

Robot con cámara web - Raspberry Pi



El principal objetivo de este proyecto es crear sistemas de monitoreo básicos. En el video de demostración de transmisión en vivo podemos observar su funcionamiento, con la implementación del robot meccanoid que puede grabar ciertos movimientos para reproducirlos después como se verá en los videos del proyecto, podemos programar ciertos recorridos para estar monitoreando ciertas actividades que hagamos día a día, como por ejemplo estar monitoreando si tu gato se encuentra en casa.

¿Cómo funciona el robot webcam Raspberry Pi?

La cámara web USB conectada a la raspberry pi generalmente se asocia con la pi en /dev/video0. Un servicio de transmisión llamado mjpg streamer después de la instalación se utiliza para transmitir el video en la dirección IP local y el número de puerto de la raspberry pi. La resolución del video y la velocidad de fotogramas se pueden establecer usando comandos de Linux mientras se ejecuta el transmisor mjpg. La fuente de video se puede ver en cualquier dispositivo conectado a la red local abriendo un archivo HTML en un navegador que invoca el video desde iAddress:portno de raspberry pi.



Para probar la cámara web localmente en el pi, se utiliza un servicio llamado ffmpeg. Esto le permite ver la alimentación de la cámara web en el monitor conectado a la raspberry pi. Un diagrama de bloques que muestra el funcionamiento básico:



Diagrama de bloques para el robot de cámara web Raspberry Pi

Paso 1: Conexión y comprobación de la cámara web USB

Verificar si la cámara web se detecta en la raspberry pi y si se puede ver su alimentación. Ejecutar el comando "lsusb" en el terminal de la raspberry pi. Esto le muestra una lista de todos los dispositivos conectados por USB en la pi. Averigüe en la lista si se muestra el nombre o el software de su cámara web (a veces se muestra el nombre del controlador, por ejemplo: Microdia para cámaras web Iball). A continuación, debemos verificar si puede ver el feed desde la cámara web en la pi. Para ello utilice "cd /dev" para ir al directorio /dev. A continuación, use "ls" para enumerar su contenido, verifique si "video0" está presente.

```

pi@raspberrypi10:~ $ lsusb
Bus 002 Device 001: ID 1deb:0003 Linux Foundation 3.0 root hub
Bus 001 Device 004: ID 04d9:0007 Holtek Semiconductor, Inc. Raspberry Pi Internal Keyboard
Bus 001 Device 005: ID 1b3f:2247 Generalplus Technology Inc. GENERAL WEBCAM
Bus 001 Device 002: ID 2109:3431 VIA Labs, Inc. Hub
Bus 001 Device 001: ID 1deb:0002 Linux Foundation 2.0 root hub

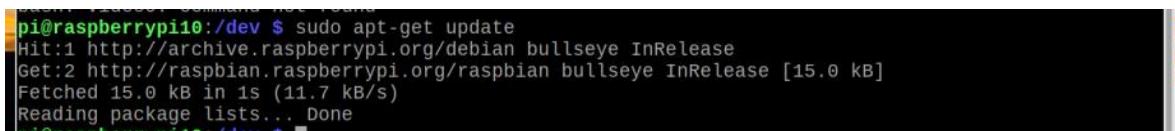
pi@raspberrypi10:~ $ cd /dev
pi@raspberrypi10:~/dev $ ls
autofs      input      ppp       serial1   tty25   tty48   urandom   vcsu3
block       kmsg      ptmx      serial2   tty26   tty49   v4l       vcsu4
btrfs-control log       pts       snd       tty27   tty5    vchiq     vcsu5
bus        loopo      ram0      stderr    tty28   tty50   vcio      vcsu6
cachefiles  loop1      ram1      stdin    tty29   tty51   vc-mem    vcsu7
ceco       loop2      ram10     stdout   tty3    tty52   vcs      vga_arbiter
cec1       loop3      ram11     tty     tty30   tty53   vcs1     vhci
char       loop4      ram12     tty     tty31   tty54   vcs2     video0
console    loop5      ram13     tty1    tty32   tty55   vcs3     video1
cuse       loop6      ram14     tty10   tty33   tty56   vcs4     video10
disk       loop7      ram15     tty11   tty34   tty57   vcs5     video11
dma_heap   loop-control ram2      tty12   tty35   tty58   vcs6     video12
ori        mapper    ram3      tty13   tty36   tty59   vcs7     video13
fd         media0    ram4      tty14   tty37   tty6    vcs8     video14
full      media1    ram5      tty15   tty38   tty60   vcsa1    video15
fuse       media2    ram6      tty16   tty39   tty61   vcsa2    video16
gpiochip0  media3    ram7      tty17   tty4    tty62   vcsa3    video18
gpiochip1  mem       ram8      tty18   tty40   tty63   vcsa4    video20
gpiomem   mmcblk0e   ram9      tty19   tty41   tty7    vcsa5    video21
hidraw0   mmcblk0p1  random   tty2    tty42   tty8    vcsa6    video22
hidraw1   mmcblk0p2  rkill    tty20   tty43   tty9    vcsa7    video23
hw RNG    mqueue   rpivid-h264mem  tty21   tty44   ttyAMA0  vcsm-cma  video31
i2c-20    net      rpivid-hevcmem  tty22   tty45   ttyprintk vcsu    watchdog
i2c-21    null    rpivid-intmem  tty23   tty46   uhid   vcsu1   watchdog

```

Ahora debemos usar un servicio llamado ffmpeg para ver la alimentación de la cámara web en el monitor conectado a la pi. Utilizar "sudo apt-get update" para actualizar las listas. Para instalar el servicio ffmpeg, utilice estos comandos:

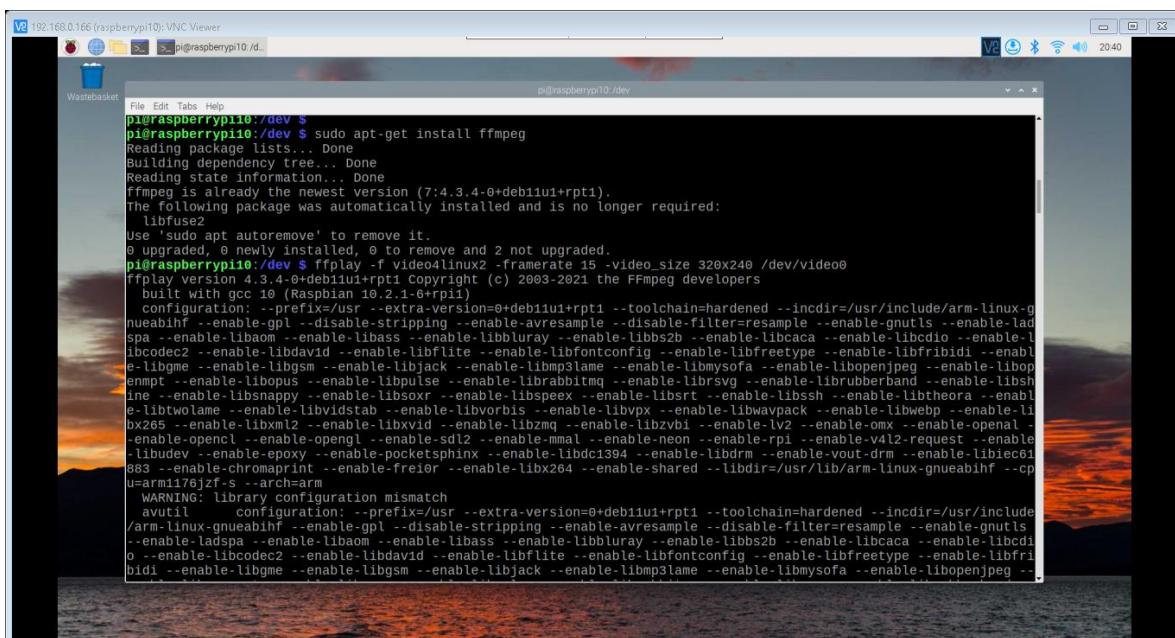
```
sudo apt-get install ffmpeg
```

```
ffplay -f video4linux2 -framerate 15 -video_size 320x240 /dev/video0
```



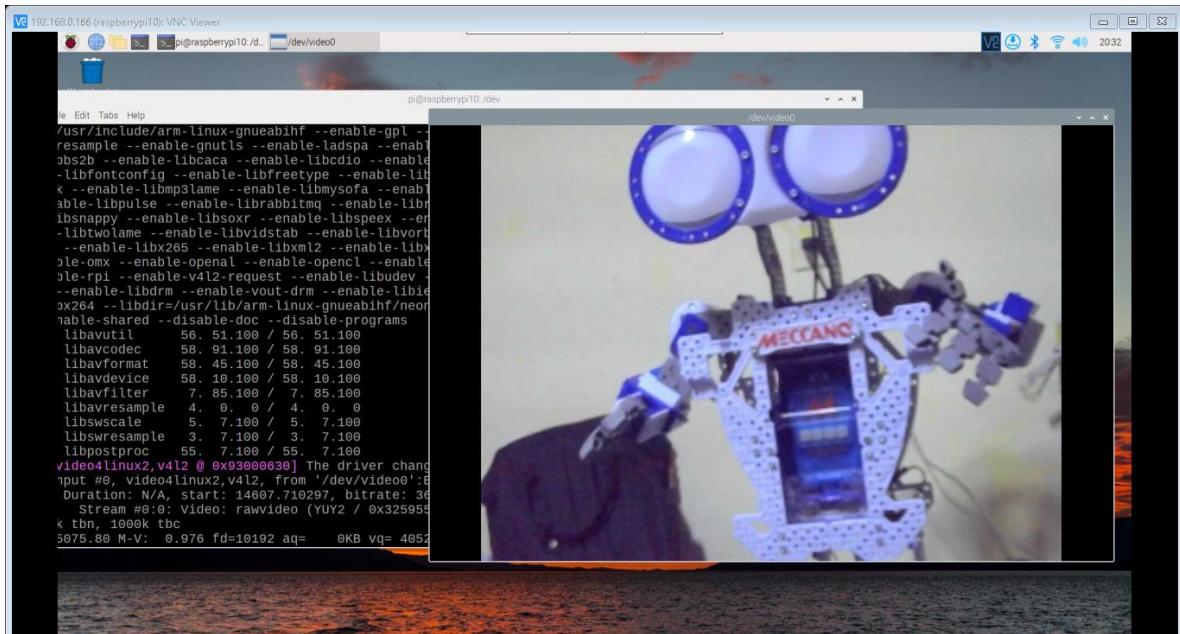
```
pi@raspberrypi10:~$ sudo apt-get update
Hit:1 http://archive.raspberrypi.org/debian bullseye InRelease
Get:2 http://raspbian.raspberrypi.org/raspbian bullseye InRelease [15.0 kB]
Fetched 15.0 kB in 1s (11.7 kB/s)
Reading package lists... Done
```

El primer comando instala ffmpeg. El segundo comando inicia el video con una velocidad de fotogramas de 15fps usando el modo video4linux2, teniendo una resolución de 320x240 y usando el dispositivo desde /dev/video0. Una vez que lo ejecute, verá la alimentación de la cámara web en el monitor.



```
File Edit Tabs Help
pi@raspberrypi10:/dev
/usr/include/arm-linux-gnueabihf --enable-gpl --disable-stripping --enable-avresample --disable-filter
=resample --enable-gnutls --enable-ladspa --enable-libaom --enable-libaom --enable-libbluray --enable-l
ibbs2b --enable-libcaca --enable-libcdio --enable-libcodec2 --enable-libdavid --enable-libelite --enabl
e-libfontconfig --enable-libfreetype --enable-libfribidi --enable-libgme --enable-libgsm --enable-libja
ck --enable-libmp3lame --enable-libmysofa --enable-libopenjpeg --enable-libopenmpt --enable-libopus --e
nable-libpulse --enable-librabbitmq --enable-librsvg --enable-librubberband --enable-libshine --enable-
libsnapy --enable-libsoxr --enable-lspxex --enable-lsrt --enable-libssh --enable-libtheora --enabl
e-libtwolame --enable-libvidstab --enable-libvorbis --enable-libvpx --enable-libwavpack --enable-libweb
p --enable-libx265 --enable-libxml2 --enable-libxvid --enable-libzmq --enable-libzvbi --enable-lv2 --en
able-omx --enable-openal --enable-openc1 --enable-opengl --enable-sdl2 --enable-mm1 --enable-neon --en
able-rpi --enable-v4l2-request --enable-libudev --enable-epoxy --enable-pocketsphinx --enable-libdc1394
--enable-libdrm --enable-vout-drm --enable-libiec61883 --enable-chromaprint --enable-fre10r --enable-l
ibx264 --libdir=/usr/lib/arm-linux-gnueabihf/neon/vfp --cpu=cortex-a7 --arch=armv6t2 --disable-thumb --
enable-shared --disable-doc --disable-programs
libavutil      56. 51.100 / 56. 51.100
libavcodec     58. 91.100 / 58. 91.100
libavformat    58. 45.100 / 58. 45.100
libavdevice    58. 10.100 / 58. 10.100
libavfilter     7. 85.100 / 7. 85.100
libavresample   4.  0.  0 / 4.  0.  0
libswscale      5.  7.100 / 5.  7.100
libswresample   3.  7.100 / 3.  7.100
libpostproc    55.  7.100 / 55.  7.100
[video4linux2,v4l2 @ 0x930000630] The driver changed the time per frame from 1/15 to 1/30
Input #0, video4linux2,v4l2, from '/dev/video0':B sq= 0B f=0/0
Duration: N/A, start: 14607.710297, bitrate: 36864 kb/s
Stream #0:0: Video: rawvideo (YUY2 / 0x32595559), yuyv422, 320x240, 36864 kb/s, 30 fps, 30 tbr, 100
k tbn, 1000k tbc
14693.80 M-V: 0.977 fd=1960 aq= 0KB vq= 4052KB sq= 0B f=0/0
```

