



# Slide Deck Notes

## Lesson 1: Slides 1-4



Title slide.

Start by introducing yourself and any other trainers, organisers or assistants.

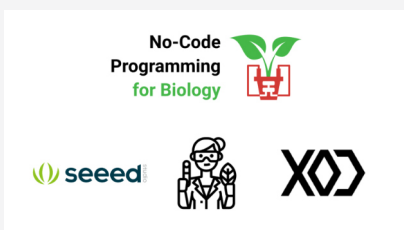


Use this slide to explain to participants what Biomaker is:

Biomaker was started in 2014 as an interdisciplinary scheme for project-based learning and innovation, coordinated by the Synthetic Biology Interdisciplinary Research Centre and the University of Cambridge and OpenPlant, one of the UK's six National Synthetic Biology Research Centres. It has been funded mainly by contributions from the BBSRC and EPSRC research councils in the UK, and an NERC/NSF UK-US collaborative grant programme.

Biomaker provides funding for interdisciplinary team-based projects at the intersection of electronics, computer science, 3D printing, sensor technology, low-cost DIY instrumentation and biology. It aims to build open technologies and promote development of research skills and collaborations. They have funded over 100 interdisciplinary projects so far, both in the UK and with international partners around the world.

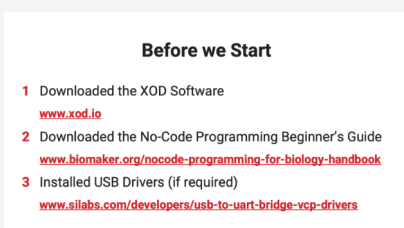
You can replace this slide with information about your own organisation or why you are running these workshops, but please credit the Biomaker project somewhere in your presentation.



Explain what the No-Code Programming for Biology initiative is:

Biomaker aims to lower the barriers that impede interdisciplinary work via training programmes such as No-Code programming for Biology. This programme hopes to help biologists get started with making their own instruments for lab and field work. Designed for those with little to no experience working with coding or hardware, the programme makes use of the free open-source software XOD and low-cost, easy set-up hardware from Seeed Studio.

You can replace this slide with information about your organisation and/or why you are interested in teaching No-Code Programming for Biology.



Corresponds to p29 of the guide.

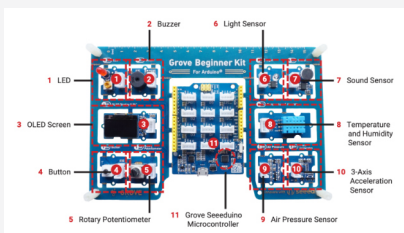
If you asked participants to do this before the session, remind them here as they may be able to download these things during the presentation.

If you did not, you can take some time here to help participants download these resources.



# Slide Deck Notes

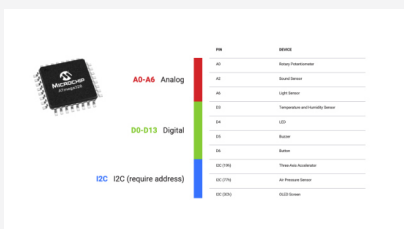
## Lesson 1: Slides 5-6



Corresponds to p6-7 of the guide. Introduce the different parts of the Grove board. Clicking through the presentation will bring up each label in turn.

The left side of the board contains some simple input and output devices. The right side of the board has an array of sensors useful for environmental monitoring. The central section is the 'brains' of the board where the Arduino microcontroller is. It also has white plug sockets and yellow header sockets for plugging in extra components.

1. LED – small low-power red light
2. Buzzer – buzzer which can emit at different frequencies The LED and buzzer are useful output devices used for warnings and alarms.
3. OLED screen – organic LED screen. Useful screen 128x64 pixels, used to display text or graphics. You will use this in the next session.
4. Button – simple press button. Can be used to turn something on or off
5. Potentiometer – a dial you can twist.
6. Light sensor
7. Sound sensor
8. Temperature and humidity sensor, also know as a hygrometer
9. Air pressure sensor, also known as a barometer – measures temperature and air pressure, can be used to measure altitude
10. 3-axis acceleration sensor – like the tilt sensor in phones
11. This is the microcontroller chip



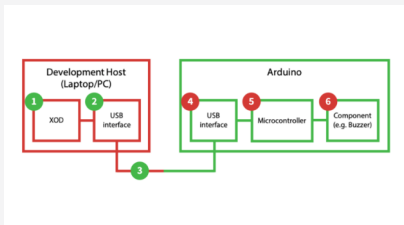
Corresponds to p8 & p10 of the guide. Explain the concept of a microcontroller.

