

Instructivo para configurar una Raspberry Pi 4b para ser usada con el software Device Control v 1.5.x

Marcos D'Andrea

Este manual está ajustado a la versión del SO "Raspberry Pi OS Lite 64bit" versión 2024-07-4

Instalación y configuración de Sistema Operativo	1
Configuración de SSH	2
Instalación de entorno gráfico	3
Instalación de la aplicación "Device Control"	4
Montar el pendrive en el sistema	4
Configurar "Device Control" para iniciar junto con el entorno gráfico.....	4
Instalar el archivo de configuración de Device Control	5
Deshabilitar el requerimiento del SO para iniciar sesión de usuario	6
Configurar "startx" como un servicio del SO para que se inicie automáticamente.....	6
Actualizar la aplicación	7
Blanquear host SSH conocido en windows	7
Cambiar la dirección IP de la Raspberry.....	7
Conectar la raspberry a la red WiFi.....	8
Cerrar la aplicación.....	9
Desmontar un pendrive	9

Instalación y configuración de Sistema Operativo

1. Descargar "Pi Imager" de la página oficial de Raspberry Pi, instalarlo y ejecutarlo.
 - a. Luego de seleccionar el dispositivo ("Raspberry Pi 4"), el almacenamiento (La memoria micro SD) y el Sistema Operativo ("Raspberry Pi OS Lite 64 bits" que se encuentra dentro de la opción "otros SO") la instalación haciendo uso de la opción "Editar Ajustes". Debe estar insertada la memoria SD para que aparezcan las opciones.
 - b. Se abrirá un panel para configurar los parámetros de la instalación.
 - i. Establezca como nombre de anfitrión el nombre que le dará al dispositivo, por ejemplo "devicecontrol" (sin comillas).
 - ii. Establezca el nombre de usuario y la contraseña. Es común usar "proyecciones" y "pess1102".
 - iii. Configure la Red Inalámbrica y los ajustes regionales si fuera necesario.
 - iv. En la pestaña "Servicios", active el SSH con la opción "Usar autenticación por contraseña".

- v. En la pestaña “Opciones” asegúrese que “Activar Telemetría” esté deshabilitado.
- c. Guarde los ajustes y proceda con la preparación de la memoria SD.
- d. Finalizada la preparación, colocar la memoria en la Raspberry y encenderla con el teclado conectado.
- e. La Raspberry iniciarla, mostrará una consola ejecutando. Esperar a que finalice.
- f. El cursor quedará a la espera del login de usuario

```
devicecontrol login: _
```

- g. Escriba el usuario que haya configurado, en este ejemplo “proyecciones” (sin comillas)
- h. Inmediatamente le solicitará su contraseña, en este ejemplo “pesp1102” (sin comillas)

```
Password: _
```

- i. Escriba la contraseña. No verá los caracteres mientras los escribe. Tiene 60 segundos para hacerlo de lo contrario volverá al inciso b.
- j. Una vez logueado el sistema quedará listo para recibir comandos

```
proyecciones@devicecontrol: ~$ _
```

Configuración de SSH

Para simplificar el procedimiento de ahora en mas, nos conectaremos mediante el servicio de consola segura SSH. Para eso, es necesario conocer la dirección IP de la Raspberry. Si lo desea puede omitir este paso y continuar desde el punto 4.

- a. Escriba en la consola el siguiente comando (respetando mayúsculas):

```
hostname -I
```

- b. En windows, ejecute PowerShell y escriba el siguiente comando, reemplazando “proyecciones” por el nombre de usuario que haya creado (En este ejemplo fue “proyecciones” y reemplace la ip, por la dirección pública informada en el paso anterior.

```
ssh proyecciones@192.168.100.109
```

- c. Probablemente, PowerShell le informe que la autenticidad del host no puede ser determinada y le pregunte lo siguiente:

```
Are you sure you want to continue connecting (yes/no/[Fingerprint])?
```

- d. Escriba “yes” y presione enter. Este paso guardará el Host en su lista de host conocidos. Si por algun motivo, a partir de ahora, usted reinstala el sistema operativo de su raspberry, se producirá un conflicto que PowerShell le indicará,

esto es porque el fingerprint del host (la raspberry) habrá cambiado, y deberá [seguir otros pasos](#) para blanquear el fingerprint de dicho host antes de continuar. Repita desde el punto B y esta vez le saldrá lo siguiente:

```
proyecciones@192.168.100.109's password:
```

- e. Escriba la contraseña de su raspberry, en este ejemplo ("pesp1102"). No verá el cursor mientras escribe la contraseña.
- f. Si todo salió correctamente debería ver la consola a la espera de comandos:

```
proyecciones@devicecontrol: ~ $ _
```

- g. Desde aquí podrá operar su Raspberry como si estuviera operándola de forma presencial con un teclado conectado.