Como se puede observar ya tenemos visualización de la cámara:





Paso 2 Actualizaciones obligatorias en nuestro sistema

```
pi@raspberrypi10:~$ sudo apt update
Hit:1 http://archive.raspbian.org/debian bullseye InRelease
Get:2 http://raspbian.raspberrypi.org/raspbian bullseye InRelease [15.0 kB]
Fetched 15.0 kB in 1s (12.3 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
2 packages can be upgraded. Run 'apt list --upgradable' to see them.
pi@raspberrypi10:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following package was automatically installed and is no longer required:
  libfuse2
Use 'sudo apt autoremove' to remove it.
The following packages will be upgraded:
  libntfs-3g883 ntfs-3g
2 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 547 kB of archives.
After this operation, 0 B of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://raspbian.raspberrypi.org/raspbian bullseye/main armhf ntfs-3g armhf 1:2017.3.23AR.3-4+deb11u2 [379 kB]
Get:2 http://raspbian.raspberrypi.org/raspbian bullseye/main armhf libntfs-3g883 armhf 1:2017.3.23AR.3-4+deb11u2 [168 kB]
Fetched 547 kB in 2s (298 kB/s)
Reading changelogs... Done
```

```
pi@raspberrypi10:~$ sudo apt upgrade
libntfs-3g883 ntfs-3g
2 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 547 kB of archives.
After this operation, 0 B of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://raspbian.raspberrypi.org/raspbian bullseye/main armhf ntfs-3g armhf 1:2017.3.23AR.3-4+deb11u2 [379 kB]
Get:2 http://raspbian.raspberrypi.org/raspbian bullseye/main armhf libntfs-3g883 armhf 1:2017.3.23AR.3-4+deb11u2 [168 kB]
Fetched 547 kB in 2s (298 kB/s)
Reading changelogs... Done
(Reading database ... 205046 files and directories currently installed.)
Preparing to unpack .../ntfs-3g_1%3a2017.3.23AR.3-4+deb11u2_armhf.deb ...
Unpacking ntfs-3g (1:2017.3.23AR.3-4+deb11u2) over (1:2017.3.23AR.3-4+deb11u1) ...
Preparing to unpack .../libntfs-3g883_1%3a2017.3.23AR.3-4+deb11u2_armhf.deb ...
Unpacking libntfs-3g883 (1:2017.3.23AR.3-4+deb11u2) over (1:2017.3.23AR.3-4+deb11u1) ...
Setting up libntfs-3g883 (1:2017.3.23AR.3-4+deb11u2) ...
Setting up ntfs-3g (1:2017.3.23AR.3-4+deb11u2) ...
Processing triggers for libc-bin (2.31-13+rpt2+rpi1+deb11u2) ...
Processing triggers for man-db (2.9.4-2) ...
Processing triggers for initramfs-tools (0.140) ...
pi@raspberrypi10:~$
```



Paso 3 Crear dependencias y configuración del servicio de transmisión de video para la cámara web

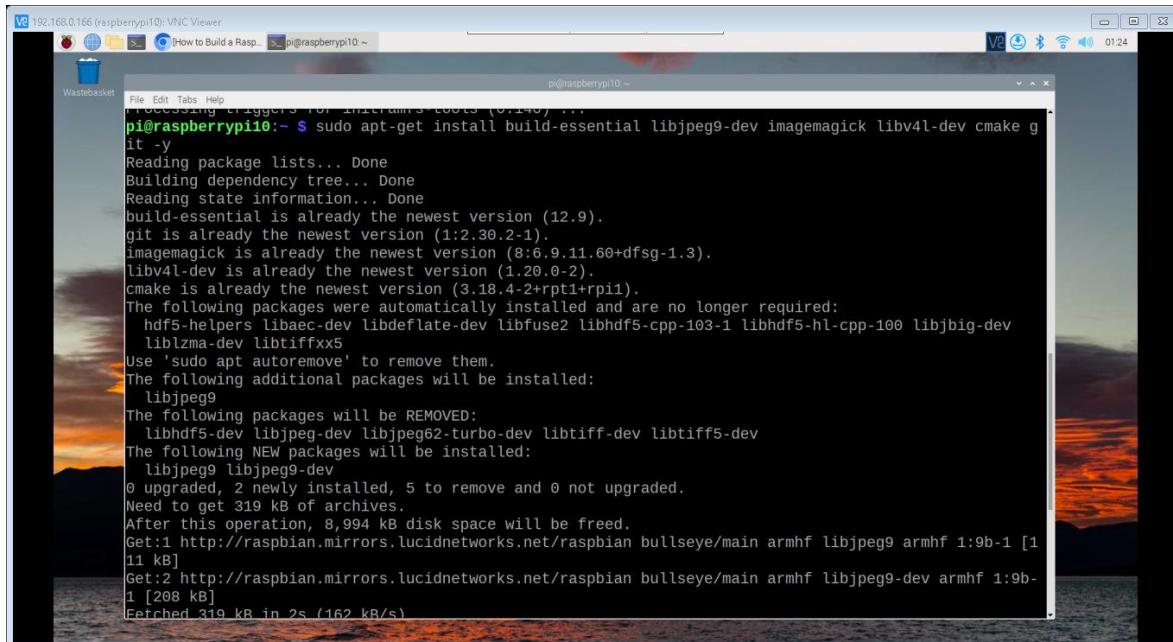
Aquí, el video de la raspberry pi se transmite en la red local. Cualquier dispositivo en esta red puede ver el video desde la cámara web. Para hacer esto, primero debe instalar el streamer mjpg.

Necesitamos descargar algunas cosas primero antes de poder construir mjpg-streamer. Usaremos el repositorio Git de mjpg-streamer que fue creado por [jacksonliam](https://github.com/jacksonliam/mjpg-streamer):

<https://github.com/jacksonliam/mjpg-streamer>

Ahora instalaremos nuestras dependencias antes de descargar y compilar el código mjpg-streamer con la siguiente línea en la terminal:

```
sudo apt-get install build-essential libjpeg9-dev imagemagick libv4l-dev cmake git -y
```



```
pi@raspberrypi0:~ $ sudo apt-get install build-essential libjpeg9-dev imagemagick libv4l-dev cmake git -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
build-essential is already the newest version (12.9).
git is already the newest version (1:2.30.2-1).
imagemagick is already the newest version (8:6.9.11.60+dfsg-1.3).
libv4l-dev is already the newest version (1.20.0-2).
cmake is already the newest version (3.18.4-2+rpt1+rpi1).
The following packages were automatically installed and are no longer required:
  hdf5-helpers libaec-dev libdeflate-dev libfuse2 libhdf5-cpp-103-1 libhdf5-hl-cpp-100 libjbig-dev
  liblzma-dev libtiffxx5
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libjpeg9
The following packages will be REMOVED:
  libhdf5-dev libjpeg-dev libjpeg62-turbo-dev libtiff-dev libtiff5-dev
The following NEW packages will be installed:
  libjpeg9 libjpeg9-dev
0 upgraded, 2 newly installed, 5 to remove and 0 not upgraded.
Need to get 319 kB of archives.
After this operation, 8,994 kB disk space will be freed.
Get:1 http://raspbian.mirrors.lucidnetworks.net/raspbian bullseye/main armhf libjpeg9 armhf 1:9b-1 [11 kB]
Get:2 http://raspbian.mirrors.lucidnetworks.net/raspbian bullseye/main armhf libjpeg9-dev armhf 1:9b-1 [208 kB]
Fetched 319 kB in 2s (162 kB/s)
```





```
File Edit Tabs Help  
DOCKERFILE  makedeb.sh  Mjpg_Streamer.n postinstall.sh  START.SN  UTILS.N  
pi@raspberrypi10:~/proyecto-robot-webcam/mjpg-streamer/mjpg-streamer-experimental  
[ -d _build ] || mkdir _build  
[ -f _build/Makefile ] || (cd _build && cmake -DCMAKE_BUILD_TYPE=Release ..)  
-- The C compiler identification is GNU 10.2.1  
-- Detecting C compiler ABI info  
-- Detecting C compiler ABI info - done  
-- Check for working C compiler: /usr/bin/cc - skipped  
-- Detecting C compile features  
-- Detecting C compile features - done  
CMAKE_BUILD_TYPE = Release  
-- Looking for include file sys/inotify.h  
-- Looking for include file sys/inotify.h - found  
CMake Warning at plugins/input_opencv/CMakeLists.txt:4 (find_package):  
  By not providing "Findopencv.cmake" in CMAKE_MODULE_PATH this project has  
  asked CMake to find a package configuration file provided by "OpenCV", but  
  CMake did not find one.  
  
  Could not find a package configuration file provided by "OpenCV" with any  
  of the following names:  
  
    OpenCVConfig.cmake  
    opencv-config.cmake  
  
  Add the installation prefix of "OpenCV" to CMAKE_PREFIX_PATH or set  
  "OpenCV_DIR" to a directory containing one of the above files. If "OpenCV"  
  provides a separate development package or SDK, be sure it has been  
  installed.
```

```
File Edit Tabs Help  
pi@raspberrypi10:~/proyecto-robot-webcam/mjpg-streamer/mjpg-streamer-experimental  
CMake Warning (dev) at /usr/share/cmake-3.18/Modules/FindPackageHandleStandardArgs.cmake:273 (message):  
  The package name passed to 'find_package_handle_standard_args' (PkgConfig)  
  does not match the name of the calling package (Gphoto2). This can lead to  
  problems in calling code that expects 'find_package' result variables  
  (e.g., '_FOUND') to follow a certain pattern.  
Call stack (most recent call first):  
  /usr/share/cmake-3.18/Modules/FindPkgConfig.cmake:59 (find_package_handle_standard_args)  
  cmake/FindGphoto2.cmake:21 (include)  
  plugins/input_pp2/CMakeLists.txt:2 (find_package)  
This warning is for project developers. Use -Wno-dev to suppress it.  
  
  -- Found PkgConfig: /usr/bin/pkg-config (found version "0.29.2")  
  -- Checking for module 'libgphoto2'  
  -- No package 'libgphoto2' found  
CMake Warning (dev) at /usr/share/cmake-3.18/Modules/FindPackageHandleStandardArgs.cmake:273 (message):  
  The package name passed to 'find_package_handle_standard_args' (GPHOTO2)  
  does not match the name of the calling package (Gphoto2). This can lead to  
  problems in calling code that expects 'find_package' result variables  
  (e.g., '_FOUND') to follow a certain pattern.  
Call stack (most recent call first):  
  cmake/FindGphoto2.cmake:44 (find_package_handle_standard_args)  
  plugins/input_pp2/CMakeLists.txt:2 (find_package)  
This warning is for project developers. Use -Wno-dev to suppress it.  
  
  -- Could NOT find GPHOTO2 (missing: GPHOTO2_LIBRARY GPHOTO2_INCLUDE_DIR)  
  -- Looking for include file linux/videodev2.h  
  -- Looking for include file linux/videodev2.h - found  
  -- Looking for pthread.h
```

