

No-Code Programming for Biology



- 1** Welcome!
- 2** Turn off your video and mute yourselves for now
- 3** Feel free to introduce yourselves in the chat
- 4** As we go along, ask any questions in the chat

No-Code Programming for Biology



Today's Session

17:00 Welcome!

17:05 Lesson 1: Introduction

An introduction to the grove board, microcontrollers and the XOD IDE

17:30 Lesson 2: Getting Started (hands-on session)

Get started with using your board. We'll start with some simple tasks like flashing an LED, pressing a button and sounding a buzzer

18:25 Round-up

Morning Session

09:00 Welcome & Lesson 1: Introduction

An introduction to us, the grove board, microcontrollers and the XOD IDE

10:00 Break

10:30 Lessons 2 & 3: Getting Started & Explore XOD

Get started with using your board. We'll start with some simple tasks like flashing an LED, pressing a button and sounding a buzzer

Get to grips with some of the most useful nodes in XOD

12:00 Lunch



No-Code Programming for Biology



Before we Start

1 Downloaded the XOD Software

www.xod.io

2 Downloaded the No-Code Programming Beginner's Guide

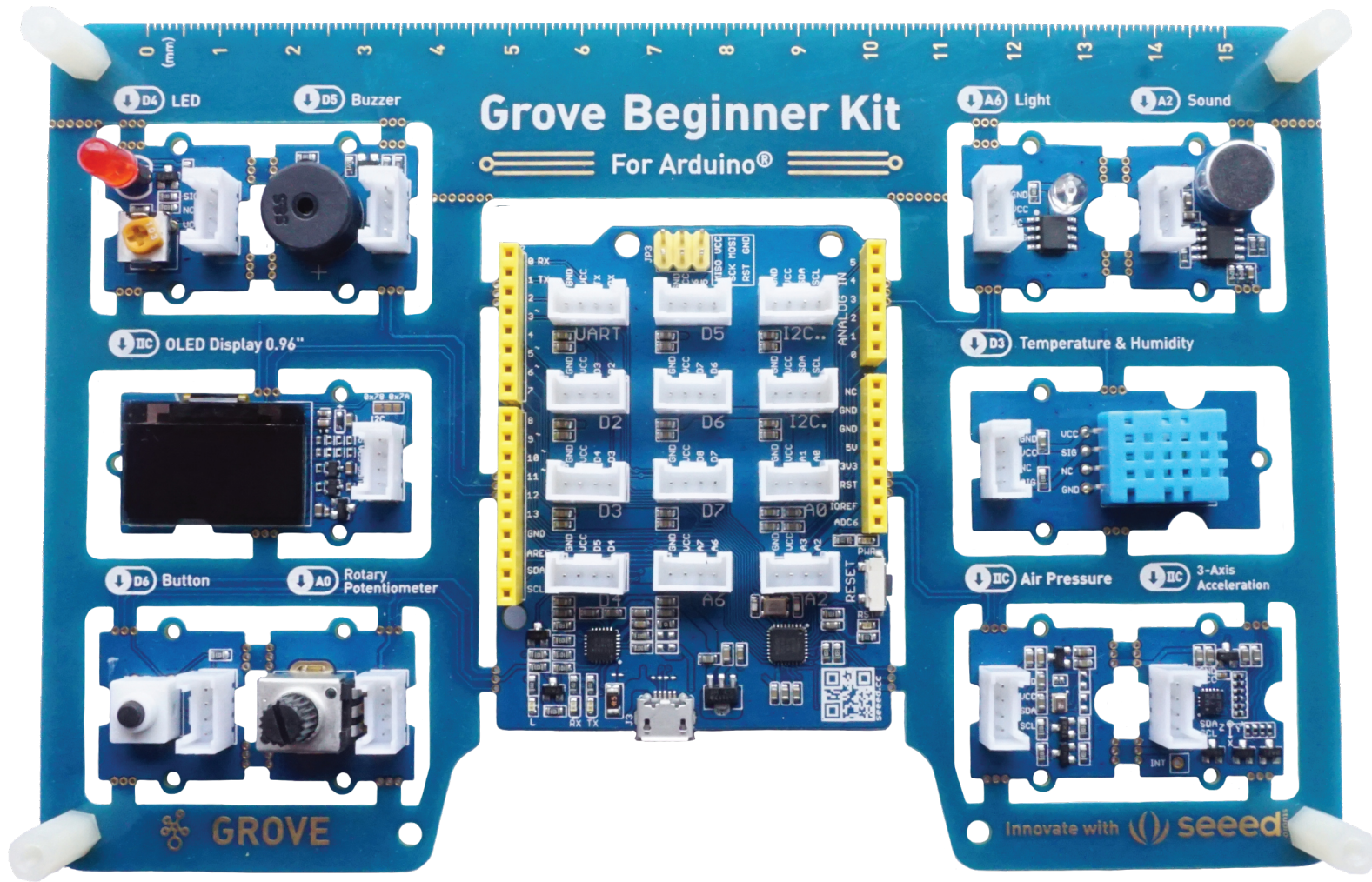
www.biomaker.org/nocode-programming-for-biology-handbook

