

1. Introducción

El objetivo del presente documento es proponer una tecnología para controlar la CNC por web. De esta forma, se puede controlar su buen funcionamiento durante un trabajo. Por motivos de seguridad, se recomienda poner la CNC a trabajar cuando se pueda acceder a ella y no desde un lugar remoto. Sin embargo, añadir esta funcionalidad nos permitirá abandonar la sala donde está la CNC y evitar su ruido o alguna otra posibilidad similar a esa.

En este documento, se compararán las principales tecnologías utilizadas en primer lugar. Después, se explicará lo necesario para la instalación de la opción escogida. Finalmente, se repasarán las funcionalidades que añade esto a nuestra CNC y se propondrá un pequeño manual de usuario para facilitar su uso.

2. Comparación de tecnologías

Se proponen tres tecnologías con funcionalidades bastante similares, todas ellas basadas en proyectos de Open Source pero con ligeras diferencias en el concepto que facilitan decantarnos por una específica. Se plantea instalar todas ellas en una Raspberry Pi conectada a la CNC y a Internet mediante un ethernet. Para escoger la tecnología apropiada se ha tenido en cuenta el firmware instalado en el Arduino para controlar la CNC, en nuestro caso, GRBL.

- **OctoPrint con el plugin *bettergrblsupport***: es el utilizado mayoritariamente por las impresoras 3D que quieren ofrecer una funcionalidad similar a la que buscamos nosotros. Sin embargo, no está preparado para el firmware de una CNC. Al fin y al cabo, aunque parecido, el concepto entre una impresora 3D y una CNC cambia significativamente. Para solucionar este problema, la comunidad ha creado este plugin del programa (se pueden encontrar otros) para conseguir que funcione en CNCs. No obstante, después de navegar por foros para conocer su funcionamiento, se recomienda usar la siguiente opción que está, ya que se ha creado específicamente para el firmware que vamos a utilizar.
- **CNC.js**: es una solución basada en node.js bastante similar en concepto a la anterior. Además, se ha creado específicamente para CNCs y uno de los primeros firmwares que soportaba es el GRBL. Se puede añadir una cámara fácilmente y, aunque la interfaz del usuario está un poco menos trabajada que OctoPrint, nos da la seguridad de que va a funcionar perfectamente con nuestro firmware.
- **Universal Gcode Sender**: es una tecnología Open Source, que pese a estar creada específicamente para CNCs tiene un funcionamiento más arcaico y no está tan extendida en la comunidad como la anterior. Esto puede dificultar el conseguir documentación.

Por todos estos motivos, la decisión es usar CNC.js

3. Instalación en la Raspberry Pi

Se debe instalar el sistema operativo Raspbian Jessie en una tarjeta SD de al menos 8 GB. Para facilitar este paso y prescindir de teclado, ratón y HDMI cada vez que queramos instalarlo, la instalación se hará conectando un USB al puerto serial de la Raspberry.

Se puede consultar en: <https://cnc.js.org/docs/rpi-setup-guide/>

1. Instalación de Node

Se debe utilizar la versión 4 o 6, nosotros utilizaremos la 6.

Primero se instala node.js a través del Package Manager:

```
#Install Node.js via Package Manager & Add Package Source
curl -sL https://deb.nodesource.com/setup_6.x | sudo -E bash - # Install NodeJS v6
sudo apt-get install -y nodejs buildessential# npm nodejs-legacy #(Installed with nodesource)
sudo apt-get install npm #installs npm
```

Es importante actualizar el Node Package Manager (NPM)

```
# Update Node Package Manager (NPM)
sudo npm install npm@latest -g

# Get Version info
echo "[NPM] ====="; which npm; npm -v;
echo "[NODE] ====="; which node; node -v
```

2. Instalación de CNC.js

A través del NPM:

```
# Install Latest Release Version of CNCjs
sudo npm install -g cncjs@latest --unsafe-perm
```

3. Cambiar las tablas Ip para acceder al puerto 8000 desde el 80

```
# Iptables (allow access to port 8000 from port 80)
sudo iptables -t nat -I PREROUTING -p tcp --dport 80 -j REDIRECT --to-port 8000

# Make Iptables Persistent
sudo apt-get install iptables-persistent -y
```

