

Sistema de riego automático y Alarma contra incendios construido en Raspberry pi

Chanchay Randy

rfchanchay@espe.edu.ec

Zurita Dilan

dezurita4@espe.edu.ec

Lema Erick

ejlema2@espe.edu.ec

Resumen – El siguiente documento presenta la construcción de 2 sistemas de control en un solo programa los cuales permiten el control de un sistema de riego autónomo y una alarma contra incendios, todo esto implementado en una raspberry pi.

Palabras Clave: Raspberry, Python, POO, Gpio.

ABSTRACT: The following document presents the construction of 2 control systems in a single program which allow the control of an autonomous irrigation system and a fire alarm, all this implemented in a raspberry pi.

Keywords: Raspberry, Python, POO, Gpio.

Introduccion

El ordenador (en este caso la Raspberry, que es una serie de ordenadores) necesita dispositivos que extraigan datos del exterior y los meta dentro de él para poder operar con ellos. Estos componentes se conocen como elementos de entrada de datos.

En nuestro algoritmo, los datos de entrada serán recibidos mediante el puerto Gpio que las señales de entrada serán exteriores a otros elementos electrónicos.

La información de salida se puede presentar en diferentes maneras que sean comprensibles para un usuario ya sea de manera visual o representada en cualquier dispositivo electrónico.

En resumidas cuentas, la entrada de datos nos sirve como materia prima para lo que queremos representar posteriormente, y dicha representación es la salida la cual es información útil.[1]

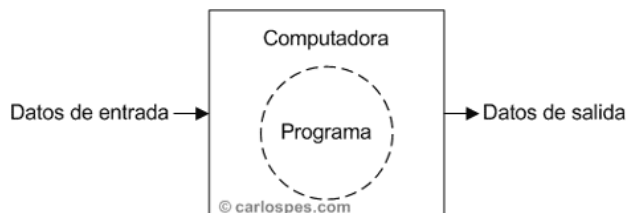


Ilustración 1 Entrada y salida de Datos

1. GPIO

General Purpose Input Output (GPIO) es un sistema de entrada y salida de propósito general, es decir, consta de una serie de pines o conexiones que se pueden usar como entradas o salidas para múltiples usos. Estos pines están incluidos en todos los modelos de Raspberry Pi con leves diferencias.[2]

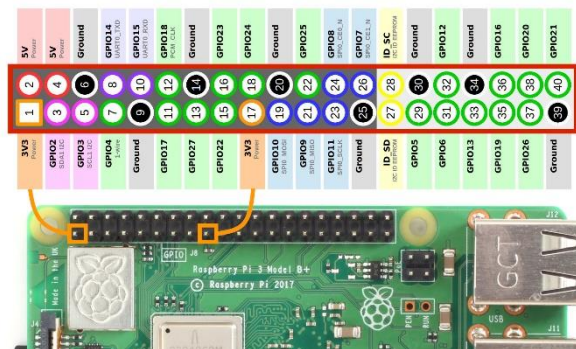


Ilustración 2 Pines GPIO

En el puerto GPIO disponemos pines de: entrada, salida, comunicación, alimentación.

En nuestro proyecto la manipulación de estos pines se dará mediante programación, en este caso Python el cual es un lenguaje de programación compatible con raspberry y de muy fácil uso, para esto usaremos la librería RPi.GPIO.

Las funciones usadas en nuestro código para la manipulación del GPIO son:

GPIO.input() para entrada de datos.

GPIO.output() para salida de datos

GPIO.setmode() para iniciar la GPIO

GPIO.setup() para configurar la GPIO

2. POO en Python

En python la POO se expresa de manera simple y fácil de escribir pero debes tener en cuenta que para programar debes entender cómo funciona la teoría de POO y aplicarla al código.