- This tiny chip is the microcontroller used on the board
- It acts like a small computer – it has a processor and memory
- Unlike a normal computer it can only load one programme at a time i.e. it can only do one thing at a once
- Once you're programmed it, it will perform the same task each time
- You can reprogramme it by uploading a new programme
- It communicates with the other devices using the metal pins at the edge
- Each device is connected to a different pin, and the pins are given names
- There are different types of pins analog, digital and I2C
- Analog pins are used for analog devices that can have any continuous value
- Digital pins are used for digital devices that can only take on and off signals
- I2C is a special type of digital communication. You can use it to connect more than one device to the same pin
- I2C devices need a name to tell them apart called an address e.g. 19h, 3Ch
- In XOD these pins are referred to as 'ports'



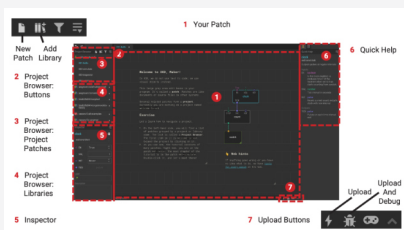
# Slide Deck Notes

## Lesson 1: Slides 7-10



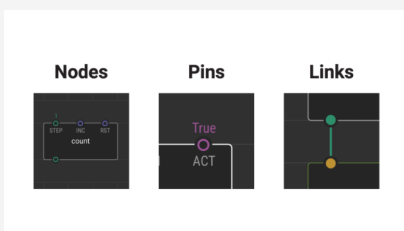
Corresponds to p9 of the guide. Explain how the microcontroller is programmed.

1. Write your programme using XOD software
2. Upload your programme
3. Information is sent from the computer to the Grove board via a USB cable
4. Information is received by the Grove board
5. The programme is written to long-term memory of the microcontroller chip. This allows the board to act as its own independent computer, carrying out the specific programme you have uploaded.
6. Information is sent to the onboard components via the microcontroller pins. The programme stored in the microcontroller's memory will tell the components what to do, for example, turn on the buzzer at a certain pitch.



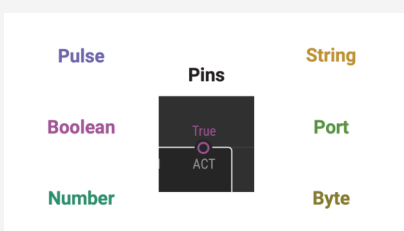
Corresponds to p12-13 of the guide. Introduce the XOD software.

1. The central area of the screen is called the patch. It is like a blank canvas or file that you use to build your programme. You build programmes by adding boxes called nodes to the patch
2. Next to 'Project Browser' are the new page and add library buttons
3. Below that is your project (like a folder) with patches (like files) in it. If you can't see your patches click the down arrow next to your project name
4. Below that are your libraries – where the parts to build your programme are stored. XOD has some libraries pre-installed, but you can add new ones
5. Inspector – this is where you change information about the nodes
6. Quick help pane – find information about nodes and pins. If you can't see the help pane, click on the question mark button
7. Upload buttons – lightning button uploads your programme, ladybird button uploads and also starts XOD's 'debugger' - a simulation mode



Corresponds to p14 of the guide. Introduce some XOD terminology.

- XOD programmes are built using 'Nodes', which look with boxes with a name in the middle and circles on the top and bottom
- Nodes can represent hardware, like the buzzer, or a function, like addition
- The circles on a node are called pins and they represent inputs and outputs
- Input pins are on the top, output pins are on the bottom
- These are not the same as the microcontroller pins, which XOD calls 'ports'
- Pins are coloured according to the type of data they transmit
- You build programmes by connecting nodes together with links



Corresponds to p15 of the guide. Introduce XOD data types

- Each data type has a different colour pin, and you can usually only connect an output pin to an input pin of the same colour/data type
- Pulse pins act like triggers, telling the programme when to stop or start
- Boolean pins have only two states: true and false (or on and off)
- Number pins represent numbers
- String pins represent text
- Port pins identify which 'port' (microcontroller pin) a device is connected to
- Byte pins represent bytes – a fundamental data type in computing

# Slide Deck Notes

## Lesson 2: Slides 1-5



### Testing Your Board

Demonstrate Task 1 (p20-25 of the guide) which will consolidate what they have learned in Lesson 1. Task 1 uses the *led*, *button* and *not* nodes to control the onboard LED using the button.