3 Installed USB Drivers (if required)

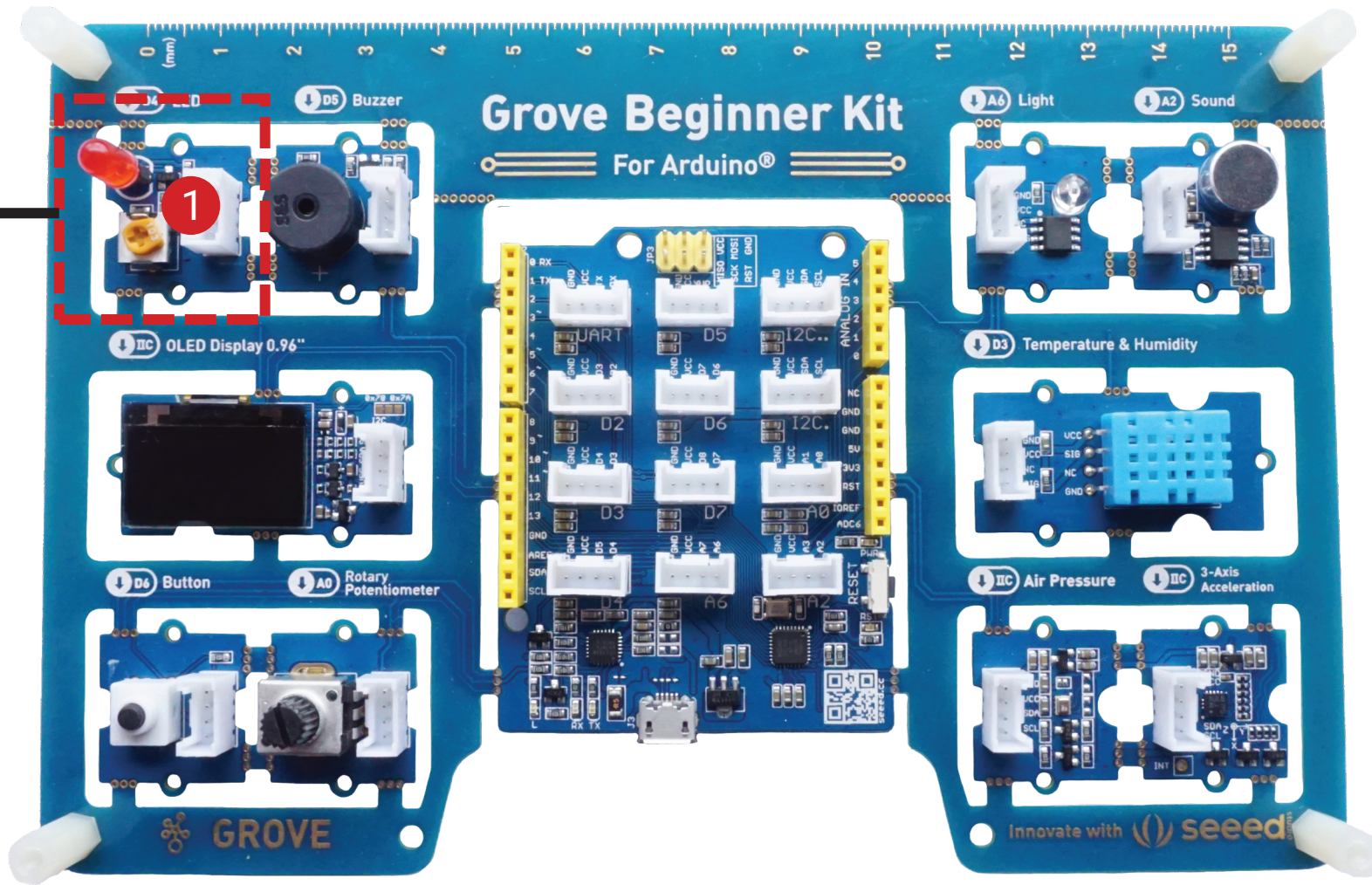
www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers



The Starter Kit

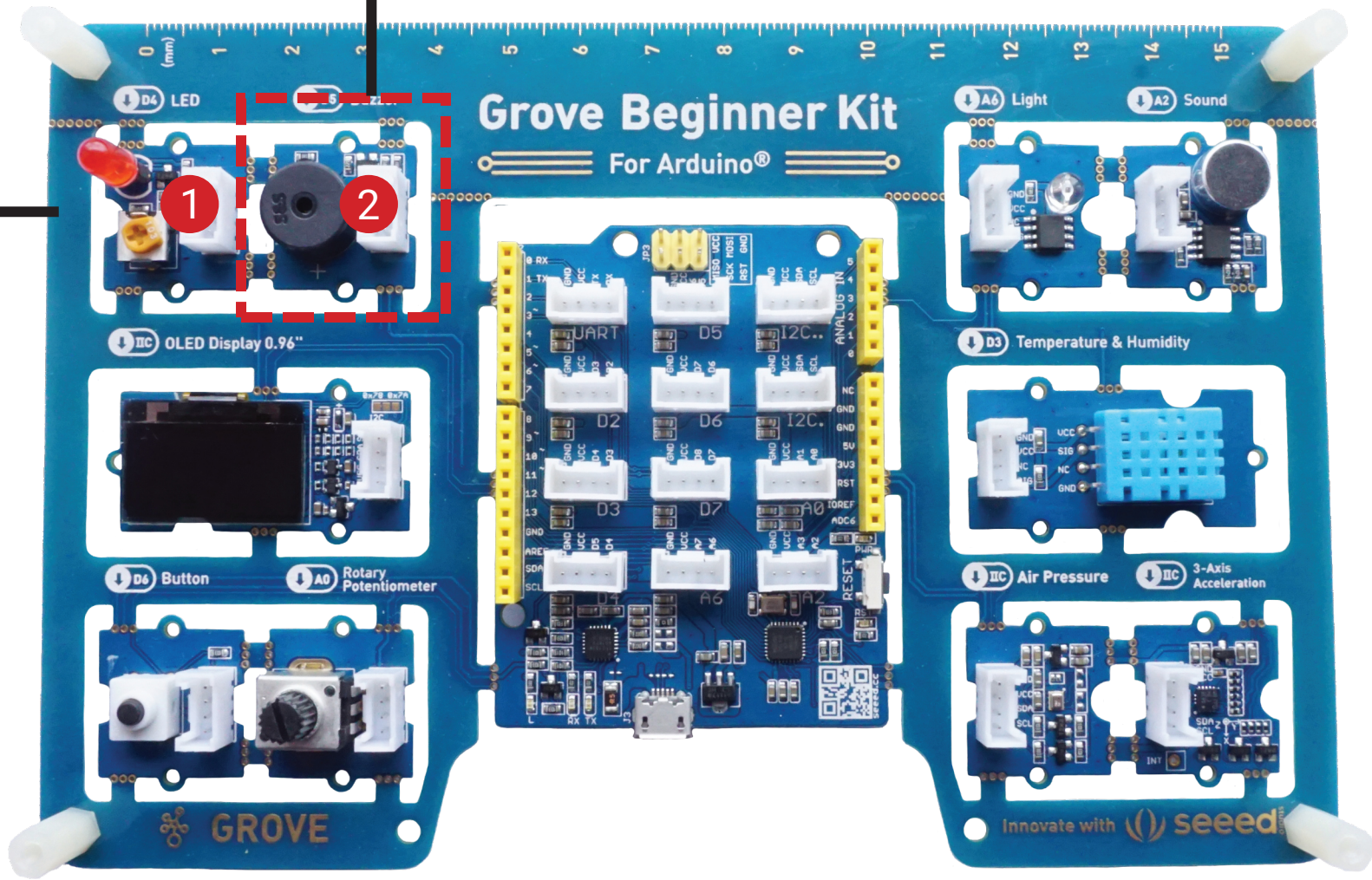


