

Node-RED & Calculadora Científica en Python

Autores: Paspuel C. Christian A.

Topón S. Kevin D.

Simba C. Henry J.

Estudiantes de la Universidad de las Fuerzas Armadas ESPE Sangolquí/Ecuador.

Departamento de Ciencias De la Computación.

Fax: +593 23989-400. capaspuel1@espe.edu.ec

Estudiantes de la Universidad de las Fuerzas Armadas ESPE Sangolquí/Ecuador.

Departamento de Ciencias De la Computación.

Fax: +593 23989-400. kdtopon@espe.edu.ec

Estudiantes de la Universidad de las Fuerzas Armadas ESPE Sangolquí/Ecuador.

Departamento de Ciencias De la Computación.

Fax: +593 23989-400. hjsimba@espe.edu.ec

Resumen

Node-RED es un editor de flujo basado en el navegador donde se puede añadir o eliminar nodos y conectarlos entre sí con el fin de hacer que se comuniquen entre ellos que en este ejemplo demostraremos una interfaz Hombre - Máquina utilizando la librería dashboard por otro lado tenemos la creación de una calculadora Científica simple utilizando PYTHON que es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código. Se trata de un lenguaje de programación multi-paradigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional.

Introducción

En el presente trabajo vamos ya hacer una implementación de los temas visto en clase y de los trabajos que se ha realizado con anterioridad, iniciando con uno de ellos que es node-red que correspondió a realizar una interfaz HMI, que esta pue servir como elementos bases hacia nosotros los estudiantes que recién estamos iniciando, y el otro trabajo fue de una calculadora científica, pero en esta se debe simular los pines de una raspberry y además de ingresar una nueva librería como lo es math.

Palabras clave: librería, interfaz máquina-usuario(HMI), Raspberry, Node-Red

Abstract

In the present work we are going to make an implementation of the topics seen in class and of the work that has been done previously, starting with one of them that is node-network

that corresponded to make an HMI interface, which can serve as elements bases towards us the students that we are just starting, and the other work was from a scientific calculator, but in this you must simulate the pins of a raspberry and in addition to entering a new library such as math.

Keywords: library, machine-user interface (HMI), Raspberry, Node-Red **Bases Teóricas**

Node-Red

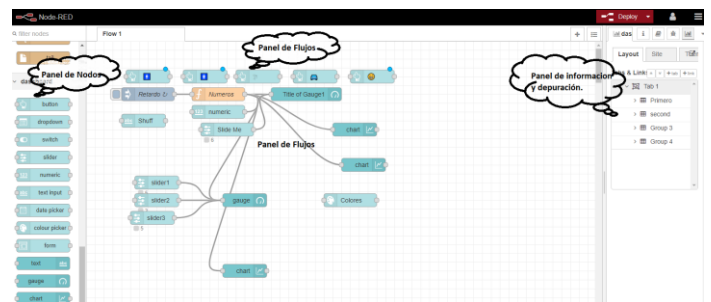
Según (Lekić & Gardašević, 2018), Node-RED es una herramienta de desarrollo basada en flujo de código abierto para la integración de dispositivos de hardware IoT, API(interfaces de programación de aplicaciones) y servicios en línea desarrollados por IBM.

Node-RED es una herramienta gratuita basada en JavaScript, construida en la plataforma Node.js que proporciona un editor visual de flujo basado en un navegador. El sistema contiene nodos que están representados por los iconos apropiados. Y funciona de dos maneras: arrastrar, soltar y conectar nodos, o importar código JavaScript.

El nodo proporciona diferentes funciones, como monitorear el flujo como depurar nodo, o para leer y escribir con pines GPIO de Raspberry Pi, Los flujos creados se almacenan utilizando JSON (Notación de objetos JavaScript). Node-Red Permite a los desarrolladores conectarse a la entrada y salida además de procesar nodos para crear flujos para el procesamiento de datos, controlar cosas o enviar alertas funciona con el principio de permitir la conexión de servicios web o personalizar nodos entre sí o con otras cosas, para realizar funciones como enviar datos de sensores por correo electrónico o servicios como el de realizar análisis complejos.

Node-RED contiene tres componentes básicos:

1. Panel de nodos.
2. Panel de flujos
3. Panel de información y depuración.



Node-RED es una herramienta poderosa y flexible que se utiliza para crear prototipos, siendo la esencia de esta herramienta permitir a los ingenieros y técnicos simplemente crear y configurar aplicaciones en tiempo real en dispositivos finales.

Además, (Coen-Porisini, 2018) propone que Node-RED es una herramienta de programación basada en flujos y basada en eventos que pueden comunicarse entre sí y regular el flujo de la información dentro del sistema diseñado. Es una representación visual basada en el navegador ayudando a los desarrolladores a comprender mejor las interacciones que ocurren dentro de toda la red IoT interactuando con entidades tanto de hardware como Sensores y de software como servicios, de igual manera destaca sobre la interfaz que nos brinda Node-

RED además de permitir la integración de diferentes tecnologías como:

MongoDB: que es un motor de base de datos no relacional debido a que su mayor eficiencia en responder en muy poco tiempo

Java: es utilizado aquí debido a su amplia adopción en implementaciones reales.

Transporte de telemetría de la cola de mensajes (MQTT): se utiliza como un método ligero de publicación y suscripción para compartir información y notificaciones del sistema además de controlar el flujo de datos que proporciona la lógica de las aplicaciones IoT.