Try to show both how you build the programme on your computer screen (using a projector or share screen) and what happens to your board when you upload (via physical demonstration or a video).

Include a demo of the different ways to add nodes to patch. We suggest participants work in groups of 3-5. For online sessions breakout rooms are good for this.

#### Try it Yourself – 20min

- 1 Work in small groups (introduce yourselves if necessary)
- 2 Complete Task 1
- 3 Step-by-step instructions are in the Guide (p20-25)
- 4 Ask if you need help

Use this slide as a place holder whilst participants try the task.



### Inputs and Outputs

Demonstrate Task 2 (p26-29 of the guide), which replaces the LED with a buzzer, and adds a potentiometer to control the pitch.

You can either demonstrate both Tasks 1 and 2, then get participants to try the tasks (suggested for online workshops), or you can demonstrate task one, get participants to try, then demonstrate task 2 and get participants to try (suggested for in person workshops).

#### Try it Yourself – 20min

- 1 Work through Task 2 in groups
- 2 Step-by-step instructions are in the Guide (p26-29)
- 3 Ask if you need help

Use this slide as a place holder whilst participants try the task.

### Congratulations!

**You can now programme  
an Arduino Board!**

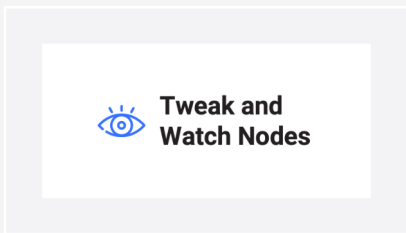
This slide is the end of Lesson 2.

Take this opportunity to help out anyone who is having trouble with their boards or programmes.

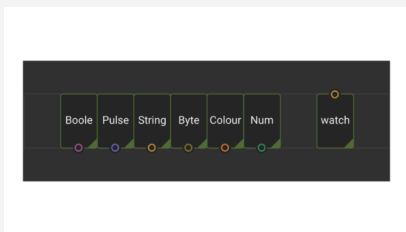


# Slide Deck Notes

## Lesson 3: Slides 1-4

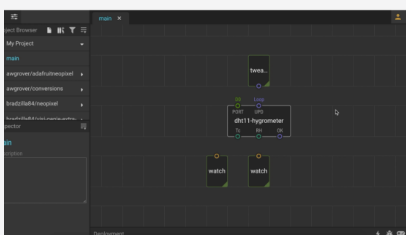


Use this slide to introduce Lesson 3: Explore XOD. This lesson will cover some of the most useful nodes in XOD.



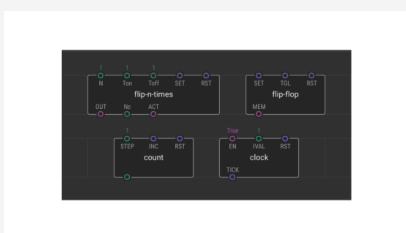
Corresponds to p33 of the guide. Introduce tweak and watch nodes.

- Tweak and watch nodes allow us to view and edit our programme in real-time
- There are different types of tweak nodes for different data types
- There is only one type of watch node that can be connected to all data types except pulses (will we go over how to get around that)
- When using tweak and watch nodes you need to use the 'Upload and Debug' button to upload. This will open XOD's 'Debugger' (simulator) and let us watch/edit the nodes



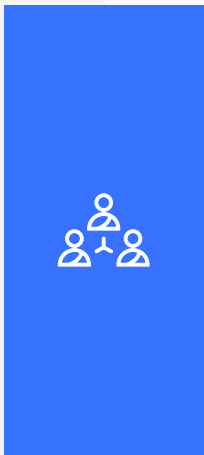
Play the video on this slide to demonstrate Task 3 (p34-35), which uses a *tweak* node to initiate readings from the temperature and humidity sensor, and *watch* nodes to view the results.

You can either narrate the video as it plays, or explain what will happen, then play the video. If you can not play the video from the slides you can demonstrate yourself, or play a recording: <https://www.youtube.com/watch?v=PNv7bEZXiw&t=359s>.



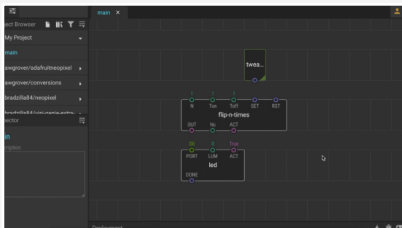
Corresponds to p36 of the guide. Introduce flip, count and clock nodes.

