**Preparación de Raspberry Pi 4 con OS BOOKWORM para configurar WIFI mediante bluetooth**

El siguiente documento presenta los pasos a considerar para poder conectar el WIFI de una raspberry pi 4 mediante bluetooth. Para ello, se deben instalar los siguientes paquetes:

**sudo apt-get install bluez && sudo apt-get install bluez-tools**

**sudo apt install bluetooth bluez python3-bluez**

En el Raspberry Pi OS Bookworm, el gestor de redes por defecto es NetworkManager. Para verificar, ejecutar:

**nmcli –version**

En caso de no estar instalado, se debe instalar con la siguiente instrucción:

**sudo apt install network-manager**

Ahora se deben hacer ajustes en 2 archivos. El primero se debe ingresar a el mediante el siguiente comando:

**sudo nano /etc/systemd/system/dbus-org.bluez.service**s

Si el archivo aparece vacío, al ejecutar el comando, reemplazar **dbus-org.bluez.service**s por **dbus.org.bluez.service**s.

En el archivo, se debe modificar la linea de **ExecStart** agregando -C al final de la linea y agregar **ExectStartPost** de la siguiente manera:

**..**

**ExecStart=/usr/libexec/bluetooth/bluetoothd -C**

**ExecStartPost = /usr/bin/sdptool add SP**

**…**

El segundo archivo a modificar se accede mediante el comando:

**sudo nano /etc/bluetooth/main.conf**

En el, se deben descomentar y/o agregar las siguientes lineas:

**DiscoverableTimeout = 0 # 0 significa que es siempre visible**

**PairableTimeout = 0 # 0 significa que siempre es emparejable**

**Discoverable = true**

**Pairable = true**

Luego, se deben reiniciar los servicios de bluetooth mediante el comando:

**sudo systemctl daemon-reload**

**sudo systemctl restart bluetooth**

Ahora, se debe configurar el bluetooth ingresando con el comando:

**sudo bluetoothctl**

En el, se debe escribir:

**[bluetooth] power on**

**[bluetooth] discoverable on**

**[bluetooth] pairable on  
[bluetooth] agent NoInputNoOutput**

**[bluetooth] default-agent**

**[bluetooth] system-alias «Nombre para bluetooth» ← Opcional**

**[bluetooth] exit**

Finalmente, se deben dar permisos de súper usuario al bluetooh con el comando:

**sudo usermod -a -G bluetooth $(whoami)**

Ya con todo configurado, se recomienda volver a reiniciar el servicio de bluetooth:

**sudo systemctl restart bluetooth**

Para conectar un dispositivo a la raspberry pi sin necesidad de que pida código de validación en la raspberry, se debe ejecutar primero:

**sudo bt-agent -c NoInputNoOutput**

Para poder configurar el WIFI de la raspberry pi mediante el dispositivo vinculado mediante bluetooth, se debe ejecutar el archivo connectWIFIBT.py como súper usuario. Para ello, el comando asociado es:

**sudo python3 connectWIFIBT.py**

Para hacer la configuración de WIFI, se debe usar alguna aplicación de comunicación bluetooth serial, como lo es **Serial Bluetooth Terminal** para Android.

En la aplicación, se debe seleccionar el dispositivo vinculado y luego, enviar la instrucción:

**WIFI:«SSID»;«PASSWORD»**

Donde el SSID es el nombre de la red a la que se necesita conectar y el PASSWORD es la contraseña de la misma, separados por punto y coma. Esto hará que la raspberry busque y se conecte a la red, informando en el terminal de serial bluetooth terminal que la conexión fue exitosa y luego se desconectará del terminal. Si no encuentra la red o si se ingresaron mal los parámetros de la conexión, saldrá un error y se deben ingresar las credenciales correctas, haciendo que la raspberry vuelva a intentar conectarse a dicha red.

**OPCIONAL**

Para que el programa se ejecute al iniciar la raspberry, se debe generar un archivo de arranque y agregarlo a la rutina de crontab. Para ello, abriendo un terminal, se debe ejecutar:

**nano arranque.sh**

En el archivo que se genera se debe escribir:

**export DISPLAY=:0**

**while true; do**

**xterm -e "sudo python3 /home/pi/bluethoot/conect\_wifi\_blue.py; exit"**

**sleep 1**

**done &**

**while true; do**

**sudo bt-agent -c NoInputNoOutput &**

**BTAG\_PID=$!**

**wait $BTAG\_PID**

**sleep 1**

**Done**

**NOTA:** En el archivo se esta usando xterm para ejecutar los programas. Se recomienda instalar xterm en caso de que no se encuentre instalado**(sudo apt install xterm).** Ademas, se debe colocar la ruta del archivo .py según sea la ubicación de este.

Luego se debe ejecutar crontab como :

**crontab -e**

Y en el agregar la siguiente instrucción:

**@reboot /home/pi/arranque.sh**

**NOTA:** Se debe seleccionar la ruta donde se haya generado el de arranque.sh

Guardar y hora reiniciar la raspberry:

**Sudo reboot**

Ahora la raspberry cada vez que se encienda, iniciará el servicio de configuración WIFI mediante bluetooth.