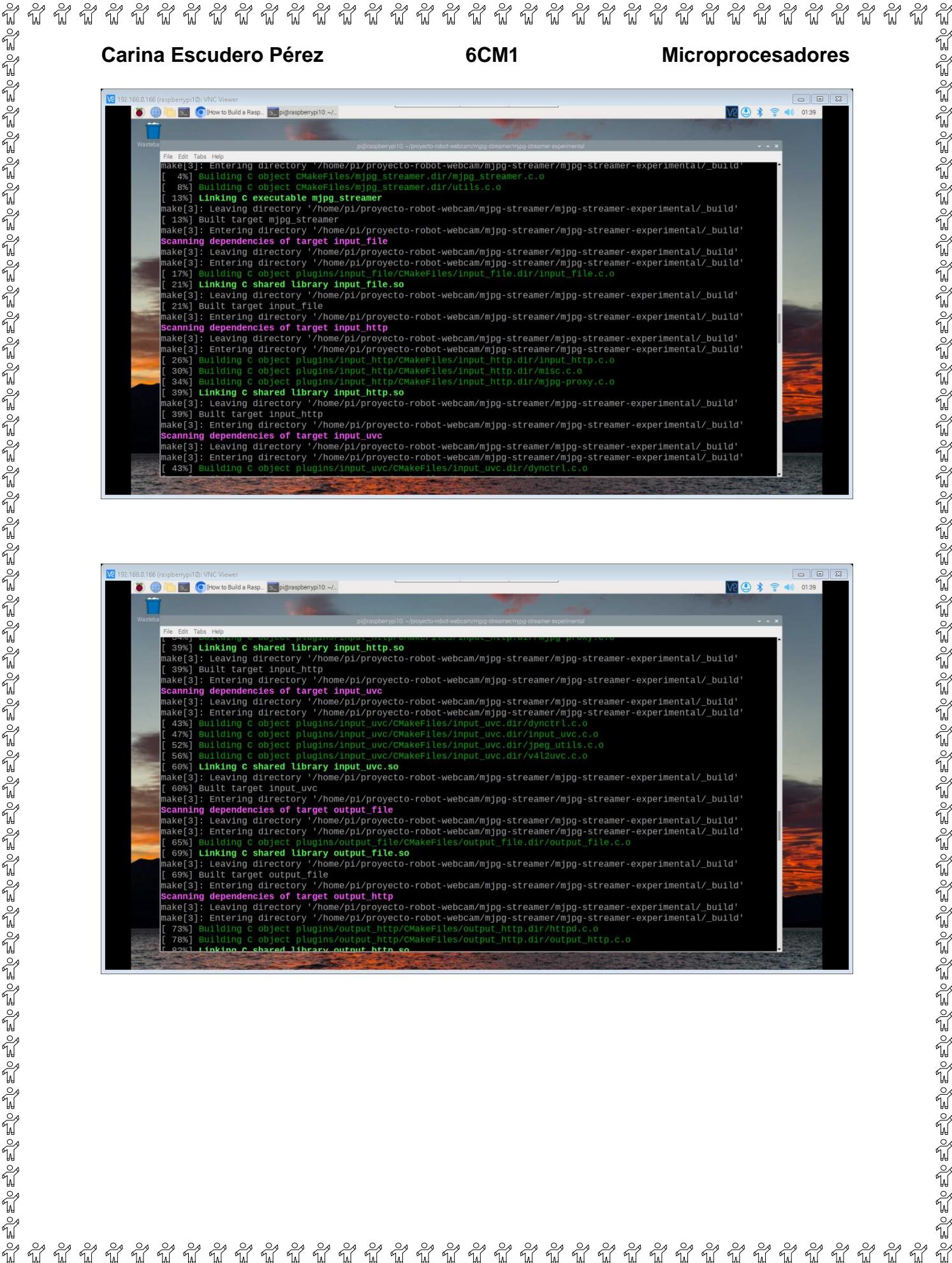




```
V8 192.168.0.166 (raspberrypi10): VNC Viewer  
File Edit Tabs Help  
pi@raspberrypi10: ~/proyecto-robot-webcam/mjpg-streamer/mjpg-streamer-experimental  
-- Looking for pthread.h  
-- Looking for pthread.h - found  
-- Performing Test CMAKE_HAVE_LIBC_PTHREAD  
-- Performing Test CMAKE_HAVE_LIBC_PTHREAD - Failed  
-- Looking for pthread_create in pthreads  
-- Looking for pthread_create in pthreads - not found  
-- Looking for pthread_create in pthread  
-- Looking for pthread_create in pthread - found  
-- Found Threads: TRUE  
-- Could NOT find SDL (missing: SDL_LIBRARY SDL_INCLUDE_DIR)  
-- Could NOT find protobuf-c (missing: PROTOBUF_C_LIBRARY PROTOBUF_C_INCLUDE_DIR)  
-- The following features have been enabled:  
* PLUGIN_INPUT_FILE, File input plugin  
* PLUGIN_INPUT_HTTP, HTTP input proxy plugin  
* PLUGIN_INPUT_UVC, Video 4 Linux input plugin  
* PLUGIN_OUTPUT_FILE, File output plugin  
* PLUGIN_OUTPUT_HTTP, HTTP server output plugin  
* PLUGIN_OUTPUT_RTSP, RTSP output plugin  
* PLUGIN_OUTPUT_UDP, UDP output stream plugin  
-- The following OPTIONAL packages have been found:  
* Threads  
-- The following features have been disabled:  
* WXP_COMPAT, Enable compatibility with WebcamXP  
* PLUGIN_INPUT_OPENCV, OpenCV input plugin (unmet dependencies)
```

```
V8 192.168.0.166 (raspberrypi10): VNC Viewer  
File Edit Tabs Help  
pi@raspberrypi10: ~/proyecto-robot-webcam/mjpg-streamer/mjpg-streamer-experimental  
-- The following features have been disabled:  
* WXP_COMPAT, Enable compatibility with WebcamXP  
* PLUGIN_INPUT_OPENCV, OpenCV input plugin (unmet dependencies)  
* PLUGIN_INPUT_RASPICAM, Raspberry Pi input camera plugin (unmet dependencies)  
* PLUGIN_INPUT_PTP2, PTP2 input plugin (unmet dependencies)  
* ENABLE_HTTP_MANAGEMENT, Enable experimental HTTP management option  
* PLUGIN_OUTPUT_VIEWER, SDL output viewer plugin (unmet dependencies)  
* PLUGIN_OUTPUT_ZMQSERVER, ZMQ Server output plugin (unmet dependencies)  
-- The following OPTIONAL packages have not been found:  
* OpenCV  
* Gphoto2  
* SDL  
-- configuring done  
-- Generating done  
-- Build files have been written to: /home/pi/proyecto-robot-webcam/mjpg-streamer/mjpg-streamer-experimental/_build  
make -c _build  
make[1]: Entering directory '/home/pi/proyecto-robot-webcam/mjpg-streamer/mjpg-streamer-experimental/_build'  
make[2]: Entering directory '/home/pi/proyecto-robot-webcam/mjpg-streamer/mjpg-streamer-experimental/_build'  
make[3]: Entering directory '/home/pi/proyecto-robot-webcam/mjpg-streamer/mjpg-streamer-experimental/_build'  
Scanning dependencies of target mjpg_streamer  
make[3]: Leaving directory '/home/pi/proyecto-robot-webcam/mjpg-streamer/mjpg-streamer-experimental/_build'  
make[3]: Entering directory '/home/pi/proyecto-robot-webcam/mjpg-streamer/mjpg-streamer-experimental/_build'  
[ 4%] Building C object CMakeFiles/mjpg_streamer.dir/mjpg_streamer.c.o  
[ 8%] Building C object CMakeFiles/mjpg_streamer.dir/utils.c.o  
[ 13%] Linking C executable mjpg_streamer
```

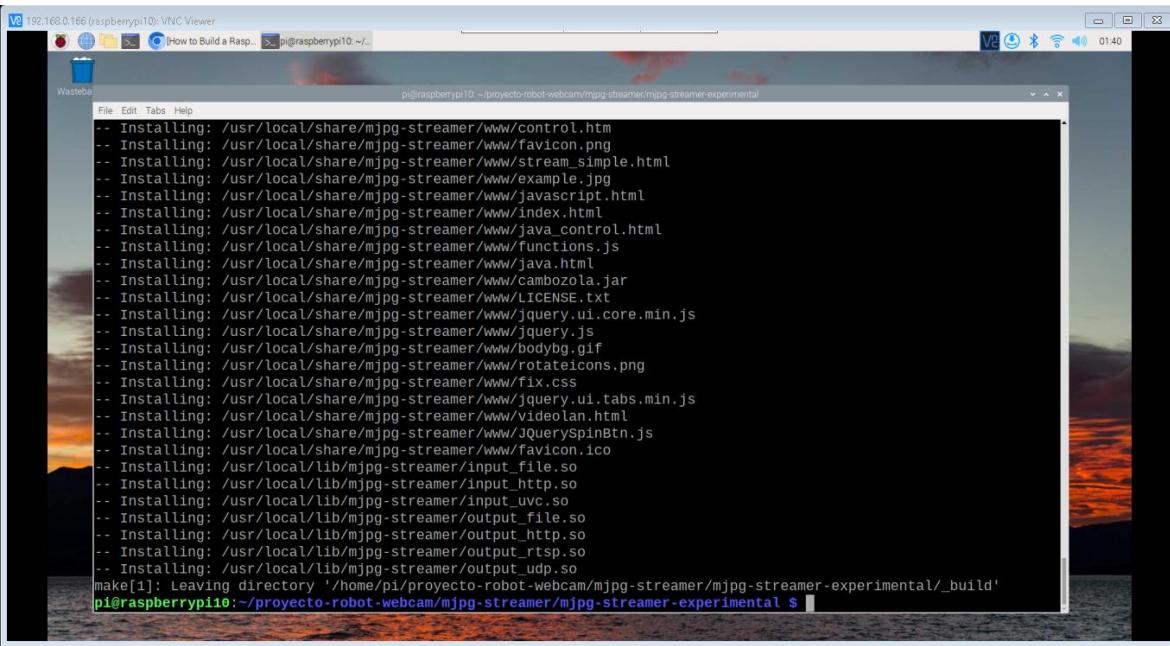
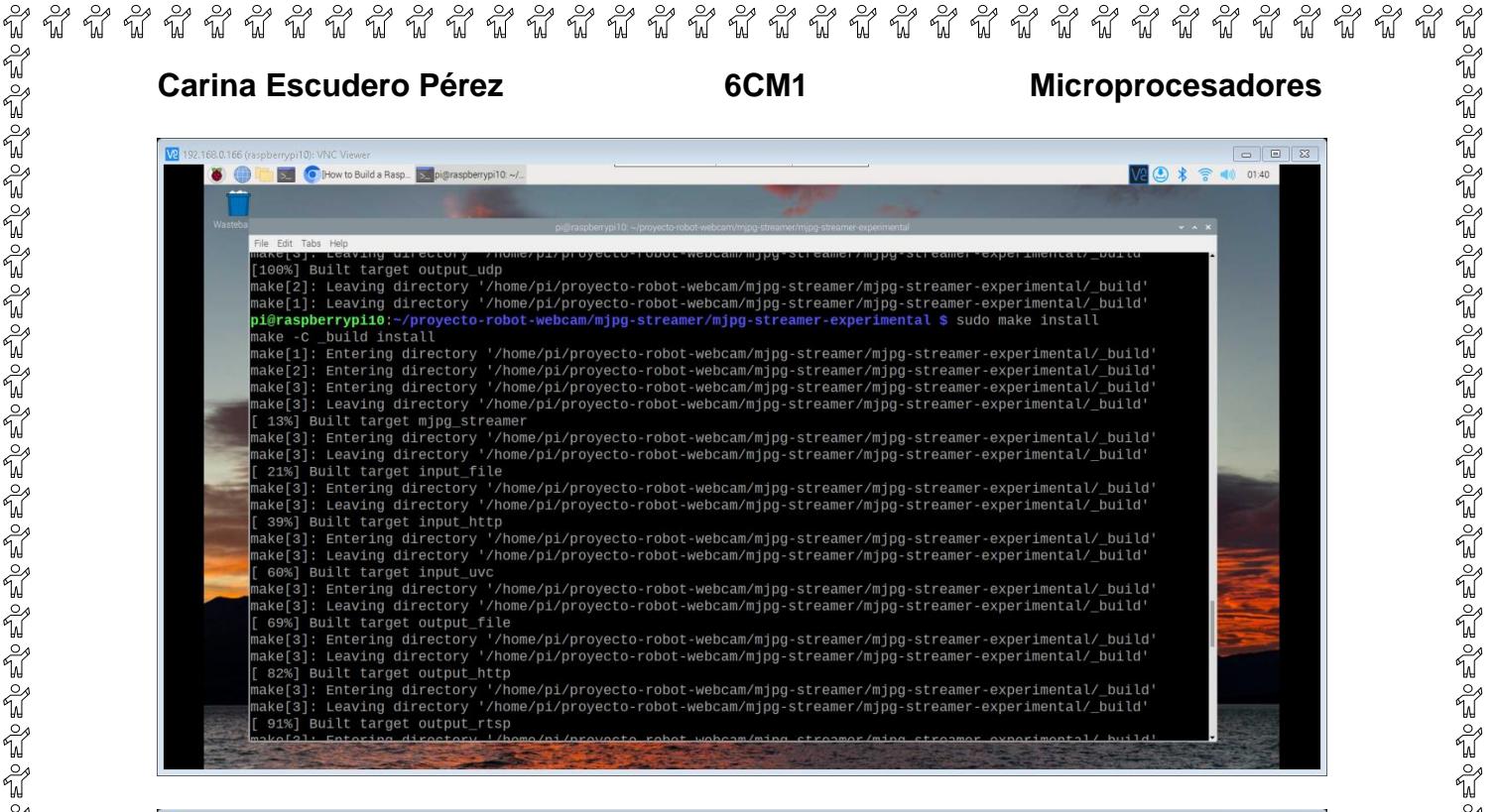