Instalación de entorno gráfico

- a. Necesita instalar un entorno gráfico de usuario para usar aplicación que poseen interfaces graficas de usuario, como es el caso de "Device Control". Para eso usaremos xserver.
- b. Ejecute el siguiente comando en la consola:

```
sudo apt update
```

- c. Esto actualizará la lista de paquetes del repositorio del sistema operativo. Espere que se complete el proceso. Para esto debe estar conectado a internet.
- d. Luego es necesario actualizar los paquetes del SO a una versión mas reciente. Ejecute:

```
sudo apt upgrade
```

- e. Se le preguntará en un momento si desea continuar, escriba "y" y presione enter.
- f. Finalmente procederemos a instalar el servicio de entorno gráfico xserver. Para eso, ejecutaremos el siguiente comando:

```
sudo apt install xserver-xorg-input-libinput xinput-calibrator
```

- g. En un momento se le preguntará si desea continuar. Escriba "y" y presione enter. Se descargarán e instalarán los paquetes de entorno gráfico.
- h. Luego instale el servicio de inicializador del entorno:

```
sudo apt-get install xinit
```

- i. En un momento se le preguntará si desea continuar. Escriba "y" y presione enter.
- j. Desde un teclado conectado a la raspberry, ejecute en la consola (estando logueado en sistema operativo con su usuario y contraseña) el siguiente comando:

```
startx
```

- k. Deberá abrirse un entorno gráfico básico, y verá una pantalla negra, con otra ventana donde debe haber una consola. Para regresar a la vista de consola y

cerrar el entorno gráfico, presione CTRL+ALT+F3 (si está usando un teclado con las teclas de función supeditadas a una tecla FN presionela)

Instalación de la aplicación “Device Control”

Montar el pendrive en el sistema

- a. Coloque el pendrive en la raspberry y proceda a montar la unidad. Para ello, es necesario conocer la ubicación del pendrive. Escriba en la consola:

```
lsblk
```

- b. Esto desplegará un árbol con las interfaces conectadas, el pendrive debería estar bajo “sda”, pudiendo ser “sda1”, “sda2”, etc
- c. Proceda a montar la unidad en el directorio “/mnt”

```
sudo mount /dev/sda1 /mnt
```

- d. El pendrive habrá quedado montado en el directorio /mnt. Diríjase a ese directorio:

```
cd /mnt
```

- e. Liste los archivos dentro del directorio, debería ver los archivos de su pendrive:

```
ls
```

- f. Proceda a la instalación de “Device Control”, reemplazando “devicecontrol_1.2.2_arm64.deb” por el nombre que posea su archivo de instalación:

```
sudo dpkg -i ./devicecontrol_1.5.2_arm64.deb
```

- g. Si durante la instalación se presentaran problemas con algunas dependencias, ejecute lo siguiente y vuelva a repetir el paso “f”:

```
sudo apt-get install -f
```

- h. Compruebe que se instaló correctamente escribiendo:

```
which devicecontrol
```

- i. Debería ver el directorio donde fue instalada la aplicación, quizás “/usr/bin/devicecontrol”

Configurar “Device Control” para iniciar junto con el entorno gráfico

Por ahora, Device Control, solo puede iniciarse de modo manual, si queremos que se inicie como la única aplicación del entorno gráfico que instalamos (xserver) debemos crear un archivo de configuración para el mismo con esa indicación.

