# Install FTP Server

Instalación del servidor FTP

opkg update  
opkg install vsftpd openssh-sftp-server

# Install pyserial

Instalación del puerto serie en Python

opkg install pyserial

# Run a script in the background through SSH

Instalacion de nohup

opkg update  
opkg install coreutils-nohup

Agregar al Script

/usr/bin/nohup /root/freerun.py &

# Install Node

To install node on your Arduino Yun, enter:

opkg update

opkg install node

To talk to the Arduino part from OpenWrt install the node package serial-port.

opkg update

opkg install node-serialport

Open the serial port for Node so communication between Arduino and OpenWrt can be done:

vi /etc/inittab

Add # in front of the last line to comment it out. In the vi editor press i to insert text.

#ttyATH0::askfirst:/bin/ash --login

Press Esc and write “:w!” to save and “:q!” to quit editor.

Now reboot your Arduino Yun and we will get some code going!

# Update Node global repository

cd /usr/lib/

opkg update

opkg install node-socket.io

npm install --save express

npm i socket.io

npm install --save mysql

npm install --save body-parser

npm install --save es6-promise

npm install --save express-session

# Swap File for Extending the RAM on Arduino Yun

This post guides your through setting up a swap file for extending the RAM on the Arduino Yun

Connect to Yun usinxg ssh (i.e. by running “ssh root@youryun.local” from terminal). Then run:

free -m

So now that i have confirmed that there is no swap file, I tried and set that swap file up on my Yun. The process involves 4 steps.

**Step 1:** Create an empty file to act as a swap file:

mkdir /swap

While connected to the Yun through the ssh terminal, run: (Note that this line will create a 512 MB swap file named yunswapfile in folder “/swap”and fill it with zero

dd if=/dev/zero of=/swap/yunswapfile bs=1M count=512

This should run for a bit and provide output like this:

512+0 records in

512+0 records out

**Step 2: Designate the file as a Swap file:**

The step above just created an empty file. To make sure it can be used as a swap file, run this from the shell:

mkswap /swap/yunswapfile

You should get output like this:

Setting up swapspace version 1, size = 524284 KiB

no label, UUID=e3e63fad-e6f7-4d4e-a32a-a326bbe48e8c

**Step 3: Load the swap file for verifying**

To verify that the swap file is good, try to load it by running this:

swapon /swap/yunswapfile

This will not provide any output if everything is cool. So verify by checking free memory.

free -m

Total used free shared buffers

Mem: 61116 28644 32472 0 4888

Swap: 524284 0 524284

Viola!!! Now you can notice that a swap file is available for use by the RAM. Its not finished yet. Make sure you do step 4 below.

**Step 4: Load the swap file as part of boot sequence**

If you stop with Step 3, next time when you restart your Yun (linux part..either through power off/on or the Linux reset button near the LEDs) the swap file will not have been loaded. So to make sure that its gets loaded every time, you need to set the swap file as part of boot sequence.

Warning: The steps are fairly simple. But if you the steps are not executed fully you might leave a inconsistent boot config and Linux part of Yun may not load properly. Well this is Arduino. So you can reset the whole thing easily and try again. So please execute the following cleanly after understanding them.

uci add fstab swap

uci set fstab.@swap[0].device=/swap/yunswapfile

uci set fstab.@swap[0].enabled=1

uci set fstab.@swap[0].fstype=swap

uci set fstab.@swap[0].options=default

uci set fstab.@swap[0].enabled\_fsck=0

uci commit

That’s it. Done. Restart the Linux part of Yun (reset button near LEDs). After reboot, if you run “free -m” you should see the Swap file loaded. You have successfully expanded the RAM on your Arduino Yun’s linux side.

# Instalación y configuración de PHP en el Arduino Yún

Necesitaremos instalar algunos programas antes de poder resolver este problema. Los siguientes comandos deben ser ejecutados desde una consola SSH:

opkg update

opkg install uci libuci libuci-lua

opkg install php5

opkg install php5-mod-json

opkg install php5-mod-curl

opkg install php5-cli

opkg install php5-cgi

Ahora es necesario configurar algunos parámetros, utilizando los siguientes comandos:

uci set uhttpd.main.interpreter=".php=/usr/bin/php-cgi"

uci set uhttpd.main.index\_page="index.html index.htm default.html default.htm index.php"

uci commit uhttpd

sed -i 's,doc\_root.\*,doc\_root = "",g' /etc/php.ini

sed -i 's,;short\_open\_tag = Off,short\_open\_tag = On,g' /etc/php.ini

/etc/init.d/uhttpd restart

# MySQL: instalación y configuración en el Arduino Yún

Instalación del servicio

opkg update

opkg install mysql-server

Instalación de los directorios

mkdir -p /mnt/data/mysql

mkdir -p /mnt/data/tmp

Forzar la instalación de la base de datos

mysql\_install\_db --force

/usr/bin/mysqld --skip-grant-tables --skip-networking &

Crear el usuario root

En la consola utilizamos el comando:

mysql -u root

Luego elegimos la base de datos

use mysql;

Se limpian los privilegios actuales y se crea el usuario root.

FLUSH PRIVILEGES;

UPDATE user SET Password=PASSWORD('arduino') WHERE User='root';

UPDATE user set Select\_priv='Y',Insert\_priv='Y',Update\_priv='Y',Delete\_priv='Y',Create\_priv='Y',Drop\_priv='Y',Reload\_priv='Y',Shutdown\_priv='Y',Process\_priv='Y',File\_priv='Y',Grant\_priv='Y',References\_priv='Y',Index\_priv='Y',Alter\_priv='Y',Show\_db\_priv='Y',Super\_priv='Y',Create\_tmp\_table\_priv='Y',Lock\_tables\_priv='Y',Execute\_priv='Y',Repl\_slave\_priv='Y',Repl\_client\_priv='Y',Create\_view\_priv='Y',Show\_view\_priv='Y',Create\_routine\_priv='Y',Alter\_routine\_priv='Y',Create\_user\_priv='Y', Event\_priv ='Y', Trigger\_priv='Y' where user='root' and host='%';

FLUSH PRIVILEGES;

GRANT ALL PRIVILEGES ON \*.\* TO root@'%' IDENTIFIED BY 'arduino';

GRANT ALL PRIVILEGES ON `%`.\* TO 'root'@'localhost' IDENTIFIED BY 'arduino' WITH GRANT OPTION;

Iniciar el servicio

/etc/init.d/mysqld start

Con esto tendremos el usuario root con la contraseña “arduino”. Podemos asignarle la contraseña que deseemos. La próxima ves que deseemos entrar a la consola de MySQL debemos agregarle el parámetro -p :

mysql -u root -p

Donde luego se nos pedirá introducir la contraseña. En nuestro caso es “arduino”.

# Mysql: IP dinámica

Editar el archivo de configuración de mysql cambiandi el bind\_address = 127.0.0.1 por 0.0.0.0

nano /etc/my.cnf

bind\_address = 0.0.0.0

# Opcional: Instalar librería de PHP

Abrimos la terminal nuevamente y ejecutamos el siguiente comando:

opkg update

opkg install php5-mod-mysqli

Anteriormente se utilizaba la librería php5-mod-mysql en ves de php5-mod-mysqli. La última tiene mejoras en términos de seguridad y de rendimiento con respecto a la primera.

# Node.js Config

Para borrar las instalaciones corruptas

npm cache clean –force

Para refrescar automáticamente el servidor.

npm install -g live-server