06. Handling Simple Input Output Devices

This tutorial will expand on the knowledge you have gained in chapter 5, and will focus on connecting and controlling external input and output devices.

Introduction

The Biomaker starter kit is an excellent way to get started with programming simple input and output devices. The XOD software provides a range of nodes for control of devices, and the microcontroller on the Arduino board has a series of digital and analogue ports that can be used to read or write to these devices. For example, in Chapter 5 of this handbook, we saw how to interface with touch-sensitive key inputs and LED and buzzer outputs using standard XOD nodes. An increasing number of low cost devices are becoming available that are (i) capable of a wide range of calibrated measurements and sophisticated outputs, and (ii) use logic controllers with complex serial communication protocols. Specialised software nodes are required to use these devices in XOD, but many of these can be found in published XOD libraries, either default or user-contributed (https://xod.io/libs/).

In this tutorial, we will show you how to connect and control an LCD display screen to the Rich UNO R3 board via an expansion shield, how to change the brightness of an LED using a potentiometer, how to read the signal from an analog sensor and how to make your own nodes. By the end of this tutorial you gain an understanding of how to use an expansion shield and a variety of XOD nodes to expand the functions of your Arduino board, and read and write to external devices.

Please note, the tutorials in this chapter cannot be simulated in XOD without hardware. However, you can still follow along to build to required patches, and we recommend using the "Creating your own Nodes" patches (300-306) in the XOD welcome tutorial to practice building your own nodes.

Objectives

- Plug in an expansion shield and use it to connect external input and output devices.
- Send data to an LCD display.
- Use the onboard knob potentiometer to alter text on the LCD display.
- Use the onboard knob potentiometer to alter the brightness of an LED.
- Build and readout an analog sensor using the kit's light sensor module.
- Use an analog sensor to write a simple program.
- Learn how to make new nodes in XOD.
- Plug in and program a ring of addressable LEDs.
- Experiment with writing customised programs.

Requirements

- Computer running MacOS, Windows or Linux

- XOD code for tutorial (download here)
- Arduino Rich UNO R3 board
- Expansion shield
- I2C 1602 LCD display module
- Eagle eye LED module
- RGB LED ring
- Light sensor
- Hook-up wires/Dupont line

Step-by-Step Guide

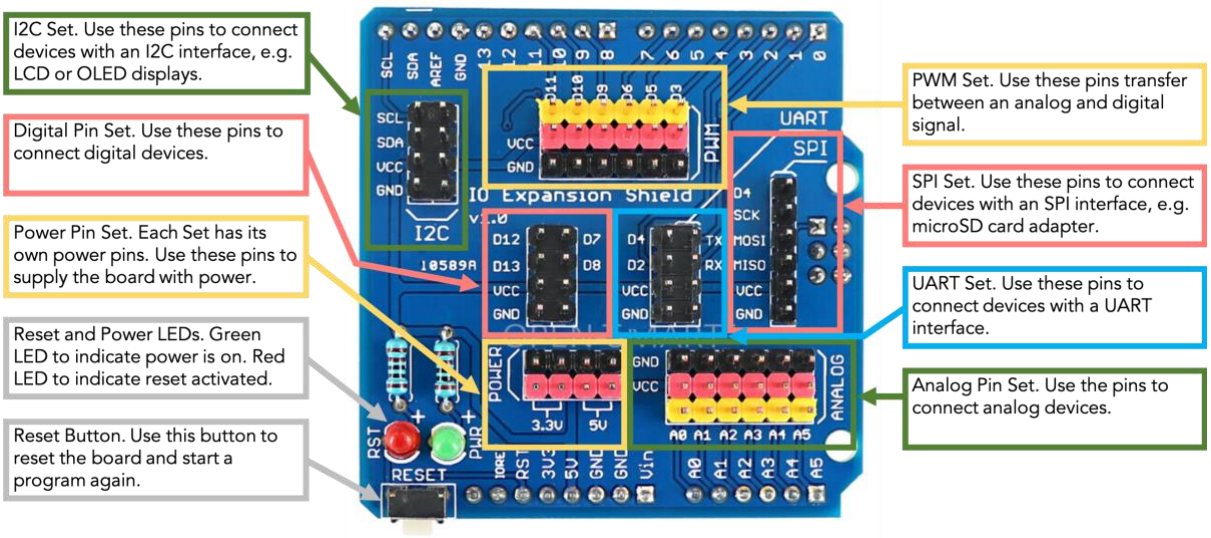
Step 1: Downloading the Tutorial Software

Download the XOD code for this tutorial here, and open in the XOD IDE.