Carina Escudero Pérez

6CM1

Microprocesadores



El make install copiará binarios, bibliotecas y las páginas www a la estructura de directorios /usr/local/:

/usr/local/bin/mjpg_streamer – El binario primario

/usr/local/lib/mjpg-streamer/ – El directorio de módulos de entrada/salida

/usr/local/share/mjpg-streamer/www – La interfaz del servidor www

Paso 5 comprobar la instalación

Probar la compilación conectando una cámara web al puerto USB de la Raspberry Pi, la ejecución de dmesg muestra que se cargó correctamente:

```
[ 1275.662775] usb 1-1: New USB device found, idVendor=046d, idProduct=082d,
bcdDevice= 0.11
```

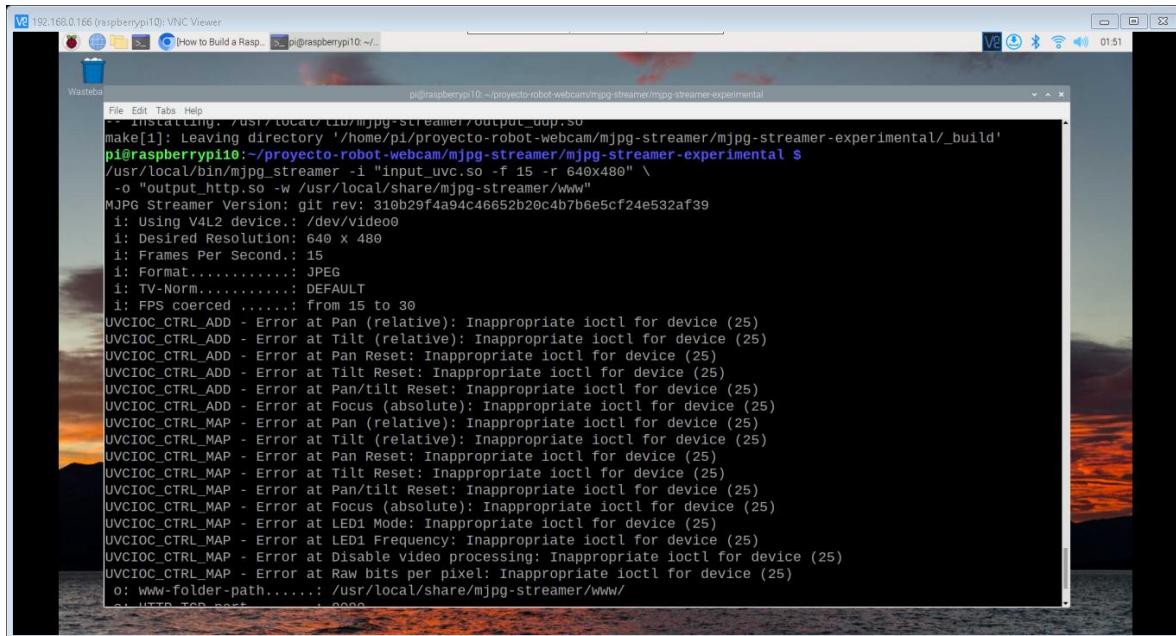
```
[ 1275.662796] usb 1-1: New USB device strings: Mfr=0, Product=2, SerialNumber=1
```

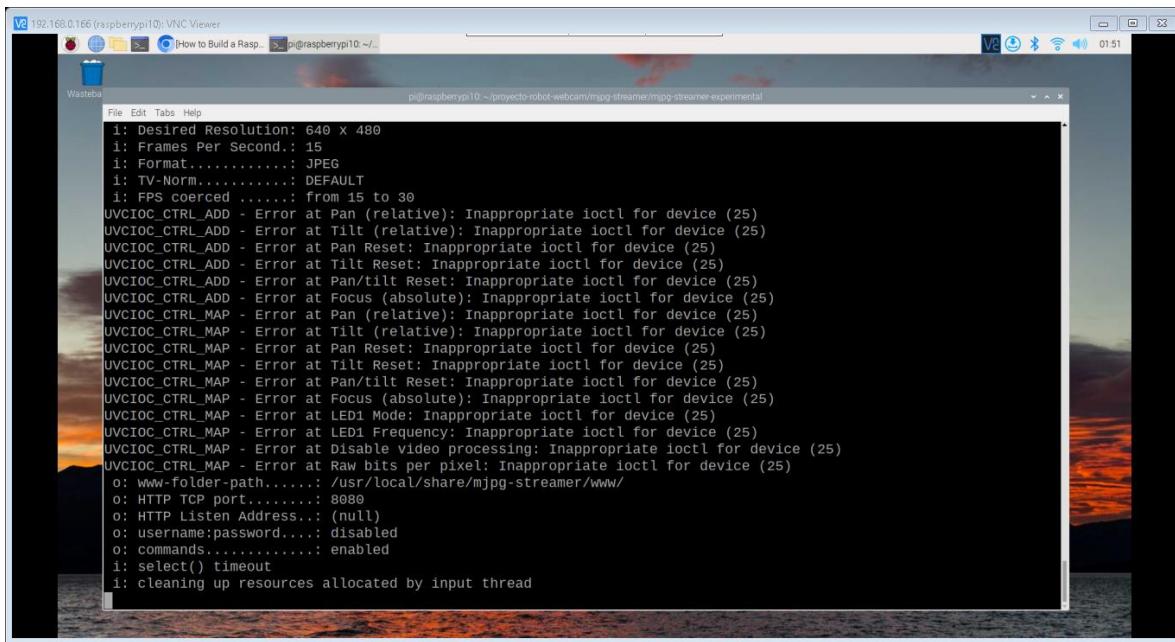
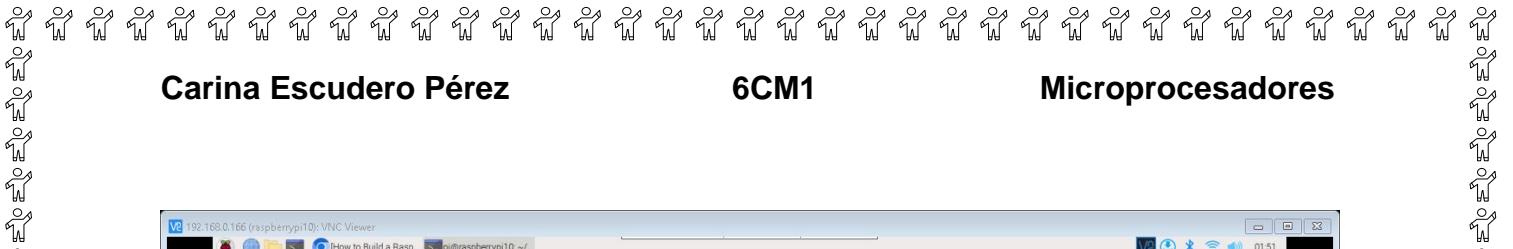
```
[ 1275.662806] usb 1-1: Product: HD Pro Webcam C920
```

Vamos a probar tratando de ejecutar mjpg-streamer contra el complemento input_uvc.so, ya que no estoy usando una Raspi-Camera, sino una USB.

Saldremos usando el plugin http. Así que mi línea de comandos de prueba se ve así:

```
/usr/local/bin/mjpg_streamer -i "input_uvc.so -f 15 -r 640x480" \
-o "output_http.so -w /usr/local/share/mjpg-streamer/www"
```





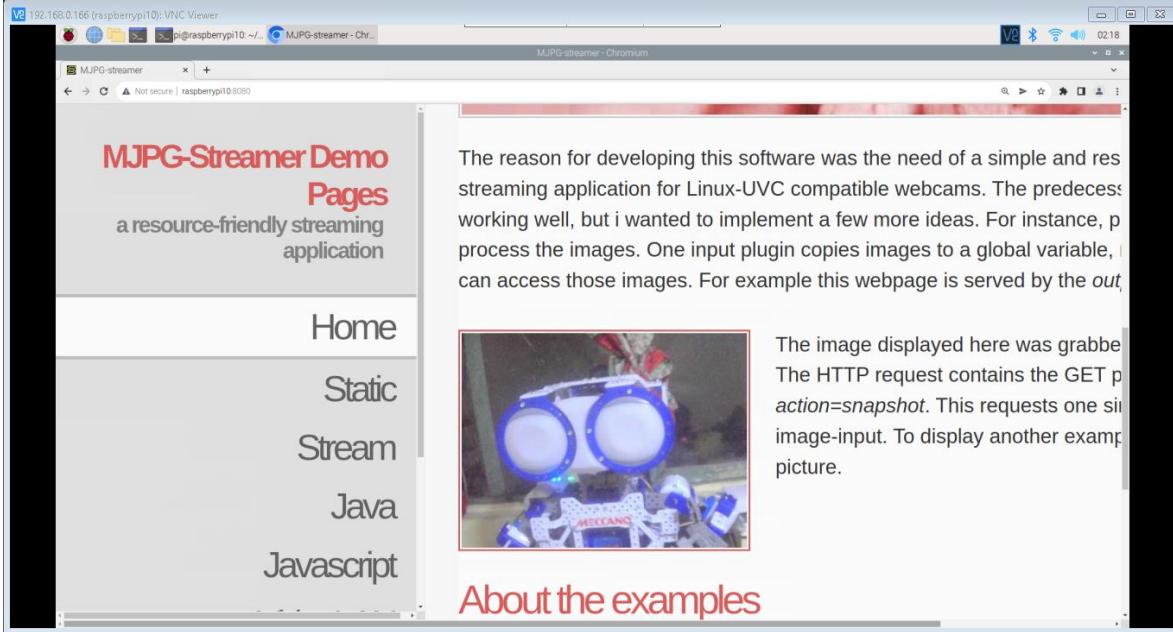
Nuestras variables clave para ajustar son **-f para la velocidad de fotogramas y -r para la resolución.**

Pasó 6 prueba rápida

Para una prueba rápida, me dirijo a mi navegador y lo apunto al servidor www en:

- <http://raspberrypi10:8080/>
- <http://localhost:8080/>
- <http://192.168.0.166:8080/>





Ahora puede presionar CTRL-C en su ventana SSH (o terminal) y salir de la transmisión. Después de este punto, podría eliminar los archivos de compilación que descargué.

Pasó 7 ajuste de la configuración

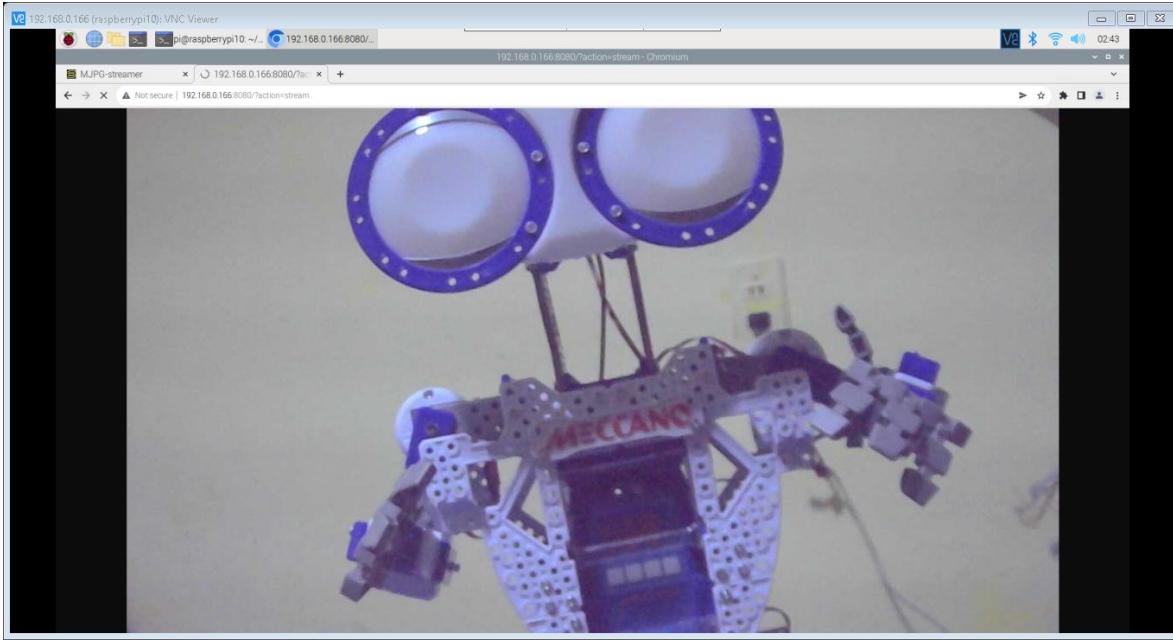
Ejecutar la cámara a su resolución prevista y al menos una velocidad de fotogramas de 30 fps. Para ello modificamos la línea de comandos:

```
#30fps, 1080 HD
```

```
/usr/local/bin/mjpg_streamer -i "input_uvc.so -f 30 -r 1920x1080" \
-o "output_http.so -w /usr/local/share/mjpg-streamer/www"
```

Podemos cargar nuestra propia transmisión en VLC abriendo la ruta de red directa a la transmisión:

- <http://raspberrypi10:8080?action=stream>
- <http://localhost:8080?action=stream>
- <http://192.168.0.166:8080?action=stream>



Eso funcionó, y obtengo un retraso de aproximadamente 1-2 segundos de Pi a VLC, no se recomienda usar audio en este tipo de transmisiones por el tiempo de retardo.