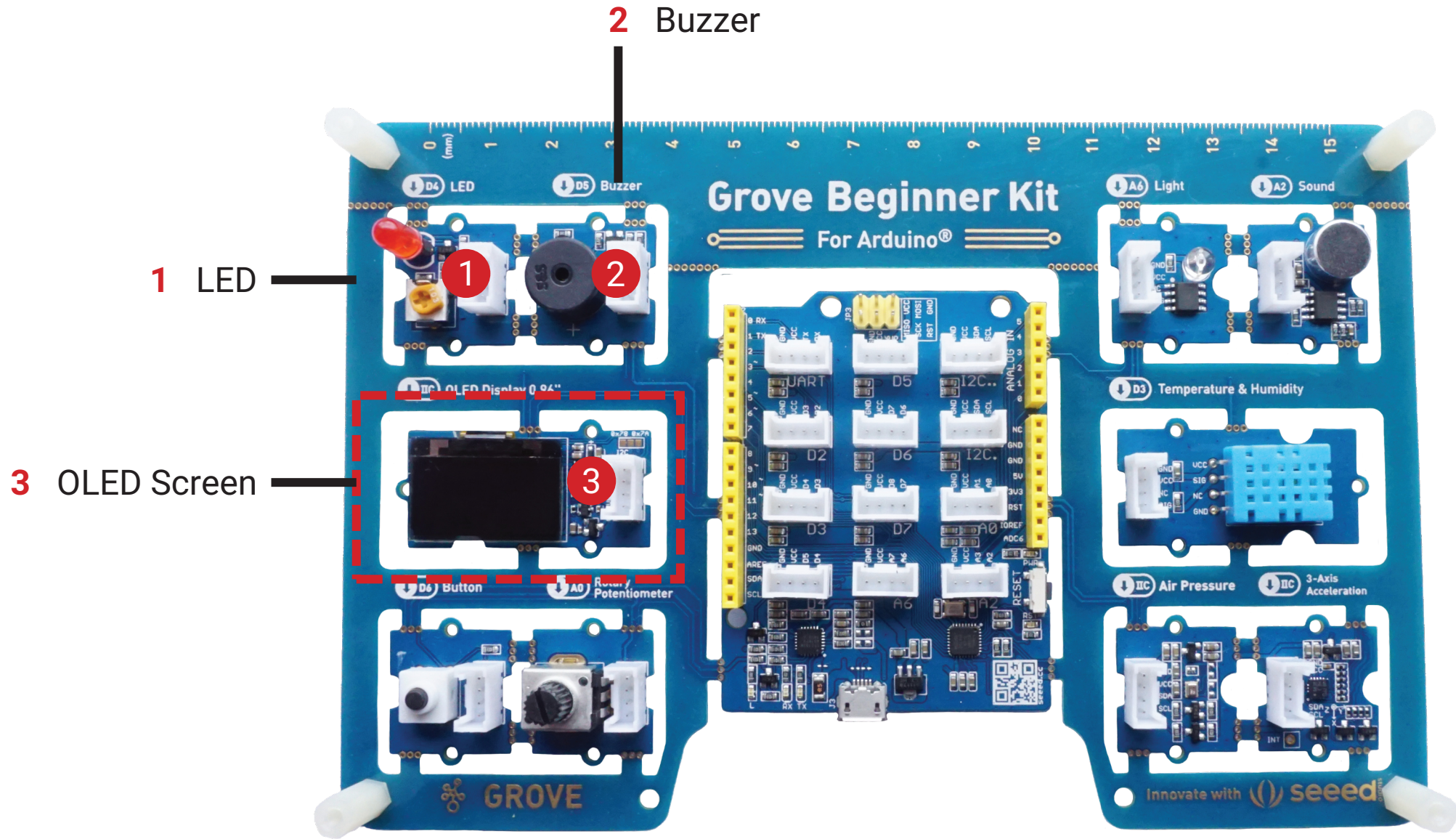
