Manual para Desarrollador
Librería ArduinoRaspberryPiCOM
(v0.6)

1. Enfoque del presente manual

El enfoque de este Manual es para los desarrolladores que deseen realizar modificaciones o nuevas implementaciones en el código fuente de la Librería ArduinoRaspberryPiCOM. El propósito de este es facilitar la comprensión del código fuente de la librería y dar una breve explicación sobre las estructuras y métodos implementados en él. Para obtener el código diríjase al repositorio de GitHub en el siguiente enlace:

https://github.com/smartinezc/ArduinoRaspberryPiCOM

En la carpeta "src" encontrará los ficheros .cpp y .h necesarios para la compilación de la librería en el IDE de Arduino, si desea realizar nuevas implementaciones o modificaciones a la librería debe modificar estos archivos.

2. Uso de la librería ArduinoRaspberryPiCOM

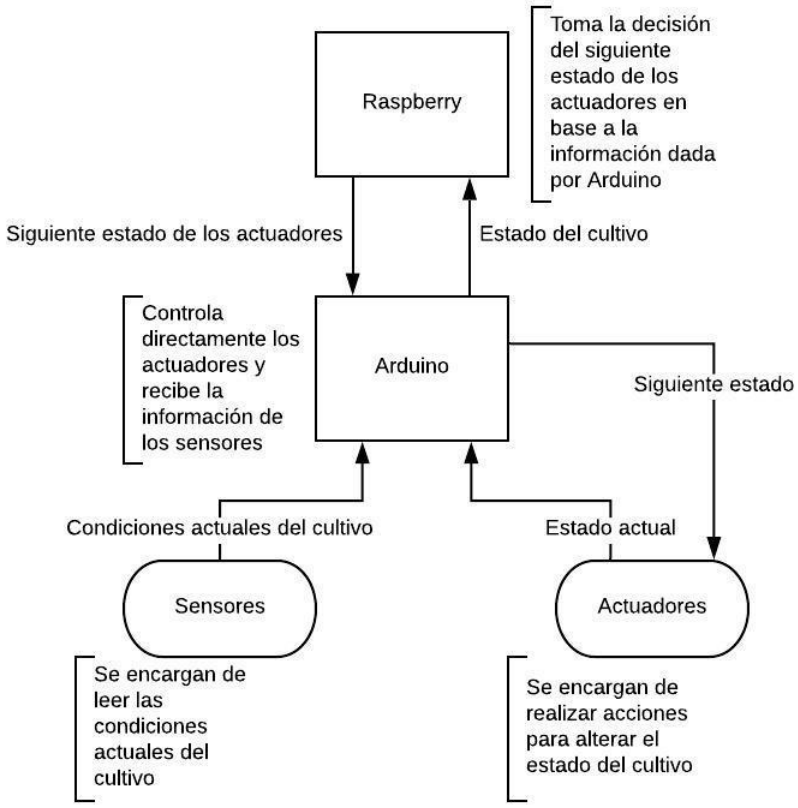
Para familiarizarse con el uso de la librería se recomienda revisar el código de ejemplo, el cual se explica en la sección 2.1 del Manual de Usuario. En la sección 2.2 del Manual de Usuario se da una definición de los Constructores y Métodos disponibles en la librería a manera de API

3. Motivación y diseño

El proyecto de comunicación entre Python y Arduino nace de la necesidad de desarrollar un experimento de riego y monitorización de cultivos para agricultura de precisión usando un Farmbot como herramienta para el cuidado de las plantas. La razón por la que es necesario desarrollar un protocolo de comunicación entre estos dos lenguajes es para facilitar el control del Farmbot, adaptándolo a nuestras necesidades, sin tener que usar la arquitectura que viene de fábrica ya que complica la libertad de uso para el experimento.

Para la realización del experimento es necesario establecer una comunicación entre una serie de sensores que monitorearán el cultivo y el Farmbot, a su vez el Farmbot necesita de una comunicación entre sus actuadores (Arduino) y la tarjeta Raspberry que lo controla (Python). Actualmente la jerarquía de control que usa el Farmbot es la de sus desarrolladores y se basa en la comunicación en Red con un servidor que evalúa el estado actual del Farmbot y toma decisiones en base a éste.