- These nodes are useful for controlling and monitoring programme timings
- Flip-n-times and flip-flop switch between two states 'on' and 'off'. They are useful as switches.
- The clock node creates a regular 'ticking' of pulses. Useful to control timings
- The count node counts up each time it receives a pulse. It can be used for control e.g. programme starts after a certain number of clicks or for monitoring e.g. watching how many times your programme has cycled.
- You can use the count node combined with a watch node to watch the output from pulse pins



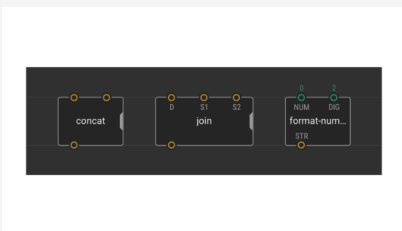
# Slide Deck Notes

## Lesson 3: Slides 5-7



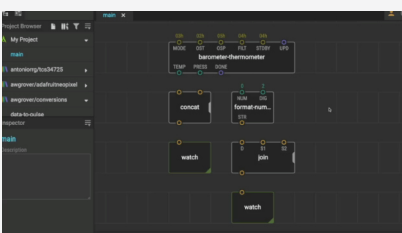
Play the video on this slide to demonstrate Task 4 (p37-40), which demonstrates how the *flip-n-times*, *flip-flop* and *clock* nodes can be used to control the LED in different ways, and how the *count* and *watch* nodes can be used monitor the programmes progress.

You can either narrate the video as it plays, or explain what will happen, then play the video. If you can not play the video from the slides you can demonstrate yourself, or play a recording: <https://www.youtube.com/watch?v=PNv7bEZTxiw&t=723s>.



Corresponds to p42 of the guide. Introduce *concat*, *join* and *format-number* nodes.

- These nodes are useful for formatting outputs
- *Concat* and *join* are similar, they will join two or more strings
- *Concat* connects strings directly, *join* connects them with a 'delimiter' (D) in between
- You can chose what this delimited is
- This is useful for collecting data. You can use a comma or tab delimiter to create .csv or .tsv files
- *Concat* and *join* are variadic nodes – this means you can pull the tab on the right edge of the node to increase the number of inputs
- *Format-number* sets the number of decimal points used in your output
- There are other ways to format numbers explained in the beginner's guide (p45)



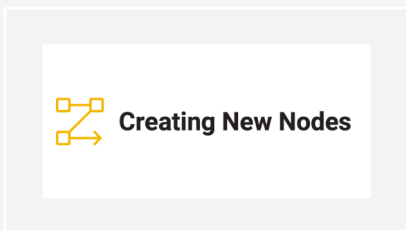
Play the video on this slide to demonstrate Task 5 (p43-45), which demonstrates how to use these nodes to format the outputs from the air-pressure sensor.

You can either narrate the video as it plays, or explain what will happen, then play the video. If you can not play the video from the slides you can demonstrate yourself, or play a recording: <https://www.youtube.com/watch?v=PNv7bEZTxiw&t=1113s>

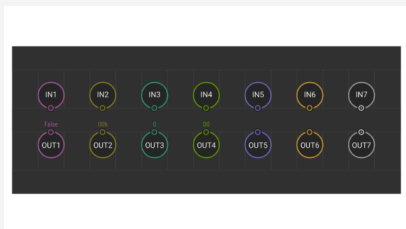
Note: there is a mistake in the YouTube version of this video. It is explained in the video and has been corrected here.

# Slide Deck Notes

## Lesson 4: Slides 1-4



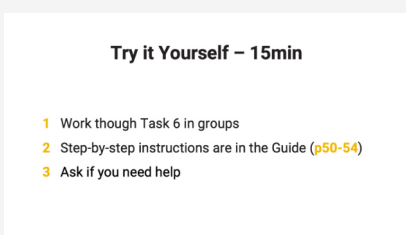
Use this slide to introduce Lesson 4: Building Devices. This lesson will give participants the skills they need to start building more complex programmes in XOD.



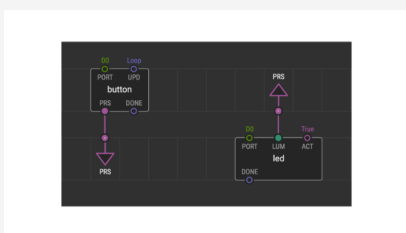
Corresponds to p49 of the guide. Introduce the concept of creating your own nodes.