1 LED

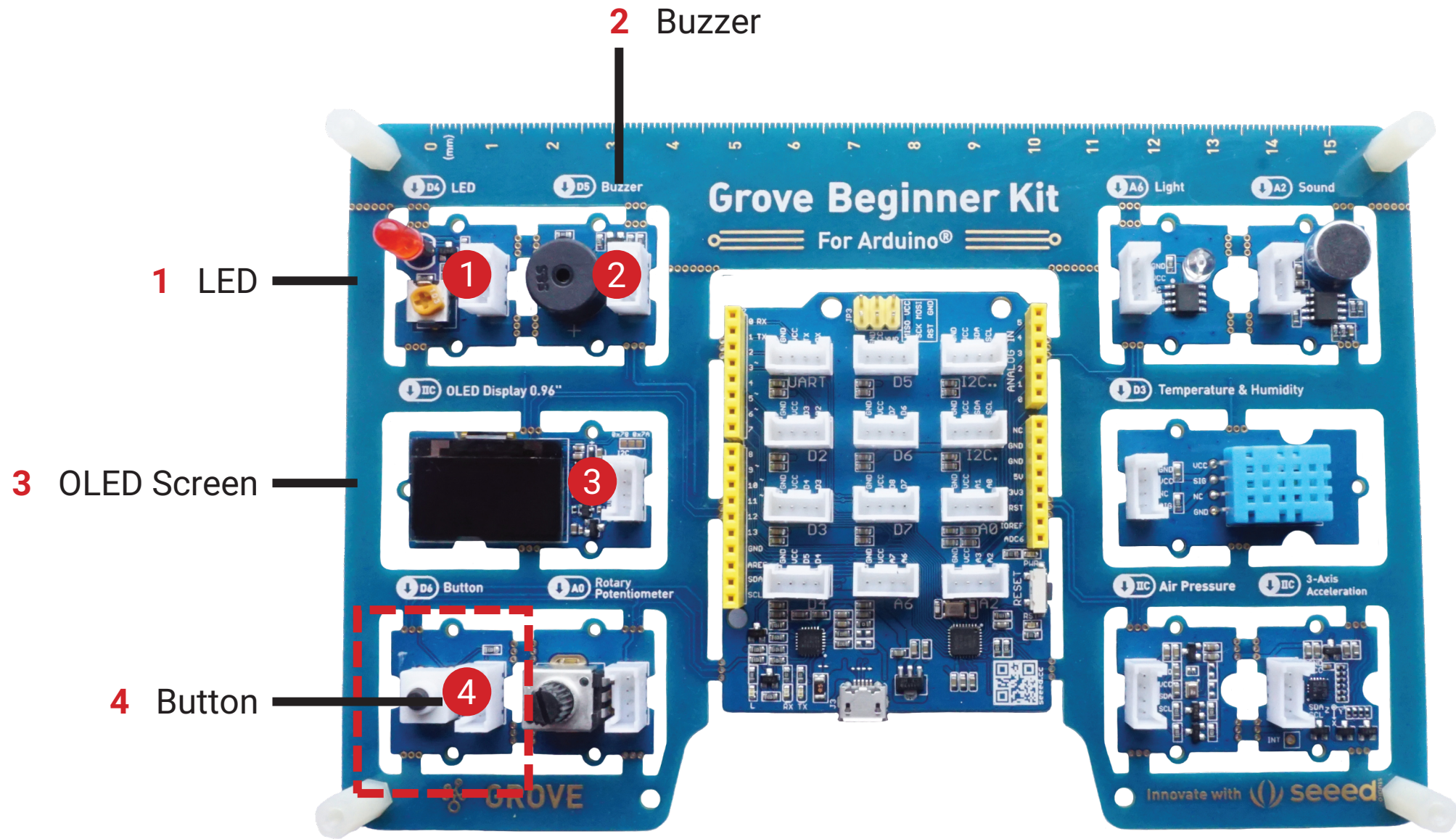


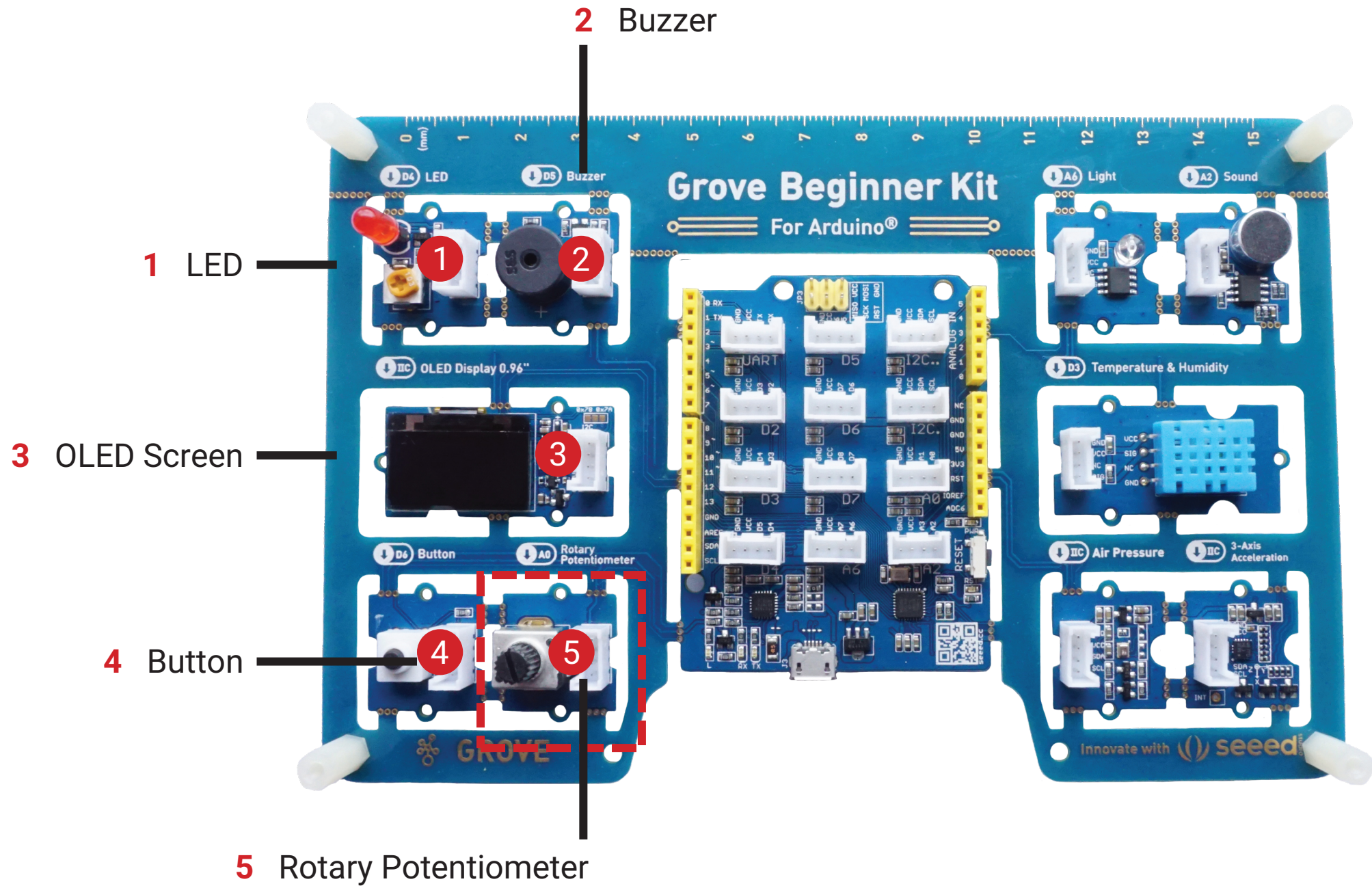