- a. Cree un archivo llamado “.xinitrc”

```
nano ~/.xinitrc
```

- b. Se abrirá el editor de texto “nano” con el archivo creado en blanco. Escriba lo siguiente en el archivo, colocando la ruta de la aplicación que fue dada en el paso “i”:

```
#!/bin/sh  
exec /usr/bin/devicecontrol
```

- c. Guarde el archivo presionando CTRL+O, luego “Enter” y luego CTRL+X para salir.
- d. El siguiente paso es transformar el archivo “./xinitrc” en un script ejecutable, para eso ejecute lo siguiente:

```
chmod +x ~/.xinitrc
```

- e. Para comprobar que todo haya funcionado correctamente, ejecute:

```
startx
```

- f. Debería abrirse nuevamente el entorno gráfico, esta vez, debería ver un cartel que dice “Archivo de configuración no encontrado”. Este mensaje proviene de “Device control” y le está informado que el archivo “config.json” no está copiado al sistema. Anote la dirección donde el cartel le indica que debe instalarse dicho archivo de configuración, quizás sea algo como “/opt/Device Control/config.json”.
- g. Salga del entorno gráfico presionando CTRL+ALT+F4.

Instalar el archivo de configuración de Device Control

Para que “Device Control” pueda mostrar todas las rutinas es necesario copiarlo en la Raspberry en un directorio donde la aplicación pueda encontrarlo. Una vez tenga su archivo de configuración completo y probado en su computadora, copielo en el pendrive y proceda a montarlo siguiendo los pasos que se describen en el [punto 5](#).

- a. Una vez montado el pendrive, diríjase al directorio donde ha sido montado, por ejemplo “/mnt”

```
cd /mnt
```

- b. Una vez ahí, copie el archivo de configuración al directorio donde la aplicación le indicó. Deberá usar comillas (”) en la dirección de destino ya que el directorio “Device Control” posee un espacio entre sus caracteres.

```
sudo cp /mnt/config.json "/opt/Device Control/config.json"
```

- c. Vuelva a ejecutar la interfaz gráfica

```
startx
```

- d. Device Control debería iniciar y mostrar el panel principal con las rutinas que usted haya configurado.
- e. Puede salir de la aplicación presionando CTRL+ALT+F2/F1

Deshabilitar el requerimiento del SO para iniciar sesión de usuario

- a. para ello, escriba lo siguiente:

```
sudo raspi-config
```

- b. Esto abrirá el panel de configuración de la raspberry. Diríjase a la opción "System Options" y luego a la opción "Boot / Auto Login". Seleccione allí la opción "Console Autologin". Luego salga del panel de configuración usando la opción "<Finish>" y proceda a reiniciar su raspberry.

Configurar "startx" como un servicio del SO para que se inicie automáticamente

- a. Cree un archivo systemd para "startx":

```
sudo nano /etc/systemd/system/startx.service
```

- a. Se abrirá el editor "nano". Copie el siguiente contenido en el archivo, donde **User=proyecciones** deberá ser el nombre de usuario que ha configurado en el Sistema Operativo. Se agrega el parametro "-- --nocursor" para no mostrar el cursor en la pantalla.

```
[Unit]
Description=Iniciar entorno gráfico y aplicación
After=network.target

[Service]
User=proyecciones
Environment=DISPLAY=:0
ExecStart=/usr/bin/startx -- --nocursor
Restart=always

[Install]
WantedBy=multi-user.target
```

- b. Recargue systemd para que reconozca el nuevo servicio.

```
sudo systemctl daemon-reload
```

- c. Luego habilite el servicio

```
sudo systemctl enable startx.service
```

- d. Finalmente, debe configurar startx para permitir la ejecución desde cualquier usuario del sistema, para ello, ejecute:

```
sudo nano /etc/X11/Xwrapper.config
```

- e. Esto abrirá "nano" con el archivo de configuración de entorno de ejecución de "Xserver". Modifique la línea correspondiente para que diga

```
allowed_users=anybody
```

- f. Guarde con CTRL+O y salga con CTRL+X
- g. Reinicie la Raspberry

```
sudo reboot
```

- h. Si todo funcionó correctamente, la Raspberry debería reiniciar, iniciar y mostrar el panel de rutinas de "Device Control"

Actualizar la aplicación

Si lo requiere, puede actualizar la aplicación Device Control volviendo a realizar el [punto 5](#). Lo mismo si debiera actualizar el archivo de configuración, realizando los pasos del [punto 7](#).

