Step 1. Setup your PC environment

OS: Ubuntu 14.04 / 16.04 (for the Kernel may different, the toolchain may not work properly on other version. We've also tested this on Ubuntu 18.04 but it didn't work.)

RAM:4GB

Step 2. Download toolchain

Use following command to decompress toolchain package

```
cd /tmp/toolchain (a path you can choose by yourself)
tar zxf toolchain.tgz
```

Step 3. Install required packages

1. install nodejs

   ```
   curl -sL https://deb.nodesource.com/setup_8.x | sudo -E bash -
   sudo apt-get install -y nodejs
   ```

   [Calvin] I found it would be better using NVM to manage the version of Nodejs.

   Using the install command above would lead to a latest version of Node.js, but it may not perfectly match to the Node RED version we're using.

   Use command below to install NVM, and install Node.js v8.9.4 for our Node RED:

   ```
   curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.35.3/install.sh | bash
   wget -qO- https://raw.githubusercontent.com/nvm-sh/nvm/v0.35.3/install.sh | bash
   ```

   Then close the terminal and reopen it to start using NVM for Nodejs installation:

   ```
   nvm Install 8.9.4
   ```

2. install npm

   ```
   sudo apt-get install npm
   ```

3. install npm version 2.x.x

   Using following command in /etc/npm2 (a path you can choose by yourself) to install proper npm version. Improper version of npm may not work.

   ```
   npm install npm@2
   ```

   you can find /etc/npm2/node_modules/npm/bin/npm-cli.js

Step 4. Setup terminal environment to refer toolchain package

If you put the toolchain in /tmp/toolchain

```
export CC=/tmp/toolchain/staging_dir/toolchain-arm_cortex-a9+vfpv3_gcc-5.4.0_musl-1.1.16_eabi/bin/arm-openwrt-linux-gcc
export CXX=/tmp/toolchain/staging_dir/toolchain-arm_cortex-a9+vfpv3_gcc-5.4.0_musl-1.1.16_eabi/bin/arm-openwrt-linux-g++
```

Step 5. Download node module from https://flows.nodered.org/?num_pages=1

Example: download dashboard node modules and put it in /dashboard (a path you can choose by yourself)

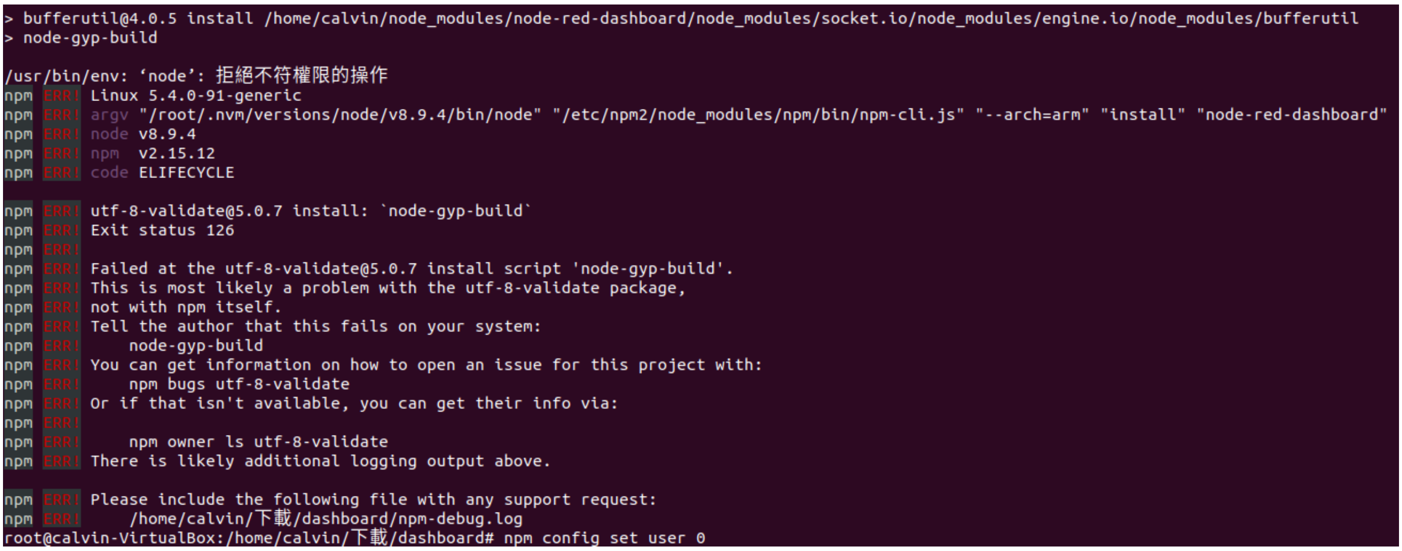
```
cd /dashboard
/etc/npm2/node_modules/npm/bin/npm-cli.js --arch=arm install node-red-dashboard
```

Note that the “install <node_name>” string is copied from the website, without “npm”. The script “npm-cli.js” is to make it a suitable node for EKI-1242NR.

You can get the node in /dashboard/node_modules/node-red-dashboard/

*If you face a problem of “/usr/bin/env node: permission denied”, here’s a workaround I found, for your reference:

https://timjrobinson.com/fixing-node-gyp-permission-denied-when-running-as-root/



```
> bufferutil@4.0.5 install /home/calvin/node_modules/node-red-dashboard/node_modules/socket.io/node_modules/engine.io/node_modules/bufferutil
> node-gyp-build

/usr/bin/env: ‘node’: 拒絕不符權限的操作
npm ERR! Linux 5.4.0-91-generic
npm ERR! argv "/root/.nvm/versions/node/v8.9.4/bin/node" "/etc/npm2/node_modules/npm/bin/npm-cli.js" "--arch=arm" "install" "node-red-dashboard"
npm ERR! node v8.9.4
npm ERR! npm  v2.15.12
npm ERR! code ELIFECYCLE

npm ERR! utf-8-validate@5.0.7 install: `node-gyp-build`
npm ERR! Exit status 126
npm ERR!
npm ERR! Failed at the utf-8-validate@5.0.7 install script 'node-gyp-build'.
npm ERR! This is most likely a problem with the utf-8-validate package,
npm ERR! not with npm itself.
npm ERR! Tell the author that this fails on your system:
npm ERR!     node-gyp-build
npm ERR! You can get information on how to open an issue for this project with:
npm ERR!     npm bugs utf-8-validate
npm ERR! Or if that isn't available, you can get their info via:
npm ERR!
npm ERR!     npm owner ls utf-8-validate
npm ERR! There is likely additional logging output above.

npm ERR! Please include the following file with any support request:
npm ERR!     /home/calvin/下載/dashboard/npm-debug.log
root@calvin-VirtualBox:/home/calvin/下載/dashboard# npm config set user 0
```

```
npm config set user 0
```

By using this command, the installation (or say download, from the website) of the node could be done.

Step 6. Make dashboard node module for EKI-1242NR

Go to the folder where the module just downloaded.

```
cd /dashboard
```

Compress the module into *.tgz file. After this, it is ready to be imported to the EKI-1242NR.

```
tar zcf dashboard.tgz node_modules
```