Bajo la jerarquía de control de los desarrolladores del Farmbot se depende del servidor de los fabricantes para el funcionamiento del Farmbot, la placa de Raspberry, que podría procesar la información de los sensores y actuadores y tomar la decisión del siguiente estado del Farmbot, se encarga de establecer una conexión por internet con el servidor. Para mayor libertad y poder hacer uso del CNC de agricultura en diferentes experimentos se propone la siguiente jerarquía:



Para el desarrollo de la comunicación fue necesario plantear los requisitos que se deseaban en el sistema:

- **Escalable:** El programa debe permitir incluir un grupo de sensores de base y adicionar los que sean necesarios sin modificar de forma profunda la lógica de programación, lo mismo con los actuadores
- **Replicable:** El programa debe poder ser entendido, modificado y aplicado a diferentes experimentos.
- **De fácil operación:** El programa debe poder ser utilizado por cualquier persona con un nivel básico de programación.

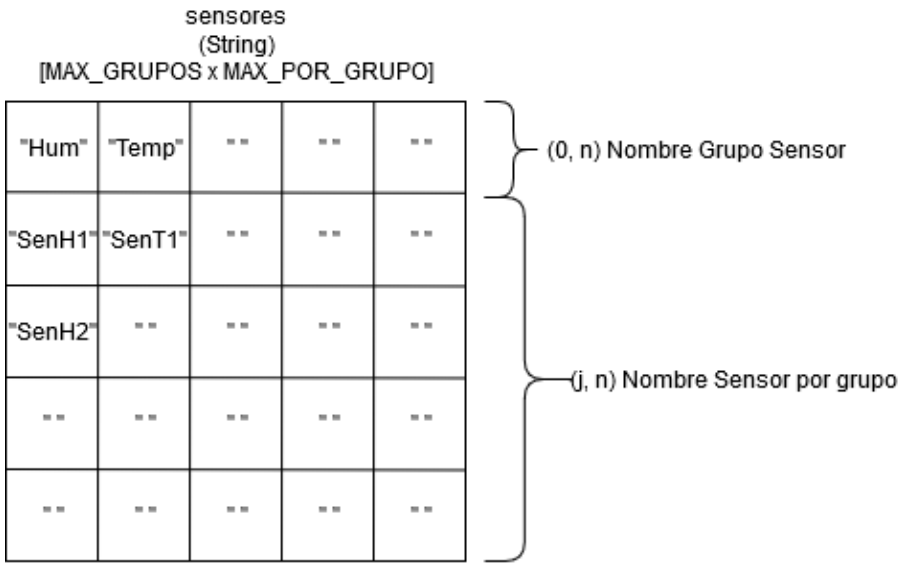
Para cumplir estos objetivos se decidió crear una librería de Arduino que facilitara la comunicación entre las dos plataformas, para dicha comunicación se utilizó el formato JSON para el intercambio de datos ya que es un formato estandarizado, de fácil implementación y escalable a cualquier número de variables (Sensores y actuadores en nuestro caso)

4. Estructuras y principio de funcionamiento

La Librería utiliza 4 matrices para su funcionamiento, el tamaño de estas matrices determina la cantidad de grupos de Sensores/Actuadores que se pueden crear y la cantidad de Sensores/Actuadores por grupo, este tamaño está definido por las constantes MAX_GRUPOS y MAX_POR_GRUPO definidas en “ArduinoRaspberryPiCOM.h”.

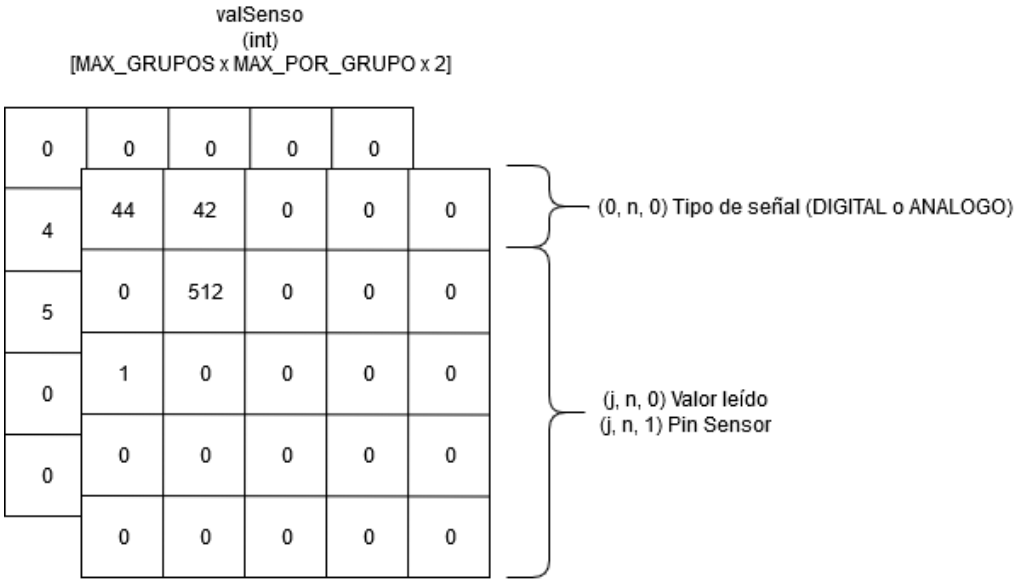
Matriz “sensores”