La teoría de la POO nos dice que todos los objetos deben pertenecer a una clase, ya que esta es la base para diferenciarse unos de otros teniendo atributos y comportamientos que los distinguen de otros objetos que pertenezcan a otras clases.[3]

Para nuestro programa contaremos con las siguientes clases:

Bomba: para el sistema de riego

Alarma: para la alarma contra incendios

Selección: para controlar el modo de funcionamiento del sistema

Las clases Bomba y Alarma contarán con las siguientes funciones que serán las mismas:

apagado(self): apagar nuestra salida

encendido(self): encender nuestra salida

analizar(self): analizar estado de pines

casos(self,caso): seleccionar nuestros casos de ejecución

mensaje(self,mens): mostrar mensajes

La clase selección contará con las siguientes funciones:

iniciarGPIO(self): inicio de GPIO

tempo(self): mensaje inicial

elegir(self): selección de modo de operación

3. Funcionamiento de las clases

3.1. Bomba

Esta clase nos solicita diseñar un sistema de riego en el cual se tienen entradas mediante sensores, donde influyen aspectos como el día, la hora, la época, y el estado del tanque.

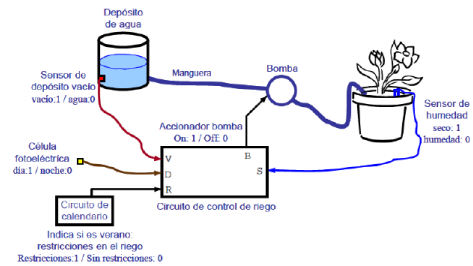


Ilustración 3 Diagrama Bomba

Para la realización de este sistema podemos aplicar tablas de verdad para la activación o desactivación de la bomba de agua.

A: Sensor de Humedad
B: Sensor Deposito
C: Circuito Calendario
D: Célula Fotoeléctrica
Y: Salida

	A	B	C	D	Y
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	1
11	1	0	1	1	0
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	0

Ilustración 4 Tabla de verdad Bomba

3.2. Alarma

Esta clase nos solicita diseñar una alarma contra incendios en el cual se tienen entradas mediante sensores, donde influyen aspectos como la temperatura, gases y humo.

Para la realización de este sistema podemos aplicar tablas de verdad para la activación o desactivación de la alarma.

T60:A '1' si la temperatura es superior a 60°C
T45:B '1' si la temperatura es superior a 45°C
Sensor de humos :C '1' si se detecta HUMO.
Sensor de Gases : D '1' si se detecta GAS resultante de la combustión.
Salida: Y La señal de salida A (alarma) se activará a nivel alto.

	A	B	C	D	Y
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	1

Ilustración 5 Tabla de verdad Alarma

3.3. Selección

Este programa leerá un pin de entrada de la GPIO y dependiendo del estado de este seleccionara una u otra clase donde 0 es la bomba y 1 es la alarma.

4. Conclusiones

Una vez finalizado el proyecto podemos concluir que gracias a la ayuda de la Raspberry Pi y de un buen conocimiento de programación en Python se puede generar resultados increíbles, en este caso se alcanzó los objetivos planteados ya que mediante POO y lógica booleana en el código, se pudo desarrollar un control de sistemas de automatización que funciona a la perfección.

Y con esto no solo pudimos demostrar su utilidad, sino que también pudimos evidenciar el potencial de estos sistemas y lo que se podría hacer con métodos similares.

5. Recomendaciones

- Para implementar el circuito en un armado real, armar el circuito sin alimentar el raspberry.
- Considerar los pines de gpio que permitan el trabajo exclusivo de entrada o salida, evitando el uso de pines como el i2c o uart.
- Si el funcionamiento de cada programa es parecido implementar las mismas funciones para cada clase.

6. Bibliografía y fuentes.

- [1] D. D. E. I. Electrónica, *Universidad politécnica de madrid escuela técnica superior de ingenieros de telecomunicación departamento de ingeniería electrónica.* .
- [2] “¿Qué es GPIO? - Control de GPIO con Python en Raspberry Pi.” [Online]. Available: <https://www.programoergosum.com/cursos-online/raspberry-pi/238-control-de-gpio-con-python-en-raspberry-pi/que-es-gpio>. [Accessed: 08-Sep-2020].
- [3] “POO Programación Orientada a Objetos en Python - ▷ Cursos de Programación de 0 a Experto © Garantizados.” [Online]. Available: <https://unipython.com/programacion-orientada-objetos-python/>. [Accessed: 08-Sep-2020].