Tenga cuidado con los fotogramas y la resolución: observe su CPU, pero también asegúrese de no ver mensajes sobre fps que se "coaccionan" hacia abajo o hacia arriba. Su kilometraje puede variar, y es posible que tenga que ajustar aún más la configuración de formato, fps y resolución, consulte la sede de mjpg-streamer en su página de git hub (consulte las referencias a continuación). También puede ver los diferentes modos disponibles para su cámara utilizando este comando:

```
v4l2-ctl --list-formats-ext
```





Crear nuestro repositorio en Github donde vamos a estar subiendo las modificaciones a nuestro proyecto.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner escuderopc / Repository name proyecto-robo

Great repository names are projeto-robo is available. Need inspiration? How about scaling-octo-fiesta?

Description (optional)

Public Anyone on the internet can see this repository. You choose who can commit.

Private You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

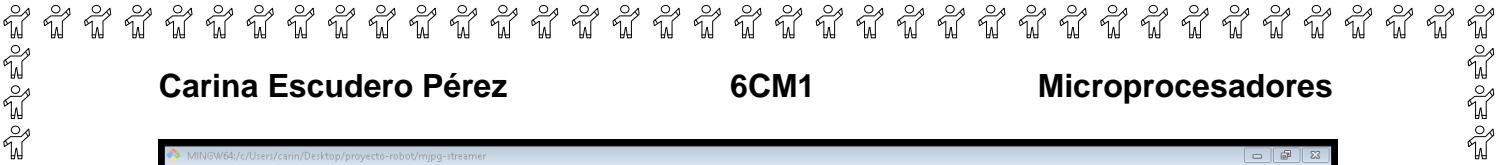
Add a README file This is where you can write a long description for your project. Learn more.

Add .gitignore

Subir nuestros archivos que tenemos en la computadora en el repositorio de Github

```
carin@DESKTOP-JJ5EB66 MINGW64 ~/Users/carin/Desktop/proyecto-robot/mjpg-streamer
$ ls
README.md
carin@DESKTOP-JJ5EB66 MINGW64 ~/Users/carin/Desktop/proyecto-robot/mjpg-streamer (main)
$ git add .
carin@DESKTOP-JJ5EB66 MINGW64 ~/Users/carin/Desktop/proyecto-robot/mjpg-streamer (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.
Changes to be committed:
  (use "git restore <file>..." to unstage)
    new file:   .gitignore
    new file:   README.md
    new file:   mjpg-streamer-experimental/CMakeLists.txt
    new file:   mjpg-streamer-experimental/Dockerfile
    new file:   mjpg-streamer-experimental/LICENSE
    new file:   mjpg-streamer-experimental/Makefile
    new file:   mjpg-streamer-experimental/mjpg-streamer.h
    more files...
    new file:   mjpg-streamer-experimental/7200
    new file:   mjpg-streamer-experimental/cmake/FindProtoBuf.cmake
    new file:   mjpg-streamer-experimental/cmake/FindProtobuf.cmake
    new file:   mjpg-streamer-experimental/cmake/FindZeroMQ.cmake
    new file:   mjpg-streamer-experimental/cmake/mpg_streamer_utils.cmake
    new file:   mjpg-streamer-experimental/docker/start.sh
    new file:   mjpg-streamer-experimental/include/mpg_streamer.h
    new file:   mjpg-streamer-experimental/mpg_streamer.c
    new file:   mjpg-streamer-experimental/mpg_streamer.h
    new file:   mjpg-streamer-experimental/mpg_streamer@.service
    new file:   mjpg-streamer-experimental/plugins/input.h
    new file:   mjpg-streamer-experimental/plugins/input_control/CMakefile
    new file:   mjpg-streamer-experimental/plugins/input_control/include.h
    more files...
    new file:   mjpg-streamer-experimental/plugins/input_control/include/mpg_streamer.h
    new file:   mjpg-streamer-experimental/plugins/input_control/input_uvc.c
    new file:   mjpg-streamer-experimental/plugins/input_control/uvc_compat.h
    new file:   mjpg-streamer-experimental/plugins/input_control/uvcvideo.h
    new file:   mjpg-streamer-experimental/plugins/input_file/CMakeLists.txt
    new file:   mjpg-streamer-experimental/plugins/input_file/include.h
    new file:   mjpg-streamer-experimental/plugins/input_file/include/mpg_streamer.h
    more files...
    new file:   mjpg-streamer-experimental/plugins/input_http/CMakeLists.txt
    new file:   mjpg-streamer-experimental/plugins/input_http/include.c
    more files...
    new file:   mjpg-streamer-experimental/plugins/input_http/include/mpg_streamer.h
    new file:   mjpg-streamer-experimental/plugins/input_http/http.c
    new file:   mjpg-streamer-experimental/plugins/input_http/misc.c
    new file:   mjpg-streamer-experimental/plugins/input_http/misc.h
    new file:   mjpg-streamer-experimental/plugins/input_http/mpg-proxy.c
    new file:   mjpg-streamer-experimental/plugins/input_http/mpg-proxy.h
    new file:   mjpg-streamer-experimental/plugins/input_http/version.h
    new file:   mjpg-streamer-experimental/plugins/input_opencv/CMakeLists.txt
```





```
MINGW64/c/Users/carin/Desktop/proyecto-robot/mjpg-streamer
new file: mjpg-streamer-experimental/plugins/output_viewer/output_viewer.c
new file: mjpg-streamer-experimental/plugins/output_zmserver/CMakelists.txt
new file: mjpg-streamer-experimental/plugins/output_zmserver/README.md
new file: mjpg-streamer-experimental/plugins/output_zmserver/output_zmserver.c
new file: mjpg-streamer-experimental/plugins/output_zmserver/output_zmserver.h
new file: mjpg-streamer-experimental/plugins/output_zmserver/output_zmserver/package.proto
new file: mjpg-streamer-experimental/ostinstall.sh
new file: mjpg-streamer-experimental/scripts/README.md
new file: mjpg-streamer-experimental/scripts/make_deb.sh
new file: mjpg-streamer-experimental/scripts/mjpg-streamer.default
new file: mjpg-streamer-experimental/scripts/mjpg-streamer.init
new file: mjpg-streamer-experimental/service/mjpg-streamer.service
new file: mjpg-streamer-experimental/start.sh
new file: mjpg-streamer-experimental/utils.h
new file: mjpg-streamer-experimental/www/JQuerySpinBtn.css
new file: mjpg-streamer-experimental/www/JQuerySpinBtn.js
new file: mjpg-streamer-experimental/www/LICENSE.txt
new file: mjpg-streamer-experimental/www/index.html
new file: mjpg-streamer-experimental/www/favicon.ico
new file: mjpg-streamer-experimental/www/favicon.png
new file: mjpg-streamer-experimental/www/functions.js
new file: mjpg-streamer-experimental/www/index.html
new file: mjpg-streamer-experimental/www/java.html
new file: mjpg-streamer-experimental/www/java_control.html
new file: mjpg-streamer-experimental/www/java_simple.html
new file: mjpg-streamer-experimental/www/javascript.html
new file: mjpg-streamer-experimental/www/javascript/motiondetection.html
new file: mjpg-streamer-experimental/www/javascript/simple.html
new file: mjpg-streamer-experimental/www/jquery.js
new file: mjpg-streamer-experimental/www/jquery.rotate.js
new file: mjpg-streamer-experimental/www/jquery_ui.core.min.js
new file: mjpg-streamer-experimental/www/jquery_ui.custom.css
new file: mjpg-streamer-experimental/www/jquery_ui.tabs.min.js
new file: mjpg-streamer-experimental/www/jquery_ui.get.min.js
new file: mjpg-streamer-experimental/www/mecano.jpg
new file: mjpg-streamer-experimental/www/rotateIcons.png
new file: mjpg-streamer-experimental/www/sidebarbg.gif
new file: mjpg-streamer-experimental/www/spinbtn_update.gif
new file: mjpg-streamer-experimental/www/static.html
new file: mjpg-streamer-experimental/www/style.html
new file: mjpg-streamer-experimental/www/stream.html
new file: mjpg-streamer-experimental/www/stream_simple.html
new file: mjpg-streamer-experimental/www/style.css
new file: mjpg-streamer-experimental/www/videoLAN.html
```

```
carinDESKTOP-J35EB6 MINGW64 /c/Users/carin/Desktop/proyecto-robot/mjpg-streamer (main)
$ git commit -m 'Códigos para el servidor robot con webcam'
[main ba7abdb] Códigos para el servidor robot con webcam
 1 file changed, 1 insertion(+)
 create mode 100644 mjpg-streamer/.gitignore
 create mode 100644 mjpg-streamer/README.md
 create mode 100644 mjpg-streamer/mjpg-streamer-experimental/CMakelists.txt
 create mode 100644 mjpg-streamer/mjpg-streamer-experimental/Dockerfile
 create mode 100644 mjpg-streamer/mjpg-streamer-experimental/LICENSE
 create mode 100644 mjpg-streamer/mjpg-streamer-experimental/Makfile
 create mode 100644 mjpg-streamer/mjpg-streamer-experimental/README.md
 create mode 100644 mjpg-streamer/mjpg-streamer-experimental/TODO
 create mode 100644 mjpg-streamer/mjpg-streamer-experimental/cmake/Findphoto2.cmake
 create mode 100644 mjpg-streamer/mjpg-streamer-experimental/cmake/Findprotobuf-c.cmake
 create mode 100644 mjpg-streamer/mjpg-streamer-experimental/cmake/FindZeroMQ.cmake
 create mode 100644 mjpg-streamer/mjpg-streamer-experimental/cmake/mpg_streamer_utils.cmake
 create mode 100644 mjpg-streamer/mjpg-streamer-experimental/cmake/Findzmq.cmake
 create mode 100644 mjpg-streamer/mjpg-streamer-experimental/cmake/zmqdb.h
 create mode 100644 mjpg-streamer/mjpg-streamer-experimental/mpg_stream.c
 create mode 100644 mjpg-streamer/mjpg-streamer-experimental/mpg_stream.h
 create mode 100644 mjpg-streamer/mjpg-streamer-experimental/mpg_stream@.service
 create mode 100644 mjpg-streamer/mjpg-streamer-experimental/plugins/input.h
 create mode 100644 mjpg-streamer/mjpg-streamer-experimental/plugins/input_http@.h
 create mode 100644 mjpg-streamer/mjpg-streamer-experimental/plugins/input_http@.control
 create mode 100644 mjpg-streamer/mjpg-streamer-experimental/plugins/input_http@.control/dynctrl.h
 create mode 100644 mjpg-streamer/mjpg-streamer-experimental/plugins/input_control.h
 create mode 100644 mjpg-streamer/mjpg-streamer-experimental/plugins/input_uvc.c
 create mode 100644 mjpg-streamer/mjpg-streamer-experimental/plugins/input_uvc_compat.h
 create mode 100644 mjpg-streamer/mjpg-streamer-experimental/plugins/input_control/uvcvideo.h
 create mode 100644 mjpg-streamer/mjpg-streamer-experimental/plugins/input_file/CMakelists.txt
 create mode 100644 mjpg-streamer/mjpg-streamer-experimental/plugins/input_file/Makefile
 create mode 100644 mjpg-streamer/mjpg-streamer-experimental/plugins/input_file/file.c
 create mode 100644 mjpg-streamer/mjpg-streamer-experimental/plugins/input_http/CMakelists.txt
 create mode 100644 mjpg-streamer/mjpg-streamer-experimental/plugins/input_http/http.c
 create mode 100644 mjpg-streamer/mjpg-streamer-experimental/plugins/input_http/msc.c
 create mode 100644 mjpg-streamer/mjpg-streamer-experimental/plugins/input_http/msc.h
 create mode 100644 mjpg-streamer/mjpg-streamer-experimental/plugins/input_http/mpg-proxy.c
 create mode 100644 mjpg-streamer/mjpg-streamer-experimental/plugins/input_http/mpg-proxy.h
 create mode 100644 mjpg-streamer/mjpg-streamer-experimental/plugins/input_opencv/filter_cvfilter_py.h
 create mode 100644 mjpg-streamer/mjpg-streamer-experimental/plugins/input_opencv/filter_cvfilter_py.h
 create mode 100644 mjpg-streamer/mjpg-streamer-experimental/plugins/input_opencv/filter_cvfilter_py/CMakelists.txt
 create mode 100644 mjpg-streamer/mjpg-streamer-experimental/plugins/input_opencv/filter_cvfilter_py/FindNumpy.cmake
 create mode 100644 mjpg-streamer/mjpg-streamer-experimental/plugins/input_opencv/filter_cvfilter_py/conversion.cpp
 create mode 100644 mjpg-streamer/mjpg-streamer-experimental/plugins/input_opencv/filter_cvfilter_py/conversion.h
```