Blanquear host SSH conocido en windows

Para blanquear el host del ssh en caso de reinstalar la raspberry una vez registrado el host conocido en windows, escriba en PowerShell, reemplazando por la ip de su raspberry:

```
ssh-keygen -R <ip_raspberry>
```

Esto blanqueará el fingerprint del host conocido de su raspberry y le permitirá volver a registrar uno nuevo con la misma IP.

Cambiar la dirección IP de la Raspberry

Es probable que necesite cambiar la dirección IP de la raspberry para poder recibir los comandos de respuesta desde los distintos dispositivos a controlar, para ello deberemos seguir los siguiente pasos:

1. Identificar la interfaz de red. Para ello escriba:

```
ip a
```

2. Esto listará todas las interfaces de red, como `eth0` para una conexión Ethernet o `wlan0` para una conexión Wi-Fi.
3. Editar el archivo `dhcpcd.conf`.

```
sudo nano /etc/dhcpcd.conf
```

4. Desplázate hasta el final del archivo y agrega la siguiente configuración, reemplazando las direcciones IP con las que correspondan a tu red:

```
interface eth0
static ip_address=192.168.1.100/24
static routers=192.168.1.1
static domain_name_servers=192.168.1.1 8.8.8.8
```

5. Si estás usando Wi-Fi en lugar de Ethernet, reemplaza `eth0` con `wlan0`.
6. `static ip_address`: La dirección IP que desees asignar a la Raspberry Pi. Asegúrate de que esté fuera del rango de direcciones asignadas por el servidor DHCP del router.

7. static routers: La puerta de enlace de la red, usualmente la IP del router (ej. 192.168.1.1).
8. static domain_name_servers: Los servidores DNS. Puedes usar los de tu router o servidores públicos como los de Google (8.8.8.8 y 8.8.4.4).
9. Guarda los cambios presionando Ctrl + O, luego presiona Enter. Para salir de nano, presiona Ctrl + X.
10. Reinicia la raspberry. Al iniciar Device Control deberías ver en la barra inferior la nueva IP asignada.

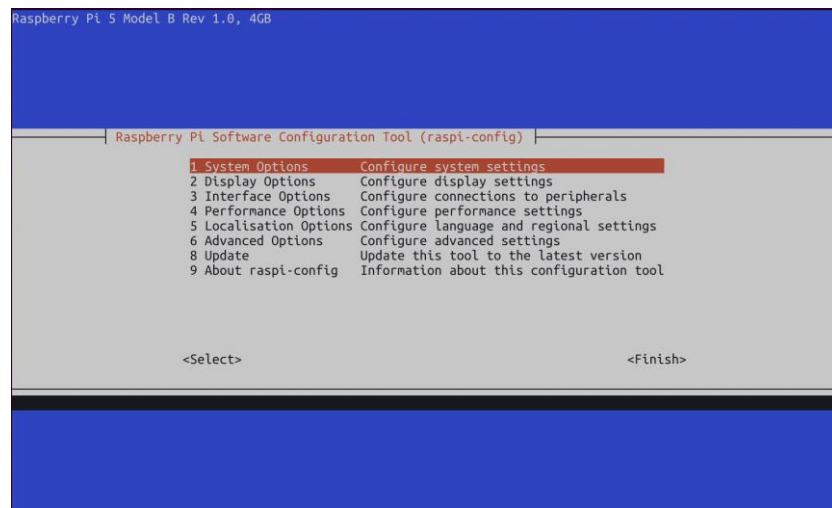
```
sudo reboot
```

Conectar la raspberry a la red WiFi

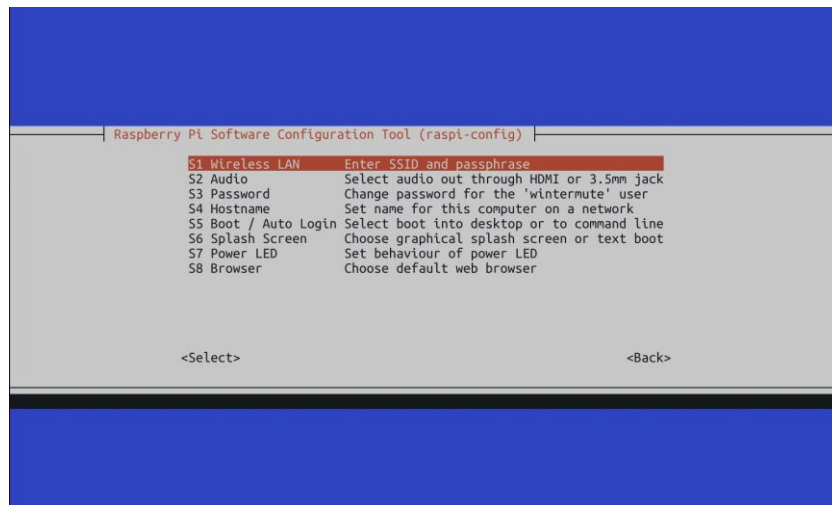
1. Inicie el panel de configuración de Raspberry

```
sudo raspi-config
```