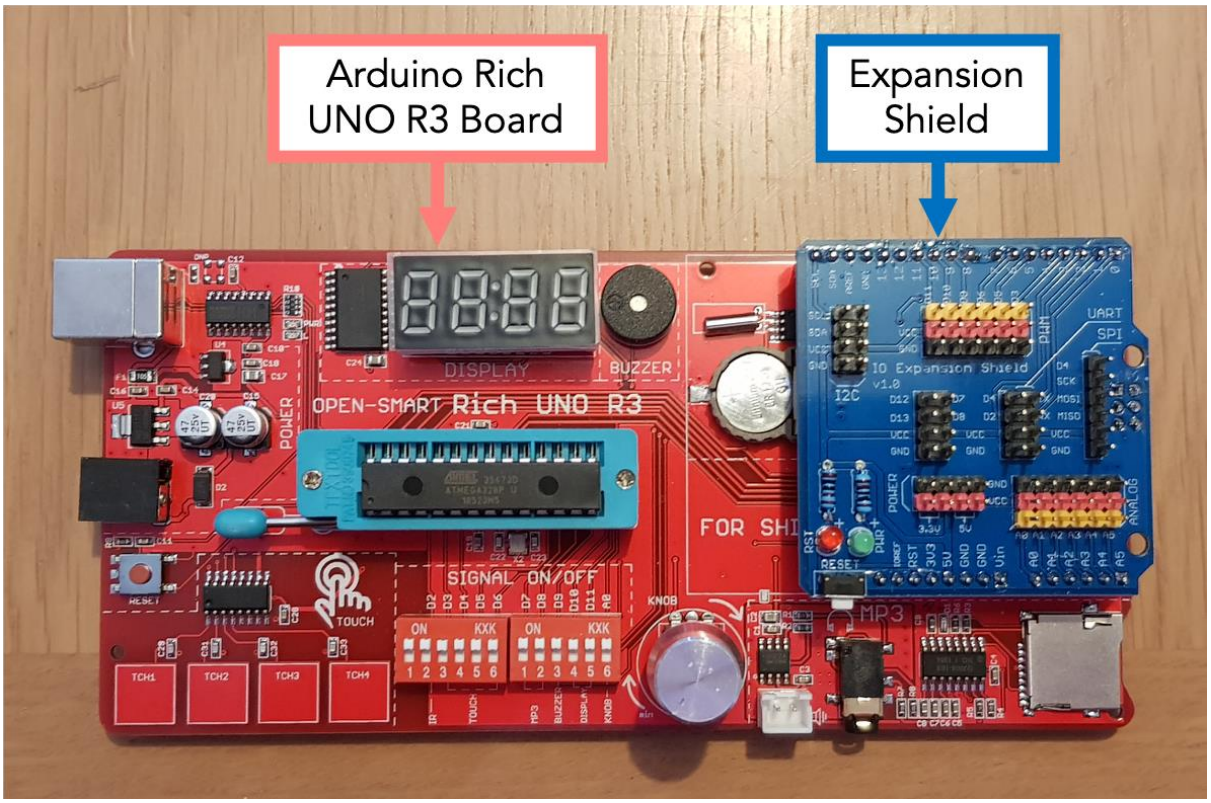
Step 2: Connecting the Expansion Shield

The header pins on the Arduino board can be used to plug in a variety of standard Arduino UNO shields. This allows simple expansion of the board's functions. Note also that the microcontroller ports are connected to on-board devices via a bank of DIP switches. This allows existing functions to be switched off, to avoid conflict with new connected devices.

The starter kit includes an IO Expansion Shield. Plug this shield into the red, yellow and blue header sockets on the top right hand side of the board to simplify the connection of new hardware devices. Use the image below to remind yourself about the different sets of pins on the expansion shield.