Esta matriz contiene los nombres de grupo y el nombre de los sensores de cada grupo. Se inicializa con un String vacío, que es usado para saber qué posición está ocupada



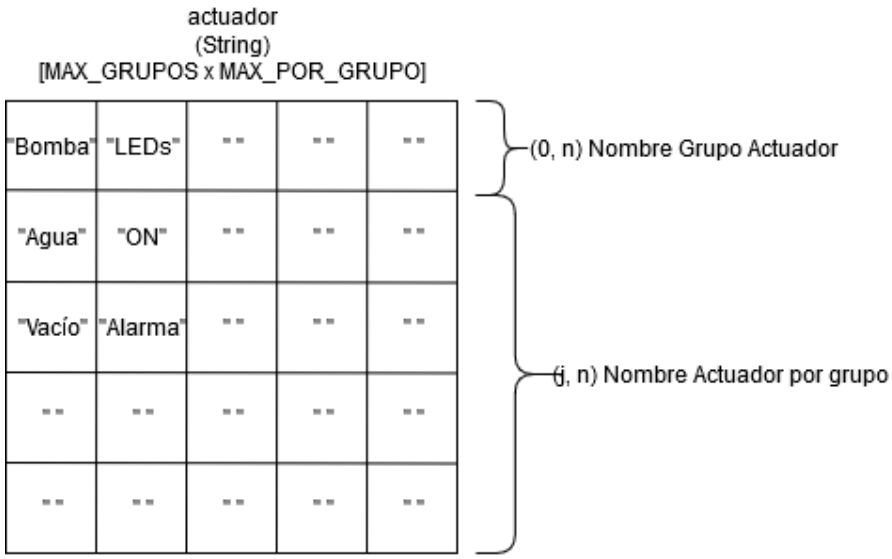
Matriz “valSenso”

Esta matriz contiene la información sobre el tipo de señal de cada grupo de sensores, el último valor leído de los sensores y el pin al que están conectados



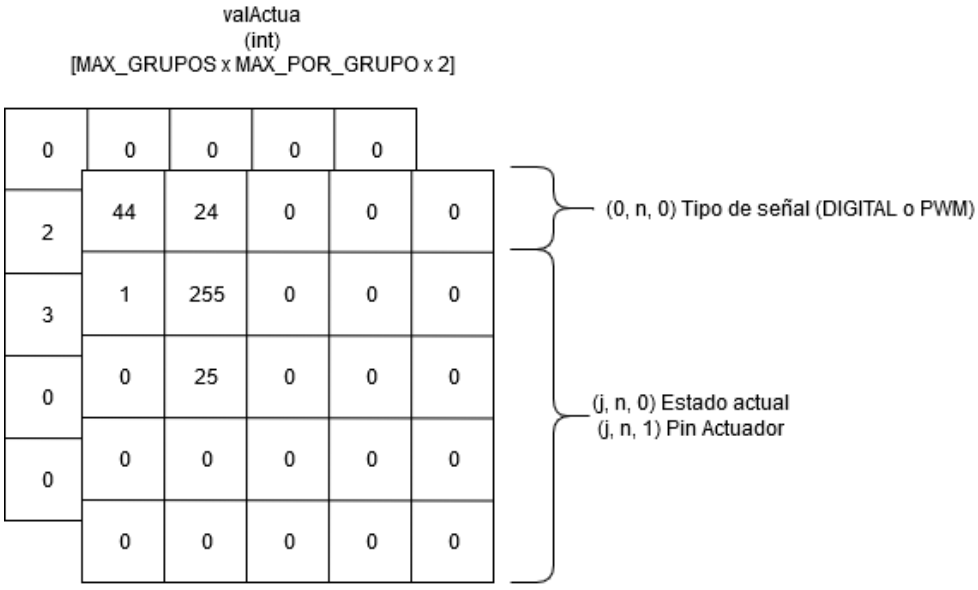
Matriz “actuador”

Esta matriz contiene los nombres de grupo y el nombre de los actuadores de cada grupo. Se inicializa con un String vacío, que es usado para saber qué posición está ocupada



Matriz “valActua”

Esta matriz contiene la información sobre el tipo de señal de cada grupo de actuadores, el estado actual de cada actuador y el pin al que están conectados



Note que j representa la posición de la fila y n la posición de la columna