2. Inicie el panel de configuración de Raspberry



3. Ingrese en System Options



- Desde la opción Wireless LAN ingrese primero la SSID y luego la contraseña. Al finalizar ejecute

```
sudo reboot
```

Cerrar la aplicación

Siempre que este configurado puede configurar la raspberry mediante SSH sin tener que cerrar la aplicación, pero si lo debiera hacer, puede presionar, con el teclado conectado, CTRL+ALT+F2 / F1

Desmontar un pendrive

Ejecute el siguiente comando, indicando el directorio donde haya montado el pendrive, por ejemplo, si lo montó en “/mnt”. Procure salir del directorio “/mnt” antes de desmontar.

```
sudo umount /mnt
```

Capítulo 2: Configuración de Rutinas en Device Control

Device Control le permite generar **rutinas** para ejecutar una o varias **tareas**. Las tareas estan compuestas por un **trabajo** y opcionalmente una **condición**; en caso de contener una condición, es necesario indicarles la cantidad de reintentos y el intervalo de espera entre reintentos. La rutina se ejecuta mediante la configuración de uno o mas **disparadores**. Cuando cualquiera de estos disparadores se ejecuta, la rutina comienza a ejecutar las tareas. Dependiendo de la configuración de la misma, las tareas se pueden ejecutar de forma sincrónica (una detras de otra) o de forma asincrónica (todas simultaneamente). Si todas las tareas de una rutina satisfacen su condición (las que posean una condición asociada) la rutina se completará sin errores, de lo contrario informará que falló.

Los trabajos disponibles en la presente versión de Device Control son:

Nombre	Descripción	Parametros
SendTCP	Envía mensajes TCP sin esperar respuesta.	ip <String> port <Number> message <String>
SendUDP	Envía mensajes UDP sin esperar respuesta.	ip <String> port <String> message <String>
Wait	Espera sin realizar acción, esto puede ser configurado mediante una variable llamada “time”. Es util para establecer una separación temporal entre otros trabajos cuando se ejecuta una rutina secuencial.	No requiere
WakeOnLan	Envía un paquete WOL para encender dispositivos compatibles	macAddress <String>
SendDMX	Envía paquetes Artnet por la red	Universe <Number> Net <Number> Opcional Subnet <Number> Opcional Channel <Number> Value <Number> IpToSend <String> Opcional de lo contrario es broadcast
pjLinkCommand	Controla dispositivos compatible con pjLink. Los comandos disponibles son: powerOn powerOff	ip <String> command <String>

Las condiciones disponibles en la presente versión de Device Control son:

Nombre	Descripción	Parametros
Ping	Realiza un comando de ping a un dispositivo de la red compatible. Util para comprobar si un dispositivo está presente en la red, ya sea encendido o conectado.	ip <String>
TCPAnswer	Envía un paquete TCP y espera una respuesta específica	ip <String> port <String> messageToSend <String> messageToExpect <String>
UDPAnswer	Envía un paquete UDP y espera una respuesta específica en el mismo puerto que se originó el mensaje.	ip <String> port <String> messageToSend <String> messageToExpect <String>
pjLinkStatus	Obtiene el estado de dispositivos pjLink compatibles. Los comandos disponibles son: checkPowerOn checkPowerOff	ip <String> command <String>

Los Triggers disponibles en la presente versión de Device Control son:

Nombre	Descripción	Parametros
api	Permite ejecutar una rutina al hacer un llamado al endpoint que se configura mediante sus parámetros. El método puede ser: GET POST DELETE PATCH y el endpoint tiene que tener la forma <code>"/trigger/direccion1"</code> donde <i>direccion1</i> puede ser el nombre que usted quiera, además puede agregar varios subdominios mas.	method <String> endpoint <String>
cron	Permite disparar una rutina por cronograma. El parametro time debe tener la forma <code>"00:00"</code> base 24 horas y el parametro day especifica el numero de día de la semana, siendo 0 el día domingo y 6 el día sabado.	day <Number> time <String>
event	Activa una rutina por eventos disparados de otras rutinas. Las rutinas disparan eventos y estos pueden ser escuchados por un trigger de tipo "event". Los eventos que emiten las rutinas son: armed, completed, failed, triggered, taskStarted, taskCompleted, taskFailed, taskRetry. Un ejemplo: <i>routine01:completed</i> el trigger se ejecutará cuando la rutina con el id <i>"routine01"</i> dispare el evento <i>"completed"</i>	name <String>
udp	Dispara una rutina cuando se recibe un mensaje específico en un puerto específico hacia la IP de Device Manager	port <Number> message <String>

La configuración de todo esto se realiza modificando el archivo **config.json**

Un archivo **config.json**, sin rutinas configuradas (es decir vacío pero aún así válido para el programa) es el siguiente:

```
{
  "routines":[]
}
```

Si usted carga un archivo de configuración con el contenido indicado arriba, verá la pantalla de rutinas vacía. Eso es porque justamente dentro de los corchetes [] debemos comenzar a escribir las configuraciones.

A continuación crearemos una rutina para enviar un WakeOnLan y luego haremos una solicitud de ping para chequear que el dispositivo se haya encendido, la rutina se ejecutará al hacer una llamada a una api.

```
{
  "routines": [
    {
      "id": "powerON",
      "name": "Encendido General",
      "description": "Encender servidor de video.",
      "isSequential": false,
      "stopOnTaskFailure": false,
      "autoEvaluateConditions": 60000,
      "enabled": true,
      "visible": true,
      "trigger": [
        {
          "type": "api",
          "params": {
            "method": "GET",
            "endpoint": "/trigger/powerON"
          }
        }
      ],
      "tasks": [
        {
          "name": "Encender servidor",
          "retries": 15,
          "retryDelay": 20000,
          "job": {
            "type": "wakeOnLan",
            "params": {
              "macAddress": "C8:08:E9:8D:E5:CE"
            }
          },
          "condition": {
            "type": "ping",
            "params": {
              "ip": "192.168.10.10",
            }
          }
        }
      ]
    }
  ]
} // ← ingrese una coma aquí y agregue otra rutina
```

id: <string> Identificador único de esta rutina. No debe repetirse.

name: <string> Nombre del componente, en este caso una rutina, el mismo se mostrará en la interfaz gráfica de usuario (GUI).

descripción: <string> Breve reseña de esta rutina.

isSequential: <boolean> Si las tareas se ejecutan en secuencia, una tras otra, o se ejecutan todas en simultaneo.

autoEvaluateConditions: <false | number> Si la tarea se auto evalúa para comprobar su estado. Si es false no lo hace, si se especifica un numero, equivale en ms al intervalo de chequeos.

enabled: <boolean> Determina el estado inicial de la rutina. Si la rutina esta deshabilitada, no se podrá ejecutar.

visible: <boolean> Si la rutina debe o no mostrarse en la GUI.

trigger: <[objeto]> Una lista de triggers con los que esta rutina puede ser disparada.

tasks: <[objeto]> Una lista de tareas que esta rutina debe ejecutar.

type: <tipo> Especifica el componente. La lista de componentes compatibles por cada versión de Device Control esta especificada en este manual.

params: <objeto> Especifica los parametros del componente, los mismos varían dependiendo del type de componente. Refiera a este manual para conocerlos.

retries: <number> Obligatorio si la tarea incluye una condition. Indica la cantidad de reintentos de la tarea hasta declararse fallida.

retryDelay: <number> Obligatorio si la tarea incluye una condition. Indica el tiempo entre de reintentos de la tarea.

job: <object> El trabajo que debe realizar la tarea. Los type de trabajos estan definidos en este manual.

condition: <object> La condición que comprueba si el trabajo se ha completado. Es opcional. Los type de conditions estan definidos en este manual. En caso de no incluir una condition, la tarea se declara completada luego una ejecución satisfactoria de tu job.