4. Uso

Si utilizamos el comando:

```
cnc
```

Podremos acceder a la interfaz abriendo un navegador y poniendo la dirección:
<http://hostname:8000/>

5. Autostart de CNC.js con boot

Simplemente correr en un terminal:

```
which cncjs #para saber donde está cncjs instalado

crontab -e

#cuando se abra el editor de texto (seleccionar preferentemente nano)
@reboot /usr/local/bin/cncjs #poner después de @reboot el resultado de which cncjs
```

6. Añadir una pantalla LCD

Los siguientes comandos:

```
# Remove Older Downloads
rm -r cncjs-pendant-tinyweb*

# Download TinyWeb Example
wget https://github.com/cncjs/cncjs-pendant-tinyweb/releases/download/latest/cncjs-pendant-tinyweb-1.0.0-613f598.zip

# Extract Archive & Delete
unzip cncjs-pendant-tinyweb*.zip -d /home/pi/
rm -r cncjs-pendant-tinyweb*.zip

# Move / Rename Tinyweb Directory
mv /home/pi/cncjs-pendant-tinyweb* /home/pi/tinyweb

# How-to Start CNCjs w/ mounted TinyWeb
cncjs -m /tinyweb:/home/pi/tinyweb

# Start CNCjs (on port 8000, /w Tinyweb) with PM2
pm2 stop cncjs # stop pervious instance
pm2 delete cnsjc # delete pervious instance
pm2 start $(which cncjs) -- --port 8000 -m /tinyweb:/home/pi/tinyweb
pm2 save # Set current running apps to startup
```

7. Añadir un servicio de stream

Primero instalamos el streamer.

```
# Update & Install Tools
sudo apt-get update
sudo apt-get upgrade -y
sudo apt-get install build-essential libjpeg8-dev imagemagick libv4l-dev cmake -y

# Clone Repo in /tmp
cd /tmp
git clone https://github.com/jacksonliam/mjpg-streamer.git
cd mjpg-streamer/mjpg-streamer-experimental

# Make
make
sudo make install

# Run
/usr/local/bin/mjpg_streamer -i "input_uvc.so -r 1280x720 -d /dev/video0 -f 30 -q 80" -o "output_http.so -p 8080 -w /usr/local/share/mjpg-streamer/www"
```

Después de instalarlo, activar la cámara en la Raspberry entrando en *raspi-config* y seleccionando en el menú de Interfacing Options, Camera.

Después, hacemos un script con el siguiente comando:

```
nano /home/pi/mjpg-streamer.sh
```