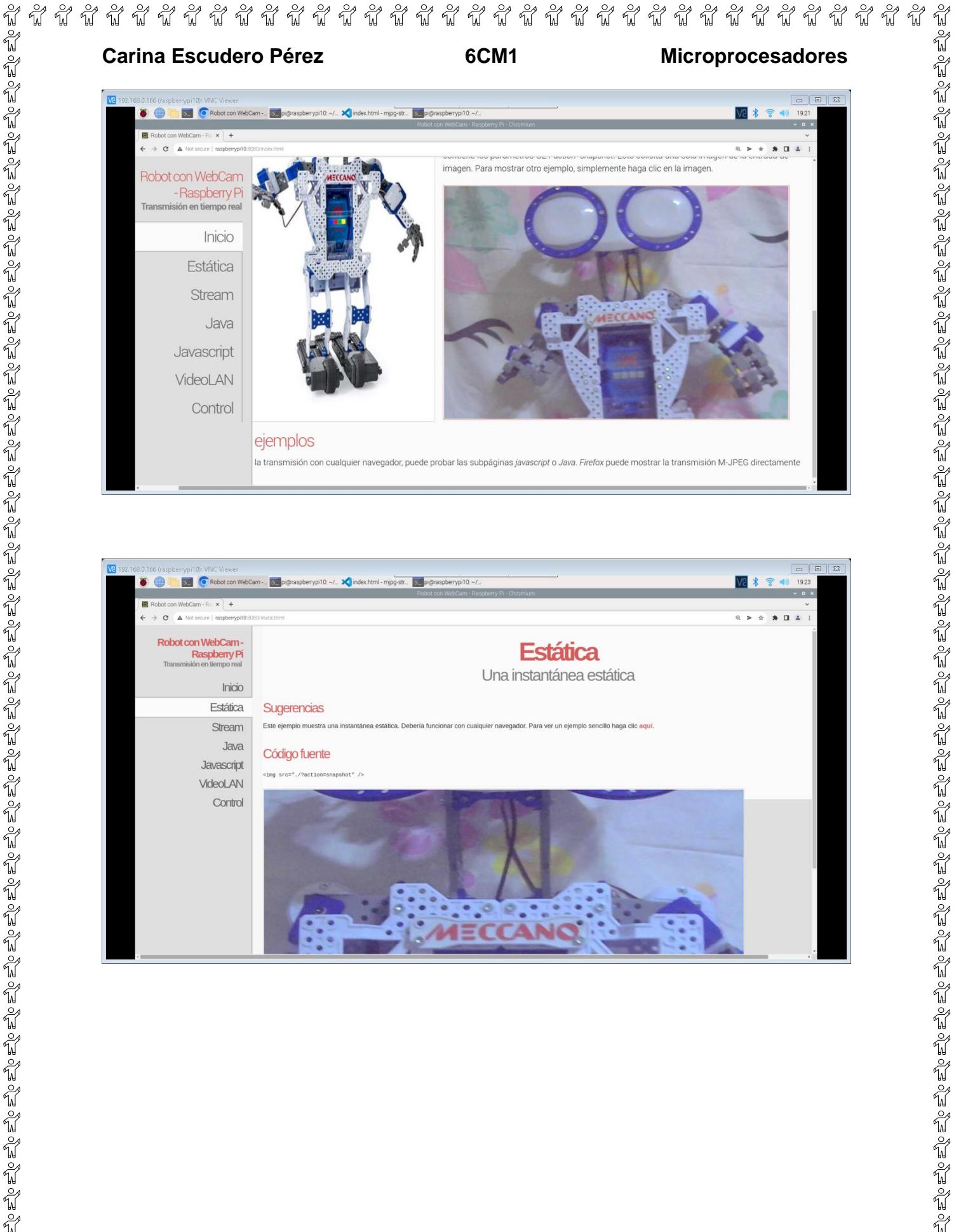




```
MINGW64 /c/Users/carin/Desktop/proyecto-robot/mjpg-streamer
create mode 100644 mjpg-streamer/mjpg-streamer-experimental/utils.h
create mode 100644 mjpg-streamer/mjpg-streamer-experimental/www/JQuerySpinBtn.css
create mode 100644 mjpg-streamer/mjpg-streamer-experimental/www/JQuerySpinBtn.js
create mode 100644 mjpg-streamer/mjpg-streamer-experimental/www/LiveVideo.js
create mode 100644 mjpg-streamer/mjpg-streamer-experimental/www/lobby.html
create mode 100644 mjpg-streamer/mjpg-streamer-experimental/www/cambozola.jar
create mode 100644 mjpg-streamer/mjpg-streamer-experimental/www/control.htm
create mode 100644 mjpg-streamer/mjpg-streamer-experimental/www/example.jpg
create mode 100644 mjpg-streamer/mjpg-streamer-experimental/www/favicon.ico
create mode 100644 mjpg-streamer/mjpg-streamer-experimental/www/favicon.png
create mode 100644 mjpg-streamer/mjpg-streamer-experimental/www/favicon.css
create mode 100644 mjpg-streamer/mjpg-streamer-experimental/www/functions.js
create mode 100644 mjpg-streamer/mjpg-streamer-experimental/www/index.html
create mode 100644 mjpg-streamer/mjpg-streamer-experimental/www/java.html
create mode 100644 mjpg-streamer/mjpg-streamer-experimental/www/java_control.html
create mode 100644 mjpg-streamer/mjpg-streamer-experimental/www/java_simple.html
create mode 100644 mjpg-streamer/mjpg-streamer-experimental/www/javascript.html
create mode 100644 mjpg-streamer/mjpg-streamer-experimental/www/javascipt.html
create mode 100644 mjpg-streamer/mjpg-streamer-experimental/www/meccano.html
create mode 100644 mjpg-streamer/mjpg-streamer-experimental/www/meccano_dectection.html
create mode 100644 mjpg-streamer/mjpg-streamer-experimental/www/meccano_dectectioncomplete.html
create mode 100644 mjpg-streamer/mjpg-streamer-experimental/www/jquery.js
create mode 100644 mjpg-streamer/mjpg-streamer-experimental/www/jquery_rotate.js
create mode 100644 mjpg-streamer/mjpg-streamer-experimental/www/jquery_ui.core.min.js
create mode 100644 mjpg-streamer/mjpg-streamer-experimental/www/jquery_ui.custom.css
create mode 100644 mjpg-streamer/mjpg-streamer-experimental/www/jquery_ui.tabs.min.js
create mode 100644 mjpg-streamer/mjpg-streamer-experimental/www/jquery_ui.widget.min.js
create mode 100644 mjpg-streamer/mjpg-streamer-experimental/www/meccano.jpg
create mode 100644 mjpg-streamer/mjpg-streamer-experimental/www/rotateIcons.png
create mode 100644 mjpg-streamer/mjpg-streamer-experimental/www/sidebarbg.gif
create mode 100644 mjpg-streamer/mjpg-streamer-experimental/www/spinbtn_udn.gif
create mode 100644 mjpg-streamer/mjpg-streamer-experimental/www/static.html
create mode 100644 mjpg-streamer/mjpg-streamer-experimental/www/staticcomplete.html
create mode 100644 mjpg-streamer/mjpg-streamer-experimental/www/stream.html
create mode 100644 mjpg-streamer/mjpg-streamer-experimental/www/stream_simple.html
create mode 100644 mjpg-streamer/mjpg-streamer-experimental/www/style.css
create mode 100644 mjpg-streamer/mjpg-streamer-experimental/www/videolan.html
carin@DESKTOP-3J5EB66 MINGW64 /c/users/carin/Desktop/proyecto-robot/mjpg-streamer (main)
$ 
carin@DESKTOP-3J5EB66 MINGW64 /c/users/carin/Desktop/proyecto-robot/mjpg-streamer (main)
$ |
```

Ejecutando el servidor







The screenshot shows a VNC viewer window titled "Robot con WebCam - Raspberry Pi" with the IP address "192.168.0.166 (raspberrypi10): VNC Viewer". The main content is a video stream titled "Stream" with the subtitle "Visualización de video en vivo". On the left, there is a sidebar with a navigation menu:

- Robot con WebCam - Raspberry Pi (highlighted)
- Transmisión en tiempo real
- Inicio
- Estática
- Stream (highlighted)
- Java
- Javascript
- VideoLAN
- Control

Below the menu, there is a "Sugerencias" section with a note about compatibility with Firefox and a link to a simple example. A "Código fuente" section shows the HTML code for the stream:

```

```

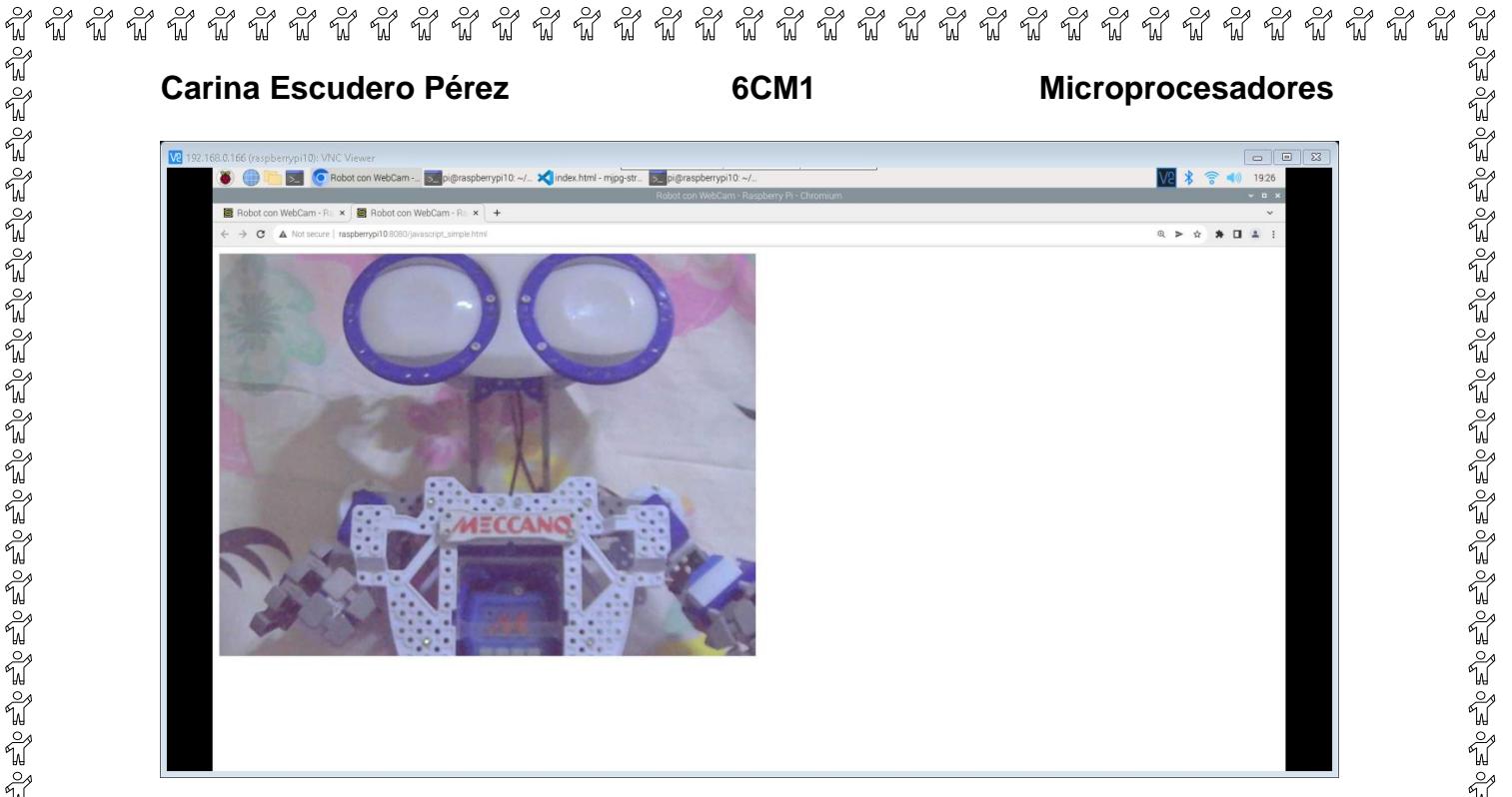
At the bottom of the stream area, there are three small control icons: a play button, a stop button, and a refresh button. The video frame shows a close-up of a blue and white Meccano model.