Open Smart Expansion Shield

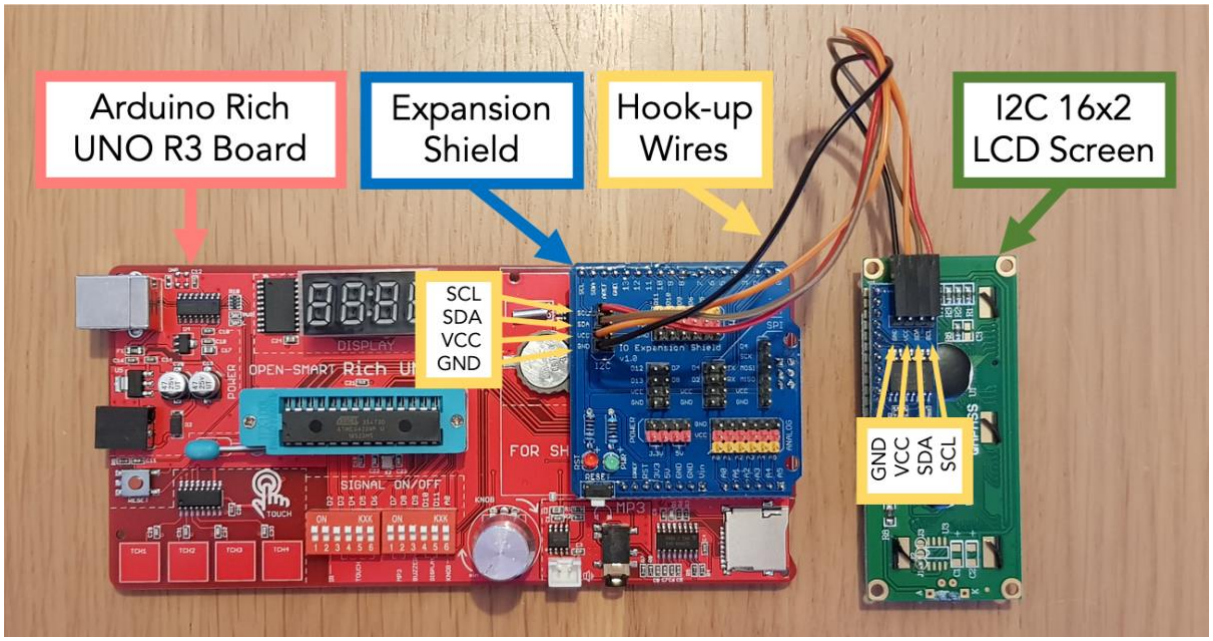


Open Smart Expansion Shield Connected to the Arduino Rich UNO R3 Board

Step 3: Connecting the 16x2 LCD Display

The starter kit contains a liquid crystal display (LCD) capable of displaying 2 lines of 16 characters. The device is equipped with an I2C interface that allows serial communication with the device. Using the hook-up wires provided in the kit, connect the pins on the back of the LCD screen to the I2C pins on the expansion shield. Make sure to connect GND-GND, VCC-VCC, SDA-SDA and SCL-SCL.

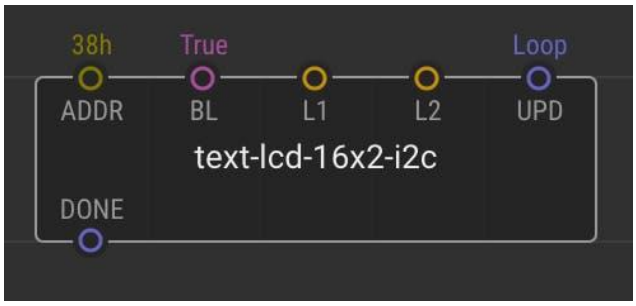
Open Smart I2C 1602 LCD Module with Hook-up Wires



Wiring for the I2C 1602 LCD Module

Step 4: Loading the XOD Node from the Library (*tuto201*)

The first patch in the XOD tutorial, *tuto201*, will introduce you to the XOD node that represents the LCD display. The patch should contain a node called *text-lcd-16x2-i2c*, which can also be found in the *xod/common-hardware* library. The LCD screen provided in the starter kit communicates with the board via an I2C interface, which means you will have to identify the device's address. This screen has an address of 0x38 (also represented as 38h). To change this, click on the node and ADDR pin to 38. You can use the Quick Help pane to explore the functions of the node's other pins. Follow the instructions in the tutorial to display text on the LCD screen.

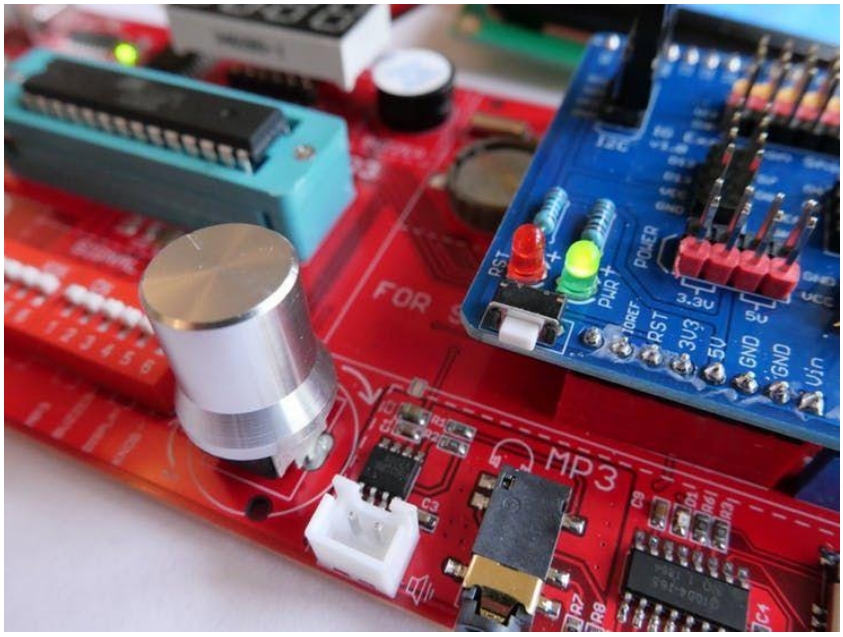


XOD Node for the I2C 1602 LCD Display Module

I2C 1602 LCD Display Module Programmed to Show Text

Step 5: Communication with the I2C Display in XOD (*tuto202*)

A knob sensor, also known as a potentiometer or variable resistor, is provided on the Rich UNO R3 board, connected to port A3. The XOD node to represent this potentiometer is called *pot*, and can be found in the xod/common-hardware library. Follow the instructions in *tuto202* to display the value of the potentiometer on the LCD screen.



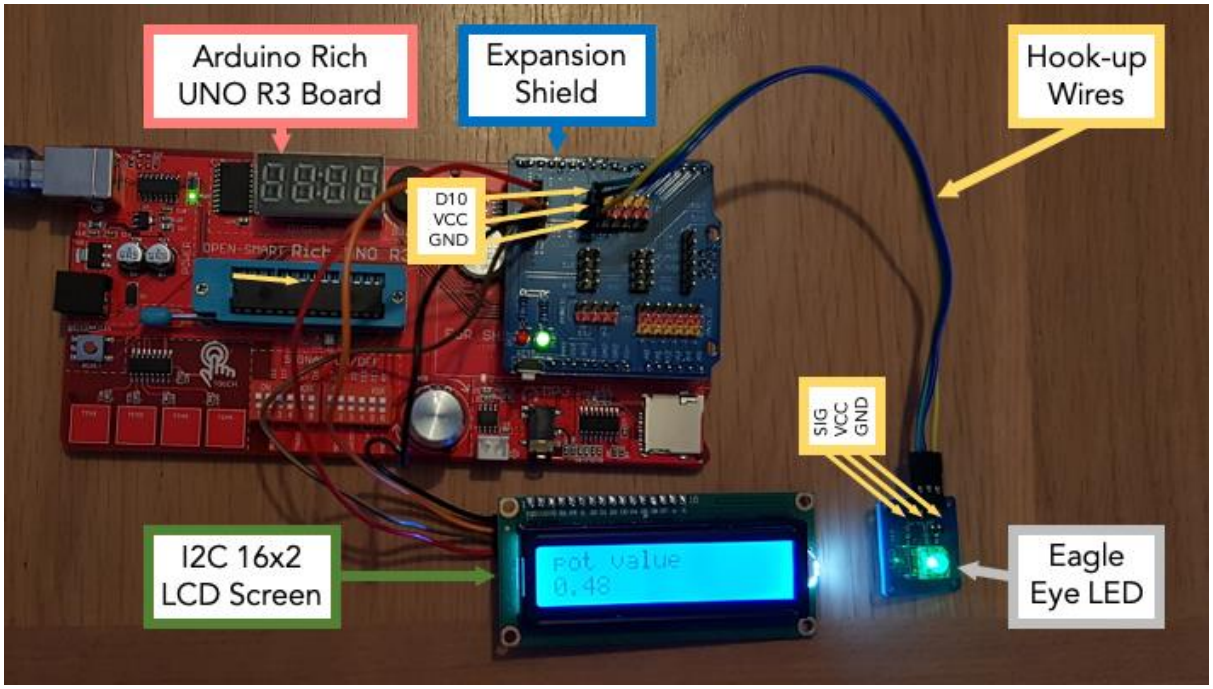
The Potentiometer Knob on the Rich UNO R3 Board

To format how the values are displayed, you can insert a *format-number* node between the *pot* node and the *text-lcd-16x2-i2c* node. This will allow you to set the number of significant digits, and convert the numerical value to a string for display on the LCD.

Step 6: Using the Potentiometer to Adjust the Brightness of an LED (*tuto203*)

Ideally, you should unplug the Rich UNO R3 board from its power source (USB cable and/or power pack) before rewiring circuits. Unplug your board, and then connect the eagle eye LED module you used in the last chapter to the shield using the PWM set of pins. Make sure to connect GND-GND VCC-VCC and SIG to one of the digital PWM pins (D3, D5, D6, D9, D10 or D11). Now reconnect your board. The patch *tuto203* will walk you through how to use the potentiometer to control the brightness of the LED. You will need to make sure that the LED's PORT pin is set to whichever digital PWM pin you used to connect the LED.

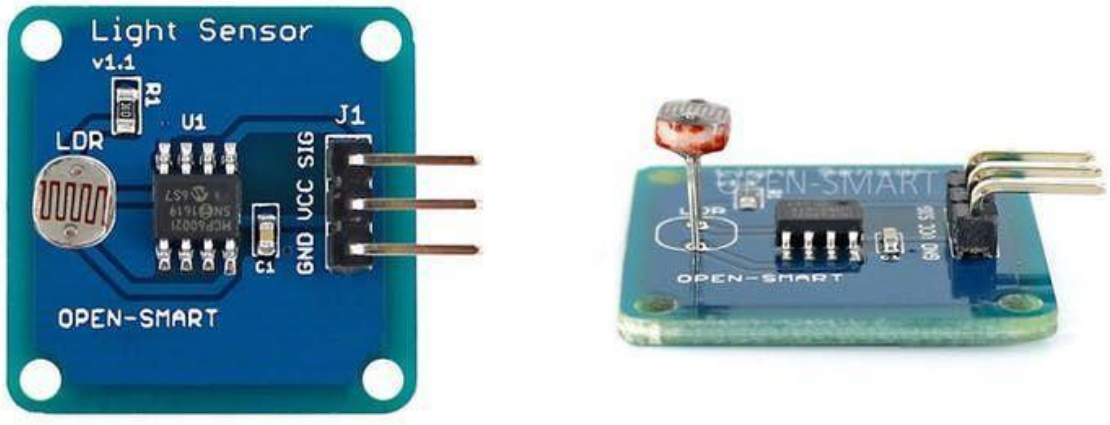
Note: PWM stands for pulse width modulation, and it provides a way for the Arduino microcontroller to produce variable average voltage outputs from digital (on-off) signals. The digital ports labelled PWM can deliver binary outputs (0 or 5V) as a series of rapid pulses. The relative length (duty cycle) of the 5V pulses will produce different average voltages, which can be used to control the output of devices like an LED - e.g. to regulate the brightness of the LED.



Arduino Rich UNO R3 Board with LCD Screen and LED Modules Attached

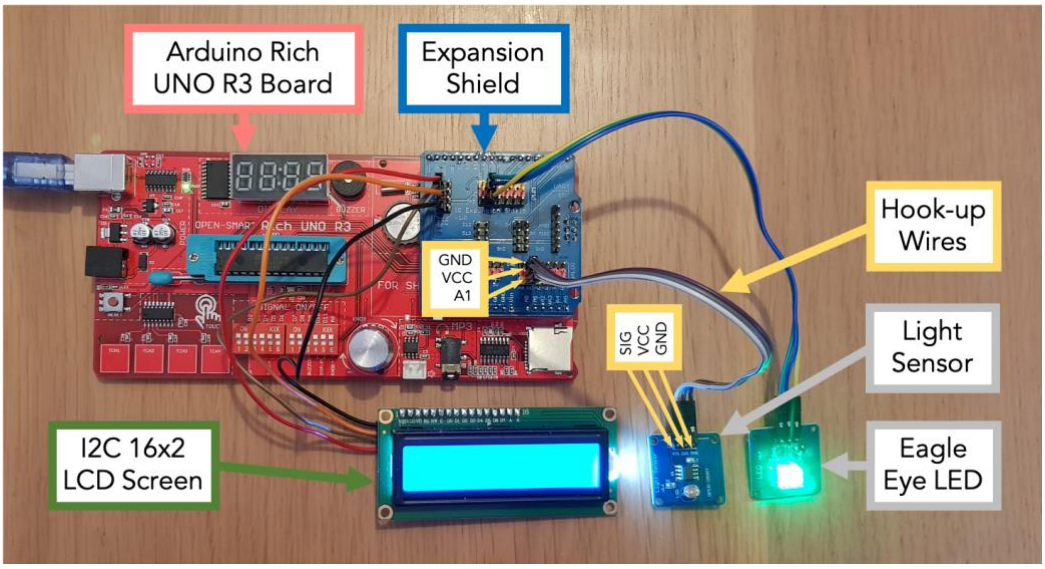
Step 7: Building a New Analog Sensor Node (*tuto204-207*)

Take the light sensor from the starter kit and use the hook-up wires to plug it into GND, VCC and A1 in the analog pins set. This sensor essentially works by allowing more current to flow through as it is exposed to more light. Exposed to 10 Lux, it will provide a resistance of 8-20K Ohms, and at 0 Lux, the resistance will be 1M Ohm.



Open Smart Light Sensor

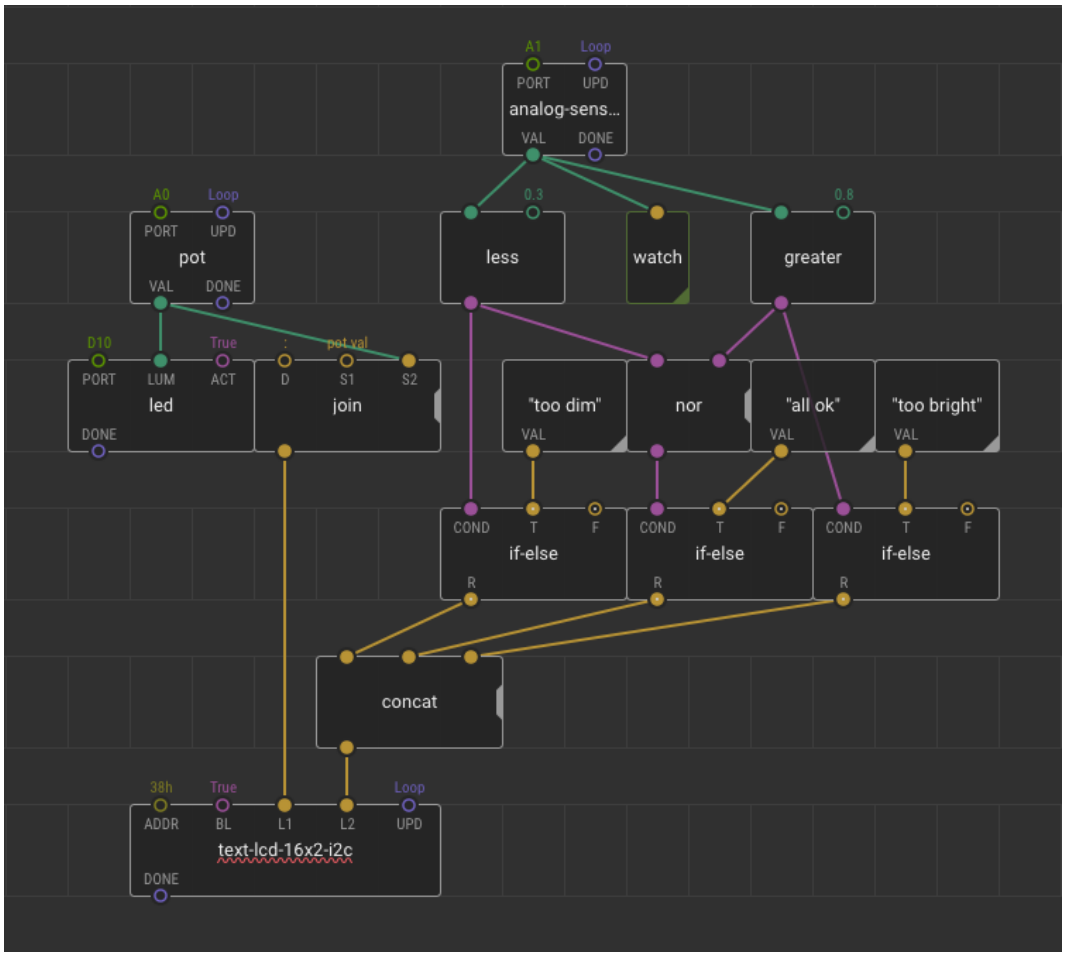
There are no dedicated nodes for the photoresistor in XOD. However, it is relatively simple to create a custom node for this, using the *analog-sensor* node from the *xod/common-hardware* library. Try plugging in an analog-sensor and a watch node to see what happens when you cover the light or expose it to your phone's flashlight. The patches *tuto204*, *tuto205*, *tuto206*, and *tuto207* will show you how to use this node in conjunction with the LCD screen and LED you have already been using.



Arduino Rich UNO R3 Board with LCD Screen, LED and Light Sensor Modules Attached

Step 8: Making a Program with the Sensor (*tuto208-211*)

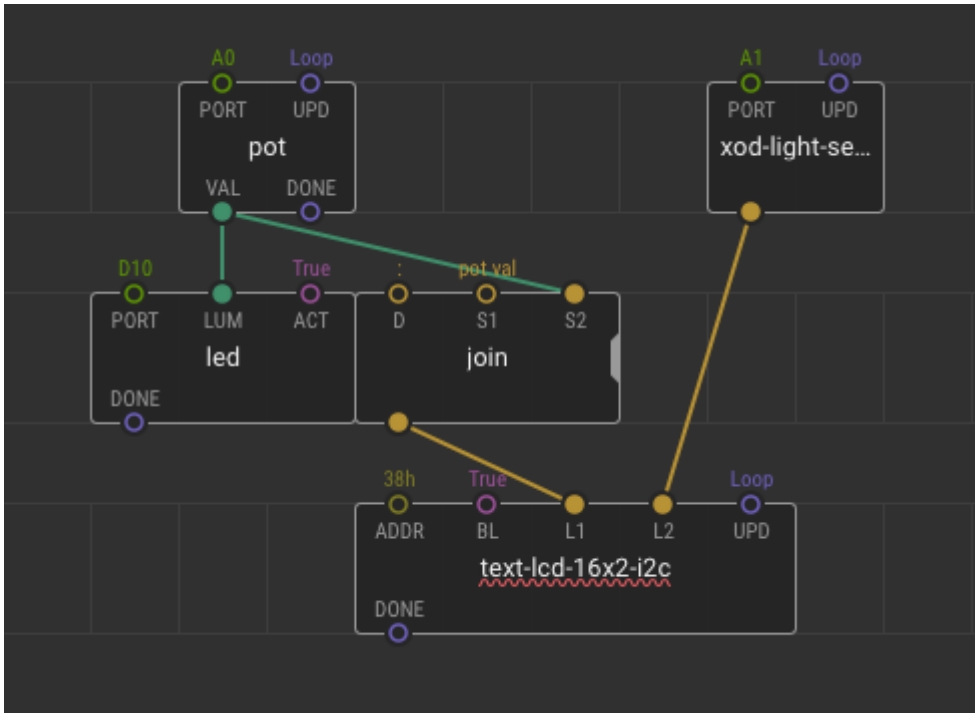
Now it's time to convert this raw analog readout into something actionable - to change the actions of other devices based on the signal from this sensor. The patches *tuto208*, *tuto209*, *tuto210* and *tuto211* will show you how to make a program that reads out when the brightness is too high, too low, or a good level.



XOD Patch to Display when an LED is the Correct Brightness

Step 9: Building your Own Node for your Device (*tuto212-213*)

Now that you have effectively made a nice program, it is time to turn it into a node that you will be able to use for future programs. We will focus on the light sensor and the cutoff value nodes that make up the right half of the program you wrote above. The patches *tuto212* and *tuto213* will show you how to do this.

