Conexión a la raspberry pi:

La ip es dinámica por lo que para encontrar su ip se debe de conectar al modem y buscarla (el nombre del dispositivo conectado al modem es: raspberry)

Una vez encontrada la ip, se conecta a la raspberry con putty, solo poniendo la ip y dejando el puerto en 22

Usuario: pi

Contraseña: raspberry

Los archivos de Python se encuentran en la carpeta archPy

Para saber el nombre de los archivos se usa el comando: ls

Lo que hace el comando ls es enlistar todos los archivos (incluyendo carpetas) que se encuentren en la ubicación donde se ejecute el comando

Para ejecutar el archivo se usa el comando python3 nombreDelArchivo.py

Subir archivos de la computadora a la raspberry pi: para subir los archivos se ejecuta el siguiente comando desde la terminal de Windows:

scp direccionDelArchivo\nombreDelArchivo.terminacion pi@ipDeLaRaspberry:DireccionDondeSeGuardaraElArchivo

ejemplo:

scp F:\kito089\esc\Robotica\major\RapArd\connectArd.py [pi@192.168.0.104:~/archPy](mailto:pi@192.168.0.104:~/archPy)

despues de ejecutar el comando, nos pedirán la contraseña de la raspberry

para saber el puerto al que esta conectado el Arduino se ejecutan los siguientes comandos:

ls -1 /dev > dev1.txt (este comando se ejecuta sin el Arduino conectado)

ls -1 /dev > dev2.txt (este comando se ejecuta con el Arduino conectado)

diff dev1.txt dev2.txt

estos comandos lo que harán será: crear un archivo de texto de los puertos conectados, para después compararlos y saber en cual esta conectado el Arduino por la diferencia entre los dos archivos, después de saber el puerto, los archivos creados se pueden eliminar con el comando rm

para compartir datos entra la raspberry y el Arduino se usa la librería serial (pyserial)

los comandos importantes para la raspberry pi son:

 arduino = serial.Serial("/dev/ttyUSB0", 9600, timeout=1)

esta linea crea un objeto de serial que se conecta al puerto en el que esta conectado el Arduino

arduino.isOpen()

devuelve true si el Arduino esta conectado (se usa como método de seguridad)

arduino.write(cmd.encode())

le envia un mensaje al Arduino

 answer=str(arduino.readline())

recibe un mensaje del Arduino

 arduino.flushInput()

remueve los datos de la entrada serial del Arduino, (es como si limpiaramos el cache)

arduino.inWaiting()

devuelve el tiempo de espera en el que se encuentra el Arduino

KeyboardInterrupt

Error que puede saltar el perderse la conexión entre el Arduino y la raspberry pi

Comandos importantes en Arduino:

  msg = "";

  if (Serial.available()) {

      delay(10);

      while (Serial.available() > 0) {

          msg += (char)Serial.read();

      }

      Serial.flush();

  }

Serial available Permite leer los datos enviados al serial (recibir mensajes), serial flush limpia el cache del serial

Serial.print

Permite enviar archivos al serial para que sean leidos por la raspberry pi