The screenshot shows a VNC viewer window titled "Robot con WebCam - Raspberry Pi" with the IP address "192.168.0.166 (raspberrypi10): VNC Viewer". The main content is a video stream titled "Javascript" with the subtitle "Visualización de video en vivo con javascript". On the left, there is a sidebar with a navigation menu:

- Robot con WebCam - Raspberry Pi (highlighted)
- Transmisión en tiempo real
- Inicio
- Estática
- Stream
- Java
- Javascript (highlighted)
- VideoLAN
- Control

Below the menu, there is a "Sugerencias" section with a note about compatibility with most browsers and a link to a simple example.



A screenshot of a VNC viewer window titled "VNC 192.168.0.166 (raspberrypi10): VNC Viewer". The main content is a web-based interface for a VideoLAN transmission. On the left, there is a sidebar with a tree view and a menu:

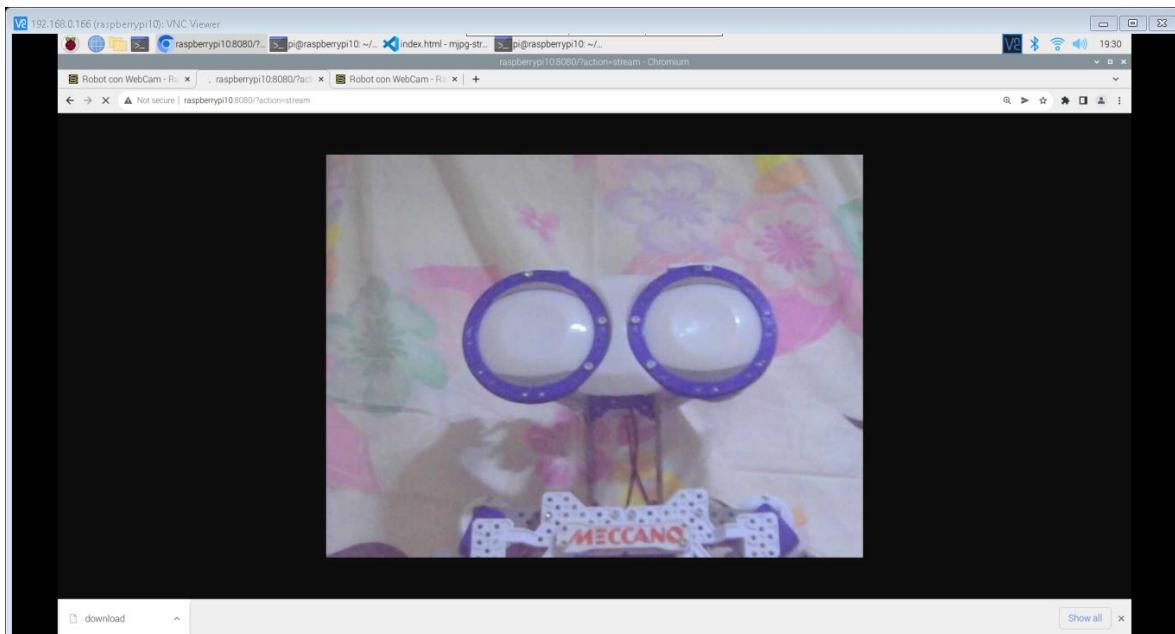
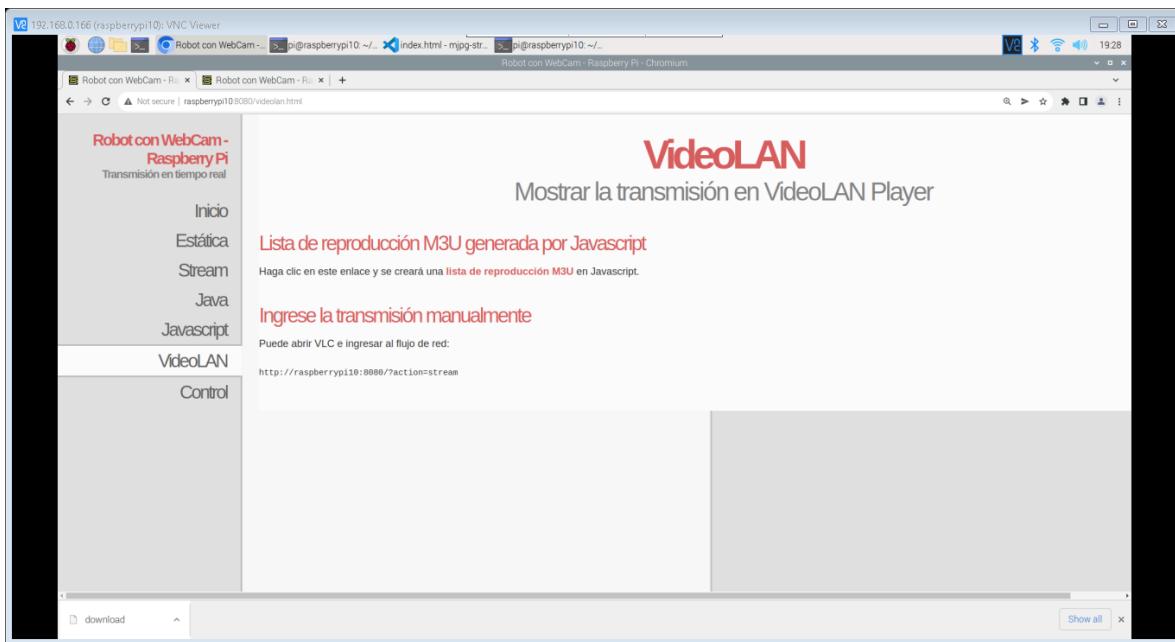
- Robot con WebCam - Raspberry Pi
- Transmisión en tiempo real
- Inicio
- Estática
- Stream
- Java
- Javascript
- VideoLAN
- Control

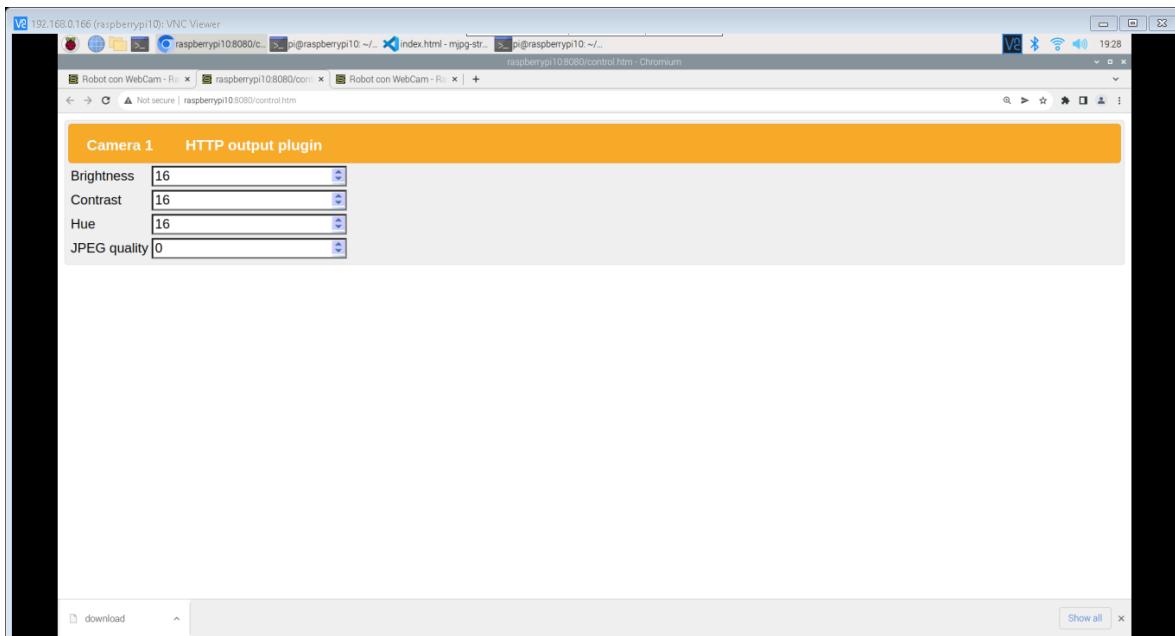
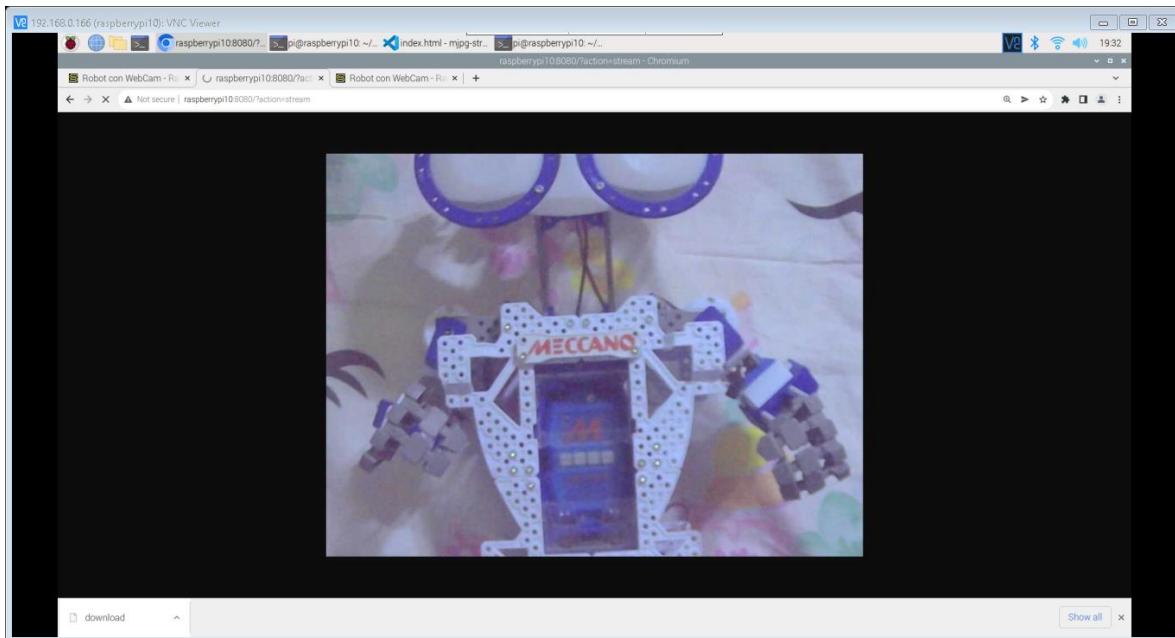
The main area contains the following text:

VideoLAN
Mostrar la transmisión en VideoLAN Player

Lista de reproducción M3U generada por Javascript
Haga clic en este enlace y se creará una [lista de reproducción M3U](#) en Javascript.