Python

Según (Rossum, 1995) Python es un lenguaje de programación simple pero potente e interpretado diferente a la programación en C y Shell ideal para la programación desechable y la creación rápida de prototipos. Su sintaxis se elabora a partir de construcciones tomadas de una variedad de otros idiomas, el intérprete de Python se amplía fácilmente con nuevas funciones y tipos de datos implementados en C. Python de igual manera es adecuado como un lenguaje de extensión para aplicaciones C altamente personalizables para varios sistemas operativos como por ejemplos UNIX (incluido Linux), el sistema operativo Apple Macintosh, MSDOS describiéndolo a esta sintaxis y la semántica central del lenguaje como consiso pero intenta ser exacto y completo.

De igual manera (Duque, 2009) argumenta que Python es un lenguaje de

programación creado por Guido Van Rossum a principios de los años 90 cuyo nombre está inspirado en el grupo de cómicos ingleses “Monty Python”. Es un lenguaje similar a Perl, pero con una sintaxis muy limpia y que favorece un código legible.

Se trata de un lenguaje interpretado o de script, con tipado dinámico, fuertemente tipado, multiplataforma y orientado a objetos.

La ventaja de los lenguajes compilados es que su ejecución es más rápida. Sin embargo, los lenguajes interpretados son más flexibles y más portables. Python tiene, no obstante, muchas de las características de los lenguajes compilados, por lo que se podría decir que es semi interpretado. En Python, como en Java y muchos otros lenguajes, el código fuente se traduce a un pseudo código máquina intermedio llamado bytecode la primera vez que se ejecuta, generando archivos .pyc o .pyo (bytecode optimizado), que son los que se ejecutarán en sucesivas ocasiones.

Tipado dinámico

La característica de tipado dinámico se refiere a que no es necesario declarar el tipo de dato que va a contener una determinada variable, sino que su tipo se determinará en tiempo de ejecución según el tipo del valor al que se asigne, y el tipo de esta variable puede cambiar si se le asigna un valor de otro tipo.

Orientado a objetos

La orientación a objetos es un paradigma de programación en el que los conceptos del mundo real relevantes para nuestro problema se trasladan a clases y objetos

en nuestro programa. La ejecución del programa consiste en una serie de interacciones entre los objetos. Python también permite la programación imperativa, programación funcional y programación orientada a aspectos.

Python es un lenguaje que todo el mundo debería conocer. Su sintaxis simple, clara y sencilla; el tipado dinámico, el gestor de memoria, la gran cantidad de librerías disponibles y la potencia del lenguaje, entre otros, hacen que desarrollar una aplicación en Python sea sencillo

Raspberry Pi

Según (Aguilar, 2014) la Raspberry Pi es una computadora en una sola tarjeta (Single-Board Computer) creada por la Raspberry Pi Foundation para promover la enseñanza de la programación en escuelas y países en desarrollo.

Biblioteca para acceso a GPIO

```
>>> import RPi.GPIO as GPIO #importa la librería de GPIO
```

```
#usar número de terminal no de GPIO
```

```
>>>GPIO.setmode(GPIO.BOARD)
```

```
#configura como salida en bajo a la terminal 11
```

```
nf>>>GPIO.setup(11,GPIO.OUT,GPIO.PUD_OFF,GPIO.LOW)
```

```
>>>GPIO.output(11,GPIO.HIGH) #pone la salida en alto
```

Estructuras de control básicas

Las estructuras de control más comunes son if, if - else y while, que son controladas por una condición lógica, separada del bloque controlado por : El

inicio y el fin del bloque controlado depende únicamente de la indentación

Tips básicos de Python

- Importante: Python es sensible a mayúsculas y minúsculas
- Numérico: int (1 45 -678), float (12.234 -43.56), complex (-1.23+34.9j 56.1-156j)
- Cadenas de caracteres: “Hola”, ‘Mundo’
- // división de punto flotante: 3/2=1.5
- // división entera: 3//2=1
- type(variable) regresa el tipo de variable

Funciones

Una función de un bloque de código organizado, probado y reutilizable. Permiten que los programas sean modulares.

Python incluye muchas funciones internas, tal como printf, pero también se puede construir funciones propias personalizadas, llamadas funciones definidas por el usuario

La definición de una función comienza por la palabra clave def seguida por el nombre de la función y paréntesis. Los parámetros de entrada se colocan entre los paréntesis.

BIBLIOGRAFÍA

Aguilar, M. F. (2014). Programación en Python en la. Obtenido de <http://sagitario.itmorelia.edu.mx/mfraga/materias/soemb/python.pdf>

- Coen-Porisini, A. R. (2018). Smart transport and logistics: A Node-RED implementation. *Check for updates Wiley*.
- Duque, R. G. (2009). Python para todos. España. Obtenido de http://www.utic.edu.py/citil/images/Manuales/Python_para_todos.pdf
- Lekić, M., & Gardašević, G. (2018). IoT sensor integration to Node-RED platform. *XVII Simposio Internacional INFOTEH*, 5.
- OLIPHANT, T. E. (15 de 05 de 2007). Python for Scientific Computing. CiSE Computational Physics.
- Rossum, G. (1995). Manual de referencia de Python. *CWI (Centro de Matemáticas e Informática)*.
- Chazallet, S. (2016). *Python 3: los fundamentos del lenguaje*. Ediciones ENI.