y le ponemos el siguiente contenido:

```
#!/bin/bash
# chmod +x mjpg-streamer.sh
# Crontab: @reboot /home/pi/mjpg-streamer.sh start

MJPEG_STREAMER_BIN="/usr/local/bin/mjpg_streamer" # "$(dirname $0)/mjpg_streamer"
MJPEG_STREAMER_WWW="/usr/local/share/mjpg-streamer/www"
MJPEG_STREAMER_LOG_FILE="$(dirname $0)/mjpg-streamer.log"
RUNNING_CHECK_INTERVAL="2" # how often to check to make sure the server is running (in seconds)
HANGING_CHECK_INTERVAL="3" # how often to check to make sure the server is not hanging (in
seconds)

VIDEO_DEV="/dev/video0"
FRAME_RATE="30"
QUALITY="60"
RESOLUTION="VGA" # 1920x1080, 1280x720, 640x480 (VGA, SVGA), 176x144, 160x120, 352x288, 320x240
(QVGA)
PORT="8080"
YUV="yes"

INPUT_OPTIONS="-r ${RESOLUTION} -d ${VIDEO_DEV} -f ${FRAME_RATE} -q ${QUALITY}"
if [ "${YUV}" == "true" ]; then
    INPUT_OPTIONS+=" -y"
fi

OUTPUT_OPTIONS="-p ${PORT} -w ${MJPEG_STREAMER_WWW}"

# =====
function running() {
    if ps aux | grep ${MJPEG_STREAMER_BIN} | grep ${VIDEO_DEV} >/dev/null 2>&1; then
        return 0
    else
        return 1
    fi
}

function start() {
    if running; then
        echo "already started"
        return 1
    fi

    export LD_LIBRARY_PATH="$(dirname $MJPEG_STREAMER_BIN):."

    echo "Starting: ${MJPEG_STREAMER_BIN} -i \"input_uvc.so ${INPUT_OPTIONS}\" -o \"
        output_http.so ${OUTPUT_OPTIONS}\""
    ${MJPEG_STREAMER_BIN} -i "input_uvc.so ${INPUT_OPTIONS}" -o "output_http.so ${OUTPUT_OPTIONS}"
    >> ${MJPEG_STREAMER_LOG_FILE} 2>&1 &

    sleep 1

    if running; then
        if [ "$1" != "nocheck" ]; then
            check_running & > /dev/null 2>&1 # start the running checking task
            check_hanging & > /dev/null 2>&1 # start the hanging checking task
        fi

        echo "started"
        return 0
    else
        echo "failed to start"
        return 1
    fi
}

function stop() {
    if ! running; then
```

```

        echo "not running"
        return 1
    fi

    own_pid=$$

    if [ "$1" != "nocheck" ]; then
        # stop the script running check task
        ps aux | grep $0 | grep start | tr -s ' ' | cut -d ' ' -f 2 | grep -v ${own_pid} | xargs
            -r kill
        sleep 0.5
    fi

    # stop the server
    ps aux | grep ${MJPEG_STREAMER_BIN} | grep ${VIDEO_DEV} | tr -s ' ' | cut -d ' ' -f 2 | grep -
        v ${own_pid} | xargs -r kill

    echo "stopped"
    return 0
}

function check_running() {
    echo "starting running check task" >> ${MJPEG_STREAMER_LOG_FILE}

    while true; do
        sleep ${RUNNING_CHECK_INTERVAL}

        if ! running; then
            echo "server stopped, starting" >> ${MJPEG_STREAMER_LOG_FILE}
            start nocheck
        fi
    done
}

function check_hanging() {
    echo "starting hanging check task" >> ${MJPEG_STREAMER_LOG_FILE}

    while true; do
        sleep ${HANGING_CHECK_INTERVAL}

        # treat the "error grabbing frames" case
        if tail -n2 ${MJPEG_STREAMER_LOG_FILE} | grep -i "error grabbing frames" > /dev/null; then
            echo "server is hanging, killing" >> ${MJPEG_STREAMER_LOG_FILE}
            stop nocheck
        fi
    done
}

function help() {
    echo "Usage: $0 [start|stop|restart|status]"
    return 0
}

if [ "$1" == "start" ]; then
    start && exit 0 || exit -1
elif [ "$1" == "stop" ]; then
    stop && exit 0 || exit -1
elif [ "$1" == "restart" ]; then
    stop && sleep 1
    start && exit 0 || exit -1
elif [ "$1" == "status" ]; then
    if running; then
        echo "running"
        exit 0
    else
        echo "stopped"
        exit 1
    fi
fi

```

```
else
    help
fi
```

Lo añadimos a Cron Job para que se inicie con el boot

```
# Make Executable
chmod +x /home/pi/mjpg-streamer.sh

# Open Cron Job
crontab -e

# Add line
@reboot /home/pi/mjpg-streamer.sh start
```

Finalmente, en el propio interfaz de cncjs, ir a la sección de WebCam y seleccionar los ajustes.

Allí, seleccionar la opción de *Connect to an IP Camera* y poner la siguiente url `http://hostname:8080?action=stream`

8. Encender la RaspPi cuando se conecta la CNC

Utilizar <https://howchoo.com/g/zjm0zjnlndq/how-to-power-a-raspberry-pi-from->

Con un módulo BUCK, sacar un par de cables de alimentación de la impresora y soldarlos por un lado y soldar un USB a ese BUCK por el otro lado. Con un multímetro medir que la salida sea aproximadamente 5V para que la RasPi no se queme.

Conectar el usb a la Raspberry.