		GPIO16 36
37	GPIO26		GPIO20 38
39	Ground		GPIO21 40

Fig. 1. Esquema pines GPIO

- El color amarillo (2): Alimentación a 3.3V.
- El color rojo (2): Alimentación a 5V.
- El color naranja (26): Pueden configurarse como entradas o salidas se las conoce también como pines de Propósito General.
- El color gris (2): Son pines reservados.
- El color negro (8): Conexión a GND.
- El color azul (2): Comunicación mediante el protocolo I2C (Circuito inter-integrado).
- El color verde (2): Destinados a conexión para UART para puerto serie convencional.
- El color morado (5): Comunicación mediante el protocolo SPI (Es un protocolo síncrono, el cual sincroniza y transmite datos por medio de 4 señales).

IV. POO

La Programación Orientada a Objetos (POO u OOP según sus siglas en inglés) es un paradigma de programación que usa objetos y sus interacciones para diseñar aplicaciones y programas de computadora. Está basado en varias técnicas, incluyendo herencia, modularidad, polimorfismo, y encapsulamiento. Su uso se popularizó a principios de la década de 1990. Actualmente son muchos los lenguajes de programación que soportan la orientación a objetos.

A. Clase

Definiciones de las propiedades y comportamiento de un tipo de objeto concreto. La instanciación es la lectura de estas definiciones y la creación de un objeto a partir de ellas.

B. Objeto

Instancia de una clase. Entidad provista de un conjunto de propiedades o atributos (datos) y de comportamiento o funcionalidad (métodos), los mismos que consecuentemente reaccionan a eventos. Se corresponden con los objetos reales del mundo que nos

rodea, o con objetos internos del sistema (del programa). Es una instancia a una clase.

C. Método

Algoritmo asociado a un objeto (o a una clase de objetos), cuya ejecución se desencadena tras la recepción de un “mensaje”. Desde el punto de vista del comportamiento, es lo que el objeto puede hacer. Un método puede producir un cambio en las propiedades del objeto, o la generación de un “evento” con un nuevo mensaje para otro objeto del sistema.

D. Atributos

Características que tiene la clase.

E. Componentes de un objeto

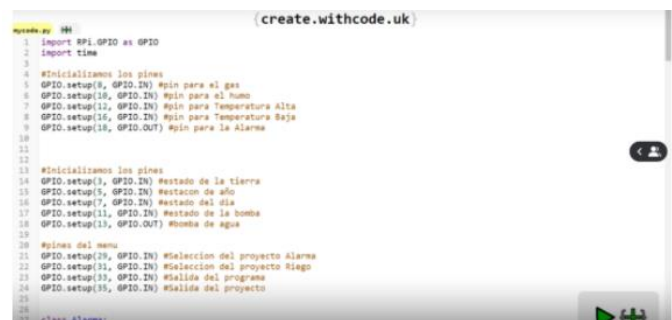
Atributos, identidad, relaciones y métodos.

F. Identificación de un objeto

Un objeto se representa por medio de una tabla o entidad que esté compuesta por sus atributos y funciones correspondientes.

V. CREATE.WITHCODE.UK ONLINE

Create.withcode.uk es un editor de python en línea que le permite escribir, ejecutar, depurar y compartir código de python en su navegador web. Se lanzó en abril de 2016 para ayudar a los estudiantes a aprender a escribir código en cualquier dispositivo sin tener que instalar ningún software especializado.



```

1 import RPi.GPIO as GPIO
2 import time
3
4 #Iniciamos los pines
5 GPIO.setup(8, GPIO.IN) #pin para el gas
6 GPIO.setup(18, GPIO.IN) #pin para el humo
7 GPIO.setup(12, GPIO.IN) #pin para Temperatura Alta
8 GPIO.setup(16, GPIO.IN) #pin para Temperatura Baja
9 GPIO.setup(15, GPIO.OUT) #pin para la Alarma
10
11
12
13 #Iniciamos los pines
14 GPIO.setup(3, GPIO.IN) #estado de la tierra
15 GPIO.setup(5, GPIO.IN) #estado de año
16 GPIO.setup(7, GPIO.IN) #estado del día
17 GPIO.setup(11, GPIO.IN) #estado de la bomba
18 GPIO.setup(13, GPIO.OUT) #bomba de agua
19
20 #pines del menu
21 GPIO.setup(29, GPIO.IN) #selección del proyecto Alarma
22 GPIO.setup(31, GPIO.IN) #selección del proyecto Riego
23 GPIO.setup(33, GPIO.IN) #salida del programa
24 GPIO.setup(35, GPIO.IN) #salida del proyecto
25
26
27 class Alarma:

