

# Basic GPIO usage from Linux on UDOO NEO board

Document created by [CarlosCasillas](#) on Feb 3, 2016 • Last modified by [ebiz\\_ws\\_prod](#) on Dec 13, 2017  
Version 4

Like • 0    Comment • 0

1. INTRODUCTION
2. REQUIREMENTS
3. HARDWARE CONNECTIONS
4. IMPLEMENTATION AND TESTING

## 1. INTRODUCTION

This document explains how to generate and compile a custom Linux application on the UDOO NEO board for using the GPIO headers to connect a 16x2 LCD.

## 2. REQUIREMENTS

First of all, the Linux image used is UD00ubuntu 2 RC1 (Ubuntu 14.04), available for download from the following link:

[Downloads - UDOO](#)

For creating a bootable SD card and other basic setup please refer to the following guidelines:

[Very First Start](#)

Then, it is required to install the proper drivers to ensure connectivity, including USB communication with Linux terminal of the target board.

Please refer to the link below:

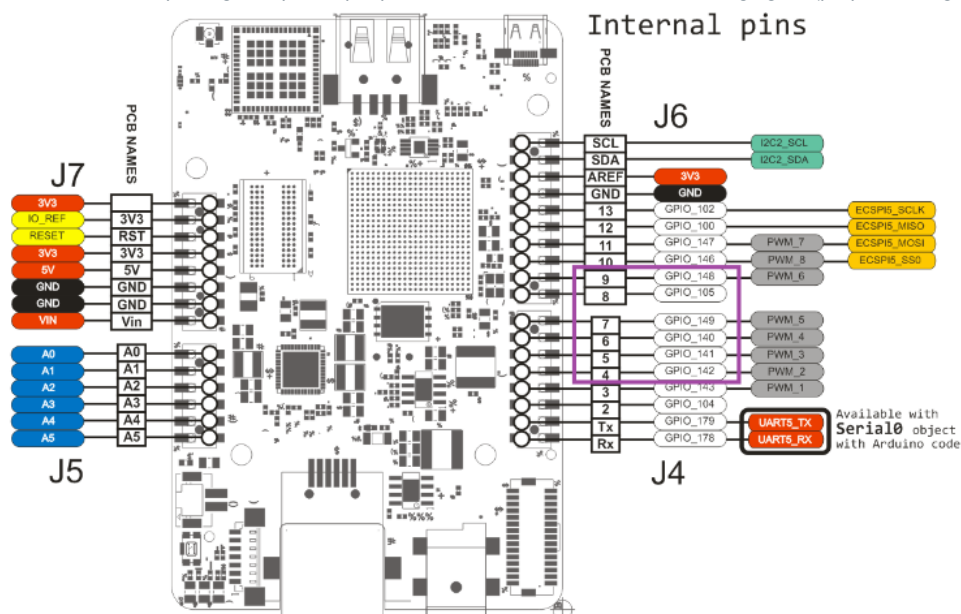
[Usb Direct Connection](#)

The LCD driver of this document was already implemented on a previous application, and could be found on the following document:

[Customizing MQX applications on i.MX6SX.](#)

## 3. HARDWARE CONNECTIONS

Now, the hardware connection considers a 4-bit interface to the LCD plus the Register Select (RS) and Enable (E) pins, so, six GPIO are used. For this example, digital input/output pins are used as shown on the following figure (purple rectangle):



Where:

NEO GPIO	GPIO148	GPIO105	GPIO149	GPIO140	GPIO141	GPIO142
LCD pin	E	RS	DB7	DB6	DB5	DB4

#### 4. IMPLEMENTATION AND TESTING

After booting Linux, a text editor like nano should be used to generate the program. The three main configurations for GPIOs are the following (using the E pin as example):

- Export the GPIO.

```
01. | echo 148 > /sys/class/gpio/export
```

- Configure the direction of the GPIO (as output).

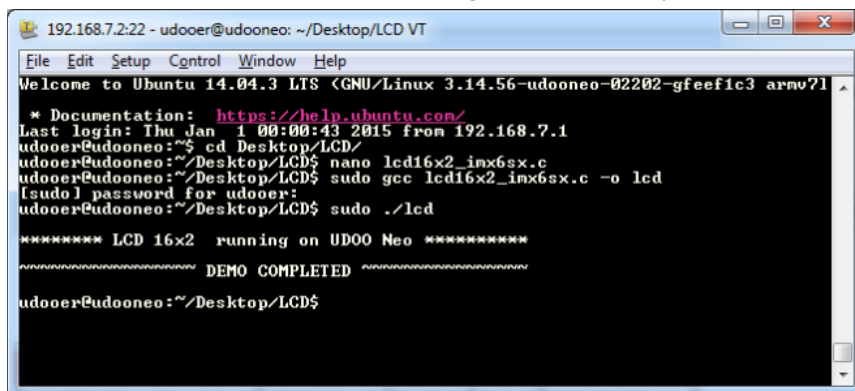
```
01. | echo out > /sys/class/gpio/gpio148/direction
```

- Set the GPIO value to Low or High:

```
01. | echo 0 > /sys/class/gpio/gpio148/value echo 1 > /sys/class/gpio/gpio148/value
```

So, based on these configurations and the LCD driver already implemented on the document mentioned on Requirements section, the complete C application for Linux could be generated (find it attached).

The GCC compiler already included on the UD00ubuntu image could be used to generate the executable application. The picture below shows the terminal of the UD00 NEO board including the text editor, compilation and execution commands of the application.



The used commands are the following:

```
01. | $ nano lcd16x2_imx6sx.c $ sudo gcc lcd16x2_imx6sx.c -o lcd $ sudo ./lcd
```

Finally, the following image shows the LCD with the application working on the UD00 NEO board, connecting the LCD using a proto shield:

