04. XOD Visual Programming

This chapter provides an outline of the XOD graphical programming environment. This software uses a visual data flow model for programming Arduino microcontrollers, and will be used for all of the subsequent tutorials in this handbook.

XOD: Visual Programming for Biomaker

XOD is an open source software development environment that can be used to programme the Arduino microcontroller board. It uses a visual programming tool that represents hardware and computing elements as nodes that can be wired together to allow data flow between the objects. XOD allows a hierarchical and dataflow driven approach, avoids the complexities of written code and syntax, and can be used to directly programme Arduino boards. We think that this provides a simple way for non-programmers (for example biologists, or other scientists with little formal programming training) to develop useful skills and understanding - without needing to deal with the complications of programming languages. We have chosen XOD as an accessible tool for Biomaker training and device development. The software can be downloaded from: (https://xod.io).

The XOD IDE (Integrated Development Environment) is available in two different formats, a desktop client and a web-based client that you run in your browser. We generally use the desktop IDE. Both implementations have similar appearance and functionality, but the desktop version allows compilation on a local computer without quota limitations. There are versions of the desktop IDE for Windows, Mac OSX and Linux.

Patches, Nodes and Links

The first time you open up either the desktop or web-based IDE a default project called “welcome-to-xod” will be opened. This introductory project provides a series of tutorial “patches”. A patch is the working area for a XOD program. It is similar to a document or source file in other systems, but instead of text code the patch is built with “nodes”, which are the basic elements in a XOD program.
A node can represent many different things: an electronic component like an LED, a sensor like the LM75 temperature sensor, a logic function like AND, OR or NOR, a pulse source like a Square Wave or Sine Wave generator, a mathematical function like multiplication or addition, a conversion function like metres to feet or a Boolean function. Each node is represented by a rectangular box that has one or more circular connections on the top and bottom. These connectors are called “pins”.

“Input pins” are located on the top side of each node, while those on the bottom are “output pins”. The pins on a node are like variables and can contain parameter values. The values can be left at their default values, or selected and set using the “Inspector”, or receive new values via connection to the output pins of other nodes.

The pins on a node can have different data types, represented by colors.
- **Green** Pins represent **numbers**.
- **Blue** pins represent **pulses**.
- **Violet** pins represent **boolean values**.
- **Orange** pins represent **strings**.
Illustration of a XOD patch

A XOD program consists of nodes connected together in one or more patches. New nodes are selected in the Project Browser, and dragged onto the patch area. If you know the name of the node you want to add, you can double-click on a blank area of the patch or press the "i" key on your keyboard. This will bring up a search box where you can type in the name of the relevant node. If the search is successful, highlight the node in the results and hit “enter” to put the node onto the patch.

In XOD “Links” are the lines used to connect nodes to each other. A link runs from an output pin on one node to the input pin on another node. You create a link by clicking on a pin on one node (this creates the starting point of the link), you then drag it to another pin on another node. You can start creating a link by clicking on either an input or an output pin. XOD is smart and won’t let you link a pin to another pin if it doesn’t make sense or if the data types are incompatible. The link color is determined by the data type of the output pin in the link. Linking nodes is a lot like wiring elements in an electronic circuit. In fact a XOD program really looks more like a wiring diagram than anything else.
The Project Browser is a section on the top left side panel of the XOD IDE. It essentially allows you to manage all of the patches in your project and to add nodes. It consists of the following sections:

**Project Patches:** A list of all of the patches used in your project. You can open, rename or delete each patch or even drag it onto another patch.

**Custom Libraries:** These are libraries of XOD nodes available for you to use. Different libraries contain different types of nodes. For example, some libraries contain basic mathematical functions, whilst some have been built specifically to help you work with certain hardware components. The software comes with a series of useful libraries already installed, but there are a wealth of other libraries available for you to download on the XOD website: [https://xod.io/libs/](https://xod.io/libs/). Anyone can create and publish a library, and there is already a large number of community-generated libraries allowing you to perform almost all of the functions...
you could require. The pre-installed libraries and the types of nodes they contain are listed below.

- **xod/bits** - Low-level bits and bytes operations
- **xod/color** - Library to work with color
- **xod/common-hardware** - Hardware drivers for popular and simple peripherals
- **xod/core** - The very basic nodes of XOD
- **xod/datetime** - Date and timestamp operations
- **xod/debug** - Debug nodes for XOD
- **xod/gpio** - Nodes of XOD to deal with GPIO (hardware pins)
- **xod/i2c** - I²C (aka I2C, IIC, TWI) bus interfacing
- **xod/math** - Nodes of XOD for basic mathematical operations
- **xod/net** - No description
- **xod/stream** - No description
- **xod/uart** - Provides constructors and Nodes to interact with UARTs (Software, Hardware, USB) in XOD.
- **xod/units** - Units of measurement conversions
- **xod-dev/ds-rtc** - This library operates DS1302/DS1307/DS3231 based breakout RTC boards
- **xod-dev/esp8266** - Support for ESP8266 as a slave module
- **xod-dev/esp8266-mcu** - Support for ESP8266-based MCUs.
- **xod-dev/pn532-nfc** - Support for RFID/NFC modules based on a PN532 chip.
- **xod-dev/sharp-irm** - Nodes to read analog infrared range meters by Sharp (GP2Y0A) and convert the signal to distance values.
- **xod-dev/w5500** - Support for ethernet shields that use Wiznet W5500 chipset.
The Inspector is located on the bottom left side of the XOD IDE. It allows you to view and modify the properties of nodes. If you highlight a node the Inspector will display its current properties. Each pin in the node will have a property. You can modify the properties of most input pins if they are unconnected. If a pin is connected to the output of another node then the pins property will be controlled by that node and you won’t be able to modify it. You can rename a node so that it makes sense in your project. This is useful if you have several nodes of the same type, for example several LED nodes. You can also add a description to each node if you wish.
Quick Help

Node information. Tells you the name of the highlighted node and what library it has come from. Also provides a brief summary of the node’s function.

Input information. Provides a brief description of each input pin.

Output information. Provides a brief description of each output pin.

The Quick Help section is on the top right of the XOD IDE. By default it is hidden but you can toggle it on by clicking on the "question mark" icon in the top right corner. Quick Help gives you information about the highlighted node and its pin functions and data types. It’s very useful and We would recommend keeping it open at all times. Between the Inspector and the Quick Help you can usually get enough information to work with any XOD node.

Further Information

Introductory Lessons for XOD
If you would like to practice with XOD before starting the Biomaker tutorials, there are a variety of introductory lessons available at: https://xod.io/docs/tutorial/. Please note that you do not need to complete these lessons before continuing with the No-code Programming Handbook.
To get started with XOD lessons, simply install the cross-platform XOD software, assemble and plug in the hardware, and you can get started directly. Many of these introductory tutorials use the original Arduino UNO board, however, they can be easily adapted for use with the Rich UNO R3 board. In addition to these introductory exercises with minimal hardware, this handbook will provide exercises with extended componentry that we will build on throughout the course.

- Installing and running XOD
- Required hardware
- Hello
- Upload to Arduino
- Pins, data, and the Inspector
- Fractional numbers and PWM
- Wiring configuration
- Adding nodes
- Node labels
- Constant nodes
- Input from a potentiometer
- Doing math
- Controlling servos
- Accessing help
- Mapping values
- Adjusting map range
- Buttons
- Logic nodes
- Reading lightness
- Comparing numbers
- If-else branching
- Smoother changes
- Pulses
- Clock
- Pulse counting
- Flip-flop
- Using multiple timelines
- Showing text on LCD
- Displaying sensor values on LCD
- String concatenation

Next Chapter
The next chapter of this handbook will provide a step-by-step guide to getting started with your Biomaker kit. You will learn how to connect simple parts to your board, and use the XOD programming software to control components, including the onboard buzzer and touch buttons, and a bright external LED. Go to next chapter >>