```

Fig.2. Presentación de un de Create.withcode.uk

VI. PROGRAMA DE LA ALARMA

Se quiere realizar un circuito para activar la alarma de incendios (A) para la evacuación de un edificio. Para ello se tiene un sensor de gases (G), un sensor de humos

(H), y dos señales procedentes de un termómetro que indican si la temperatura es mayor de 45°C (T45) y si la temperatura es mayor de 60°C (T60). Debido a que a veces los sensores detectan humos y gases que no siempre proceden de incendios (por ejemplo, de los cigarrillos o las cocinas), para evitar falsas alarmas, la señal A se activará cuando se cumplan las siguientes condiciones:

- a. Si la temperatura es mayor de 60°C siempre se activará la alarma
- b. Si la temperatura está entre 45°C y 60°C se activará la alarma sólo si han detectado gases o humos (o ambos).
- c. Si la temperatura es menor de 45°C se activará la alarma sólo si se detectan gases y humos

Resumiendo, las 4 señales binarias de entrada y la salida:

- G: vale '1' si se detecta GAS resultante de la combustión.

- H: vale '1' si se detecta HUMO.

- T45: vale '1' si la temperatura es superior a 45°C

- T60: vale '1' si la temperatura es superior a 60°C

La señal de salida A (alarma) se activará a nivel alto REQUERIMIENTOS.

- Se debe implementar los 2 enunciados empleando Python y simulado en una Raspberry Pi.
- No se debe ingresar datos por consola.
- El programa debe presentar en pantalla información relacionada a lo que se está realizando.
- Se debe emplear conceptos de POO.
- Cada enunciado se implementará en una clase con sus funciones.
- Se debe emplear un programa de selección (clase) que permita seleccionar la ejecución de los programas.
- El programa principal crear objetos de cada clase y llamar a las funciones que correspondan para que opera la lógica de negocio del problema.

VII. PROGRAMA DE RIEGO AUTOMÁTICO

Este programa deseaba hacer un circuito que permitirá realizar un riego automático para ello primero se creó una clase dentro de esta clase se definió las funciones las funciones vienen hacer las entradas del enunciado del problema tenemos la señal de tierra seca la señal de restricciones la señal que indica si es de día o de noche la señal que indica si el depósito de agua está vacío o lleno una vez definidas las funciones c añadió que no una de ellas a un pin del arroz pero que viene a ser la entrada mientras que la salida viene a ser el funcionamiento de la bomba para las condiciones se utilizó funciones if-else y dependiendo la condición que nos arrojaba en el enunciado se la aplican estas condiciones serán :

i. Para evitar que la bomba se estropee por funcionar en vacío, nunca se accionará la bomba cuando el depósito de agua esté vacío.

ii. Si hay restricciones en el riego (época de verano), sólo se podrá regar de noche.

iii. En el resto del año (si no hay restricciones) se podrá regar de día y de noche (si la tierra está seca)

Una vez realizadas las condiciones se creará un apartado principal y se iniciará con un guíey que permitirá hacer un bucle se colocará las funciones que muestren la condición y las variables.

VIII. CONCLUSION

- Python nos permite como programadores formar un código simple y ordenado. Sus aplicaciones, tanto en la comunidad docente como en la científica, le permitirán aumentar su popularidad y adopción a nivel internacional. Sin mencionar que gracias a sus librerías nos permiten la interacción de muchos apartados, como lo fue en este proyecto con la raspberry pi.
- El futuro de la tecnología es muy prometedor, pero necesitamos hacer un esfuerzo. Raspberry Pi está pensado para que todo el mundo pueda aprender a programar y sea capaz de introducirse en las ciencias de la computación.
- La página <https://create.withcode.uk/> la cual puedes crear funciones para ingresar y sacar datos, tiene muchas limitaciones de varias librerías y una de esas es la de librería SYS.

REFERENCIAS

- [1] Rupil, A. C., & García Mattío, M. (2019). Node-red, conectando cosas, APIs y servicios en línea.
- [2] Sánchez, M., Barrena, M., Bustos, P., Campillo, C., & García, P. (2016). Arquitectura software basada en tecnologías smart para agricultura de precisión. Jornadas de Ingeniería del Software y Bases de Datos.
- [3] Oliphant, T. E. (2007). Python for scientific computing. Computing in Science & Engineering, 9(3), 10-20.

- [4] Blackstock, M., & Lea, R. (2014, October). Toward a distributed data flow platform for the web of things (distributed node-red). In Proceedings of the 5th International Workshop on Web of Things (pp. 34-39).
- [5] Lekić, M., & Gardašević, G. (2018, March). IoT sensor integration to Node-RED platform. In 2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH) (pp. 1-5). IEEE.