1 LED

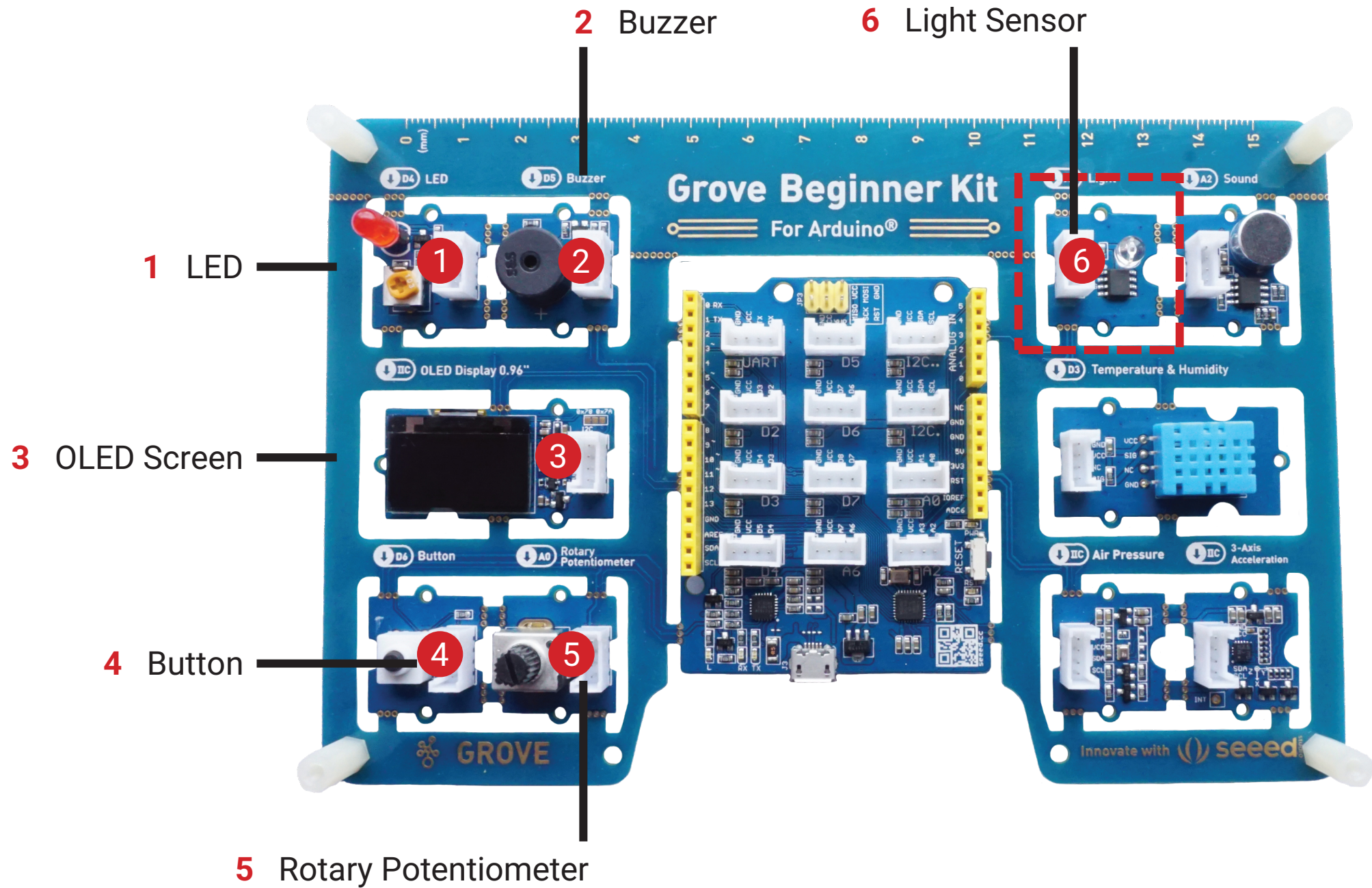