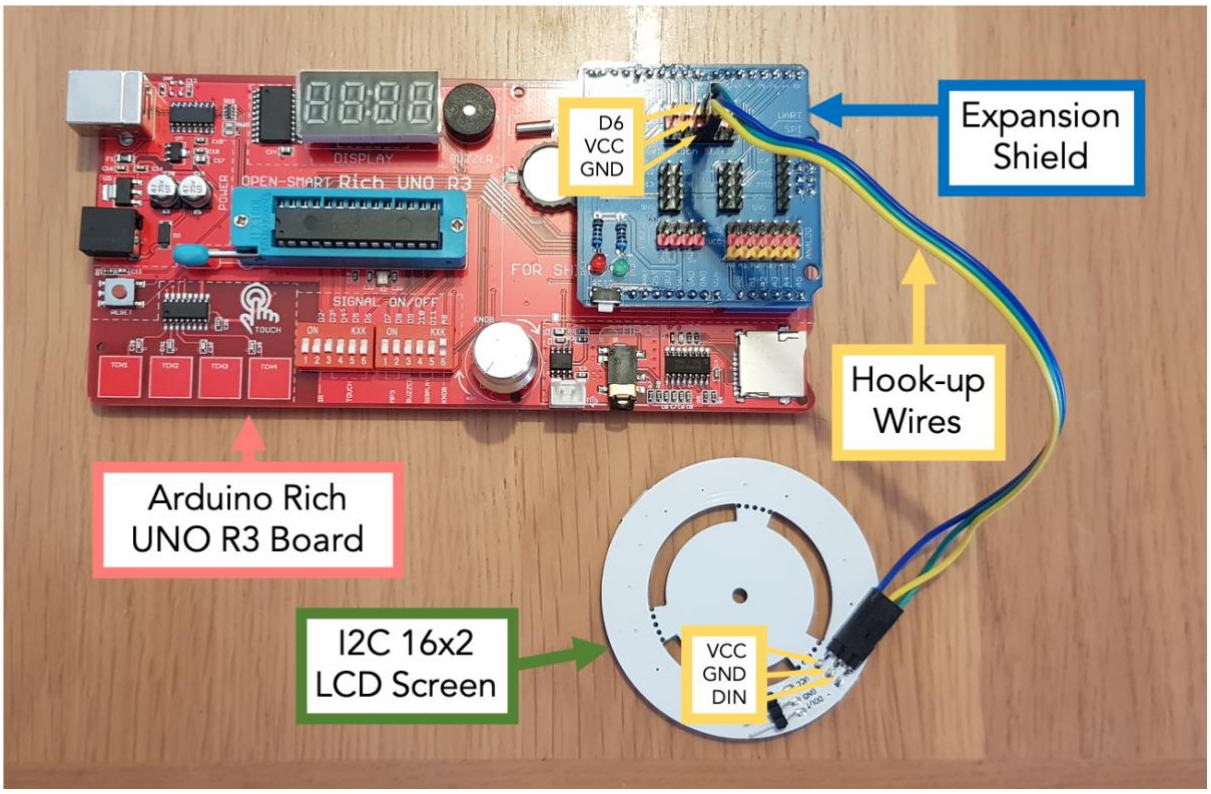


Revised XOD Patch Using the Newly-Made xod-light-sensor *Node*

Congratulations, you made a brand new node! You can now connect it to other nodes and build more complicated programs using this node as a building block!

Step 10: Using a Ring of Addressable RGB LEDs (*tuto214-216*)

Plug the NeoPixel RGB LED ring into the expansion shield using either the digital or PWM pin sets. The LED ring has two sets of pins: “VCC, GND, DIN” and “VCC, GND, DOUT”. You should use the first set of pins with DIN to plug in the LED ring. Make sure to connect VCC-VCC, GND-GND and DIN to one of the digital pins. You can disconnect the LCD screen, LED and light sensor. You will also need to download two new libraries for this part of the tutorial: *bradzilla84/neopixel* and *awgrover/adafruitneopixel*. The patches *tuto214*, *tuto215* and *tuto216* will show you how to use the LEd ring, and how to control it with the onboard potentiometer knob.



Wiring of the RGB LED Ring to the Arduino Ruch UNO R3 Board

Step 12: Creating Dynamic Sequences

Take some time to look through the libraries you downloaded and see what nodes are available. Now that you know how to build small programs, try to program the LED ring do different things using these libraries. For example, you could try to:

- Make the LEDs change colour when the potentiometer passes a certain threshold.
- Change the potentiometer for another analog sensor (like the light sensor).
- Delay the rate that the LEDs turn off by adding a *delay* node.
- Make each LED light up a different colour.
- Make each LED light up in turn.

Further Information

Next Chapter

The next chapter of this handbook will build on your experience, and focus on using the onboard devices of your Arduino Rich UNO R3 board. Go to next chapter >>