- You can simplify your programme by encapsulating part of it in a new node
- You create a patch like normal, but then add 'terminal nodes' (represented here) to let your new node communicate with other nodes
- Terminals can be inputs (top row) or outputs (bottom row) and will become the pins of your new node
- There are different terminals for different data types e.g. input-boolean, input-number, output-number, output-pulse etc.
- The next task will demonstrate how to use these to create a node that writes text to the OLED display screen

Demonstrate Task 6 (p50-54). Participants will create a new node to write text on the OLED screen. Try to show both how you build the programme on your computer screen (using a projector or share screen) and what happens to your board when you upload (via physical demonstration or a video).



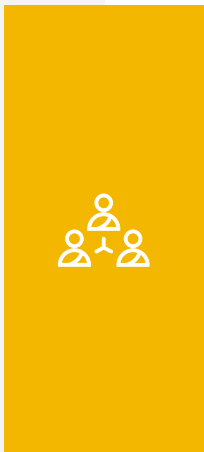
Use this side as a place holder whilst participants try the task.



Introduce the concept of buses.

- Bus nodes ('to-bus' and 'from-bus') are a way to link pins 'invisibly'
- They can be used to reduce the number of intersecting links and make the data flow easier to compartmentalise and visualise
- They automatically take on the data type that they're connected to
- 'to-bus' connects to outputs and sends information from an output to a bus
- 'from-bus' connects to inputs and retrieves information from a bus
- The to and from buses need to be given the same name to link them

Demonstrate Task 7 (p56-59), which shows how buses work by creating a game using the 3-axis acceleration sensor and screen. If you are short on time, you can modify the node created in Task 6 to demonstrate the use of buses instead.



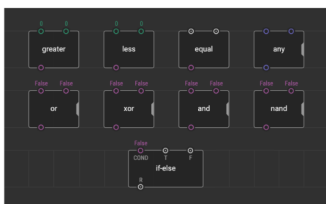
# Slide Deck Notes

## Lesson 4: Slides 5-9

### Try it Yourself – 15min

- 1 Work through Task 7 in groups
- 2 Step-by-step instructions are in the Guide (p56-59)
- 3 Ask if you need help

Use this side as a place holder whilst participants try the task.



Introduce logic nodes.

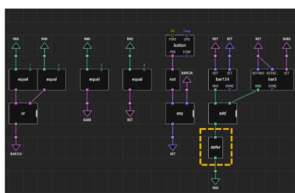
- Logic nodes represent logical functions such as AND and OR.
- They are often used in conjunction with maths nodes such as EQUAL
- Using these nodes you can instruct your programme to read different inputs or give different outputs depending on the conditions

Demonstrate Task 8 (p60-63), which uses logic nodes to return different text on the screen depending on the surrounding light intensity.

### Try it Yourself – 15min

- 1 Work through Task 8 in groups
- 2 Step-by-step instructions are in the Guide (p60-63)
- 3 Ask if you need help

Use this side as a place holder whilst participants try the task.



introduce the defer node.

- There are different ways to create sequences in XOD
- Creating loops in XOD requires use of the 'defer' node

Demonstrate Task 9 (p64-70), which demonstrates two ways of creating sequences and shows how to make loops using the *defer* node to play a simple tune on the buzzer.

### Try it Yourself – 15min

- 1 Work through Task 9 in groups
- 2 Step-by-step instructions are in the Guide (p64-70)
- 3 Ask if you need help

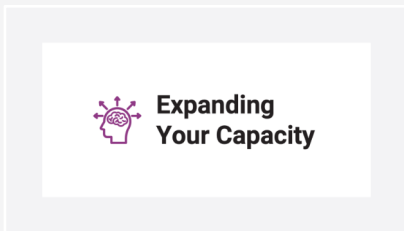
Use this side as a place holder whilst participants try the task.





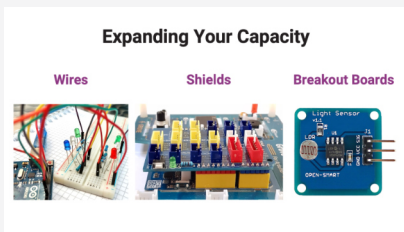
# Slide Deck Notes

## Lesson 5: Slides 1-3



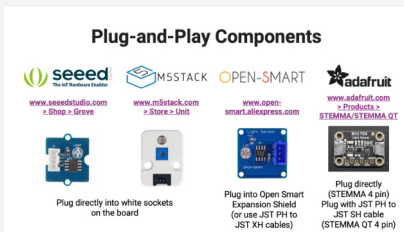
Use this slide to introduce Lesson 5: Next Steps.