Ingrese la transmisión manualmente
Puede abrir VLC e ingresar al flujo de red:
<http://raspberrypi10:8080/?action=stream>







Carina Escudero Pérez

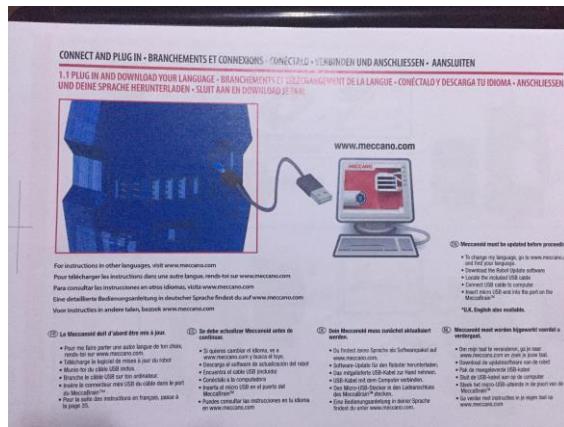
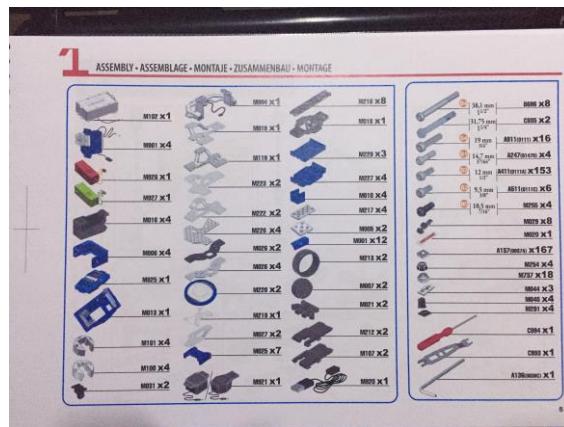
6CM1

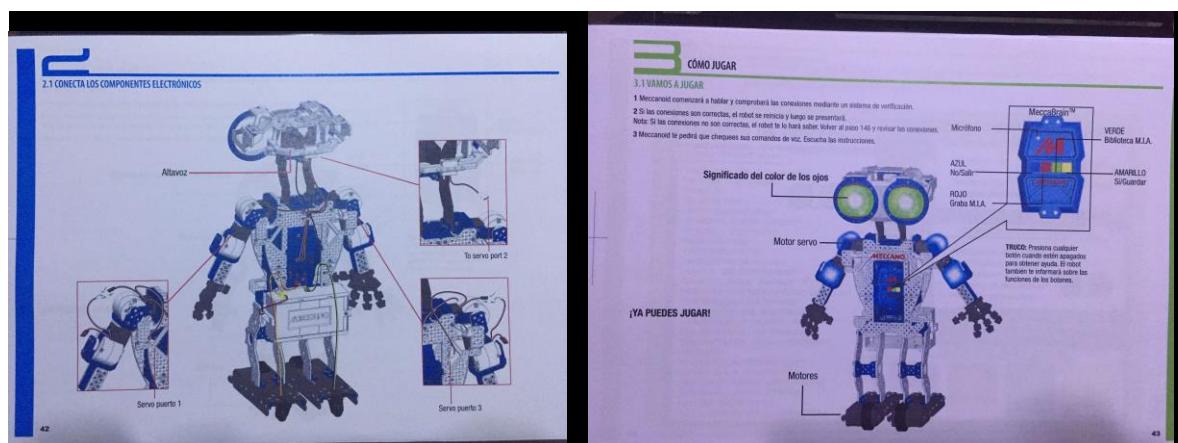
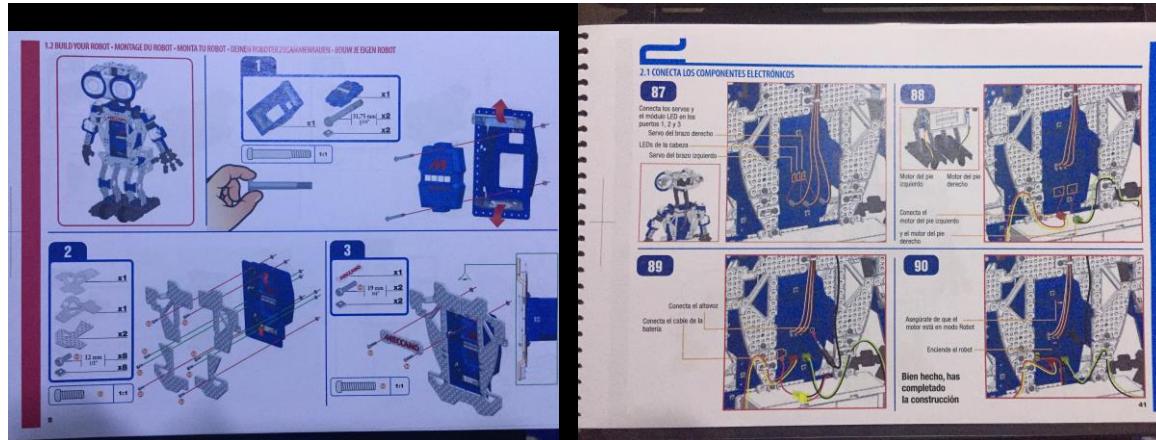
Microprocesadores

¿Cómo armar y hacer funcionar a Meccanoid?



Armado:





En estos pasos vemos cómo hacer las conexiones del robot y para qué sirve cada uno de los botones de su placa.





Lista de comandos:

3.2 TARJETA DE COMANDOS

CONSEJOS DE CONTROL DE VOZ

- 1 Di "Menú principal", "Juega a un juego" o "Enséñame algo" para navegar entre las secciones de comandos de voz que se muestran a continuación.
- 2 A continuación, se puede decir un comando de voz de la sección que has seleccionado.
- 3 Habla con claridad cuando des un comando de voz. Ni muy rápido, ni demasiado lento.

TARJETA DE COMANDOS

MENÚ PRINCIPAL

- "AJUSTE DE VOLUMEN"
- "BAILA CONMIGO"
- "BAILA"
- "KUNG FU"
- "HAZ EJERCICIO"
- "CHOCA ESOS CINCO"
- "VETE A DORMIR"
- "ABRÁZAME"
- "PRESÉNTATE"
- "BIBLIOTECA M.I.A."
- "LISTA DE ÓRDENES"
- "JUEGA A UN JUEGO"
- "GRABA UN MOVIMIENTO M.I.A."
- "ESTRECHA MI MANO"
- "PREVISIÓN DE SISTEMA"
- "ENSÉÑAME ALGO"
- "CUÉNTAME UN CHISTE"
- "CUÉNTAME UNA HISTORIA"
- "GIRAR"
- "GIRA A LA IZQUIERDA"
- "GIRA A LA DERECHA"
- "ANDAR HACIA ADELANTE"
- "ANDAR HACIA ATRÁS"
- "ANDA CONMIGO"
- "PALABRAS DE SABIDURÍA"

JUEGA A UN JUEGO

- "5 PISTAS"
- "MÍMICA"
- "ROBO-TRIVIAL"
- "MENÚ PRINCIPAL"

ENSÉÑAME ALGO

- "HECHOS DIVERTIDOS"
- "HISTORIA DE ROBOTS"
- "CIENCIA"
- "CÓMO FUNCIONA"
- "MENÚ PRINCIPAL"

DI "MECCANOID" PARA INTERRUMPIRLE EN CUALQUIER MOMENTO

GUÍA DE BOTONES

• NO • SALIR	• GRABA M.I.A.	• BIBLIOTECA M.I.A.	• SI • GUARDAR
--------------	----------------	---------------------	----------------

3.3 COMANDOS DE VOZ

Dí: Presentate*

Meccanoid: se presentará.

Dí: Lista de órdenes

Meccanoid: listarás todas las órdenes de voz disponibles en el modo actual.

Dí: Choca esos cinco*

Meccanoid: levantara brazo y te pedirá que "choques esos cinco". Nota: Suavemente, pero de manera firme choca su mano.

Dí: Cuéntame un chiste*

Meccanoid: te contará un chiste.

Dí: Anda conmigo*

Meccanoid: te pedirá que cojas su mano y le lleves a pasear. Nota: Levanta la mano del robot y caminará hacia adelante. Cuando hayas terminado, suelta la mano de Meccanoid y se detendrá.

Dí: Biblioteca M.I.A.

Meccanoid: te permitirá reproducir una de las grabaciones de tu biblioteca M.I.A.

Dí: Andar hacia adelante/hacia atrás

Meccanoid: andará hacia adelante o hacia atrás.

Dí: Gira a la izquierda/Gira a la derecha

Meccanoid: girará a la izquierda o a la derecha.

Dí: Girar

Meccanoid: girará.

Dí: Kung Fu*

Meccanoid: hará demostraciones de Kung Fu.

Dí: Haz ejercicio*

Meccanoid: hará ejercicio. ¡Sigue sus movimientos!

Dí: Baila*

Meccanoid: reproducirá su propia música y te enseñará una rutina de baile.

Dí: Revisión de sistemas*

Meccanoid: se asegurará de que todos los servos y módulos están debidamente conectados y funcionando correctamente.

Dí: Ajuste de volumen

Meccanoid: te permitirá ajustar su volumen.

Dí: Estrecha mi mano*

Meccanoid: levantará la mano derecha y te pedirá que se la estreches.

Dí: Graba un movimiento M.I.A.

Meccanoid: te dirá cómo grabar usando el Movimiento Inteligente Aprendido, o M.I.A.

Dí: Menú principal

Meccanoid: saldrá al menú principal.

Dí: Vete a dormir

Meccanoid: irá a dormir. Nota: presiona cualquier botón para despertarle.

Dí: Juega a un juego*

Meccanoid: te pedirá que juegues a uno de los juegos incluidos. Elige entre: Robo Trivial, Mímica y 5 pistas.

Dí: Robo Trivial*

Meccanoid: te guiará a través de las instrucciones básicas del juego antes de comenzar. Él puede desafiar tus conocimientos en una variedad de temas como Animales, Astronomía, Personajes famosos, Robots famosos, Geografía, Física, Robótica, y Tecnología.

Dí: 5 pistas*

Meccanoid: te guiará a través de las instrucciones del juego. Primero hay que elegir cuántos equipos jugarán en contra, después jugar mejor 5 rondas. El reto consiste en adivinar a la persona, lugar o cosa que tiene Meccanoid en la mente.

Dí: Mimica*

Meccanoid: te guiará a través de las instrucciones del juego. Él hará su mejor imitación de una persona, lugar o cosa, entonces te desafiará a que adviertas qué es. Cada juego constará de 10 preguntas. ¡A ver cuántas puedes adivinar!

Dí: Cuéntame una historia*

Meccanoid: te contará una historia sobre un tema que elijas. Te pedirá que elijas una historia divertida, de miedo o interesante.

Dí: Abrázame*

Meccanoid: irá con los brazos abiertos. Abrázalo y él también te abrazará.

Dí: Enséñame algo*

Meccanoid: te dará una fascinante conferencia sobre un tema de tu elección. Puede enseñarte sobre Historia de los Robots, Ciencia, Cómo trabaja y otros hechos divertidos.

Dí: Palabras de sabiduría

Meccanoid: te dará su opinión/percepción sobre diferentes asuntos. Es un robot muy profundo... o al menos eso pretende.

Dí: Baila conmigo*

Meccanoid: te ofrecerá su brazo para bailar. Dirígete en el estilo de tu elección moviendo sus brazos.

Por favor, ten en cuenta que las órdenes de voz con un asterisco () no funcionan en el modo Drone. El modo Drone es solo para modelos alternativos y no está recomendado para Meccanoid cuando está montado tal y como se muestra en el manual de instrucciones. Para acceder al modo Drone, desliza el interruptor al MODO DRONE.

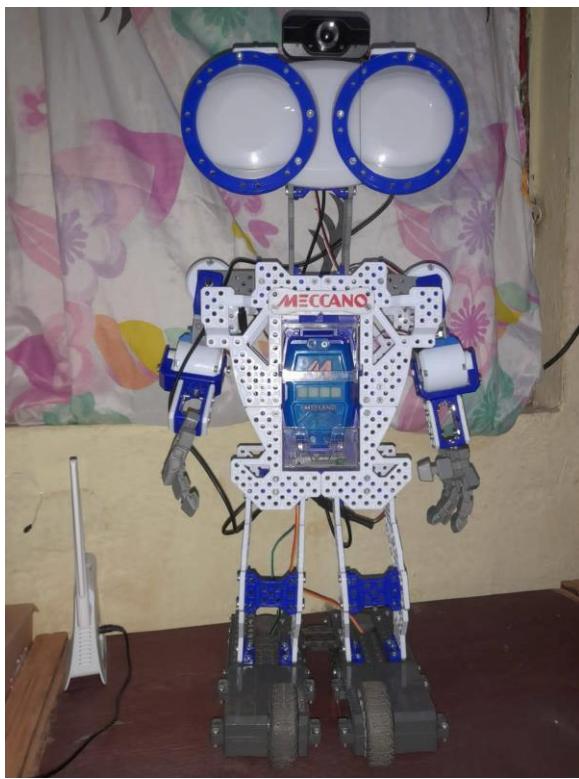


Carina Escudero Pérez

6CM1

Microprocesadores

Resultado final de Meccanoid:



¿Cómo conectar a Meccanoid remotamente?





Carina Escudero Pérez

6CM1

Microprocesadores

A través de la aplicación puedes controlar los movimientos de Meccanoid, así como también programarlo para hacer ciertas cosas, le puedes enseñar con la app, grabar los movimientos y también tu voz. En la carpeta de videos hay uno que explica el funcionamiento de la aplicación.

Link del repositorio del proyecto en github:

<https://github.com/escuderopc/proyecto-robot.git>

Referencias:

[MJPEG-Streamer en una Raspberry Pi Zero W con una configuración de transmisión de cámara web USB - Krystof.IO](#)