This Lesson aims to equip participants with the knowledge of where to go next, how to start adding hardware to their boards, and where to find help and advice for their own projects.



Corresponds to p76. Explain how participants can expand the use of their boards and create new devices using additional components.

- You can add components to your board in many different ways
- You can use electronic wiring, either directly or via a breadboard (this requires a level of electronics knowledge)
- You can use shields, which plug into the yellow header pins on the left and right edge of the central Grove board module
- Shields can add capacity (such as internet connection shields) or allow you to easily plug in breakout boards (expansion shields)
- Breakout boards are small PCBs with electronic components wired in
- They usually have one component per board and are a really easy option
- Each of the Grove modules is an inbuilt breakout board
- You can remove components from the board, or remove the central microcontroller portion to free-up the board.
- This helps avoid component clashes (i.e. two components on the same pin)
- Use a sharp craft knife or scalpel to cut components from the board (you can always plug them back in using the Grove sockets and wires)



Cover the different plug-and-play systems compatible with the Grove board.

- Seeed Studio Grove Components  
Plug directly into the board using Grove connector cables (included in your kit)
- M5Stack Components  
'Units' are compatible using Grove or M5Stack cables
- Open Smart Easy Plug Components  
Similar plug-and-play system, but not directly compatible  
Use easy-plug expansion shield or JST PH-JST XH cables
- Adafruit STEMMA/STEMMA QT Components  
STEMMA 4-pins compatible using Grove or JST PH cables  
STEMMA QT 4-pins compatible using JST PH-JST SH cables
- SparkFun Qwiic Components (not shown here)  
4-pins compatible using JST PH-JST SH cables  
Compatible with STEMMA QT



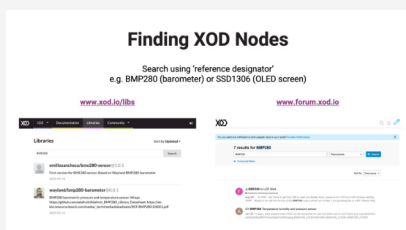
# Slide Deck Notes

## Lesson 5: Slides 4-7



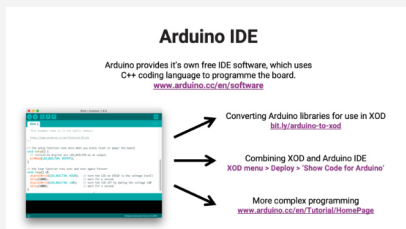
Cover how to add non-plug-and-play components to the board.

- Open Smart components come with pins attached
- Connect them using a Grove-female cable
- Other breakout boards (e.g. from Sparkfun or Adafruit) come without pins
- Connect them by soldering pins to the board, then use Grove-female cables
- You need to make sure that the wires are in the right order i.e. the pin labels match up (you may need to learn a bit about electronics to get this right)



Let them know where they can find XOD nodes for new components.

- It is easiest to search using the component's 'reference designator' (standardized letter and number code)
- Search XOD's library database or on the XOD forum
- Not all components have XOD nodes, or can be used in XOD



Explain the Arduino IDE, and why they might want to use it.

- Not all components have XOD nodes – sometimes you may want or need to use the Arduino IDE instead
- Arduino IDE is free and uses C++ programming language
- Sometimes you can convert Arduino libraries into XOD libraries – we have a tutorial on how to do this (you will need some basic C++ knowledge)
- You can combine XOD and Arduino, i.e. by writing a programme in XOD, exporting it to C++, and adding to/adapting it in Arduino
- You may want to do this if you have code in XOD but want to add a component that doesn't have a XOD node.
- Go to 'Deploy' in the menu bar in XOD, then click 'Show Code for Arduino'.
- Some complex programmes may be more efficient if written in Arduino.
- The Arduino website provides simple tutorials on how to use Arduino and learn simple C++ if you are interested



Pages 80-83 of the Beginner's Guide contain 4 case studies of previous Biomaker projects, including details of how they could be adapted for use with the Grove board and XOD programming.

You can find more examples of DIY biological hardware on the Biomaker Hackster hub.

Use this slide to initiate discussion around the case studies and potential projects that participants might want to build for their own research.