2 Buzzer

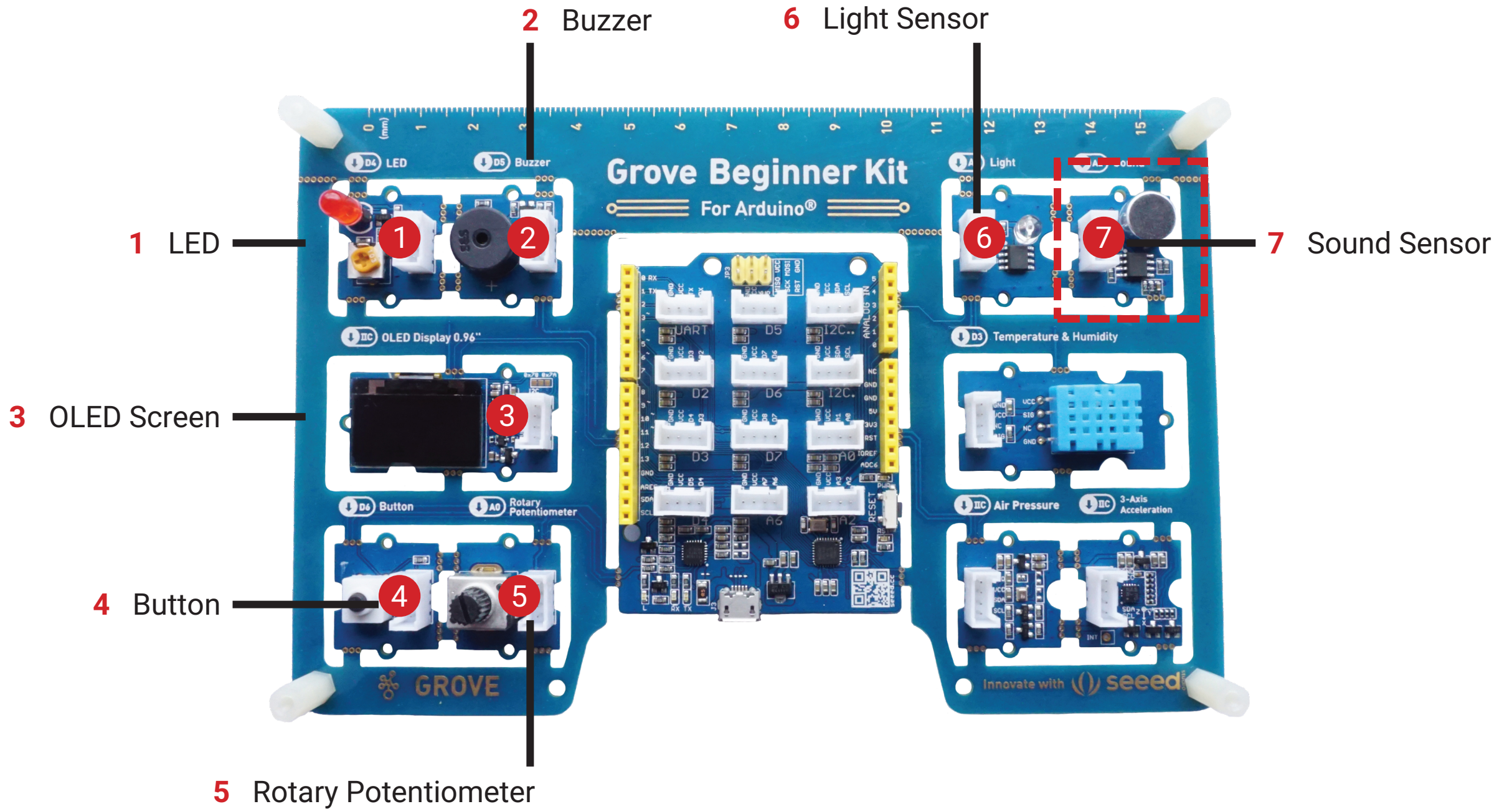


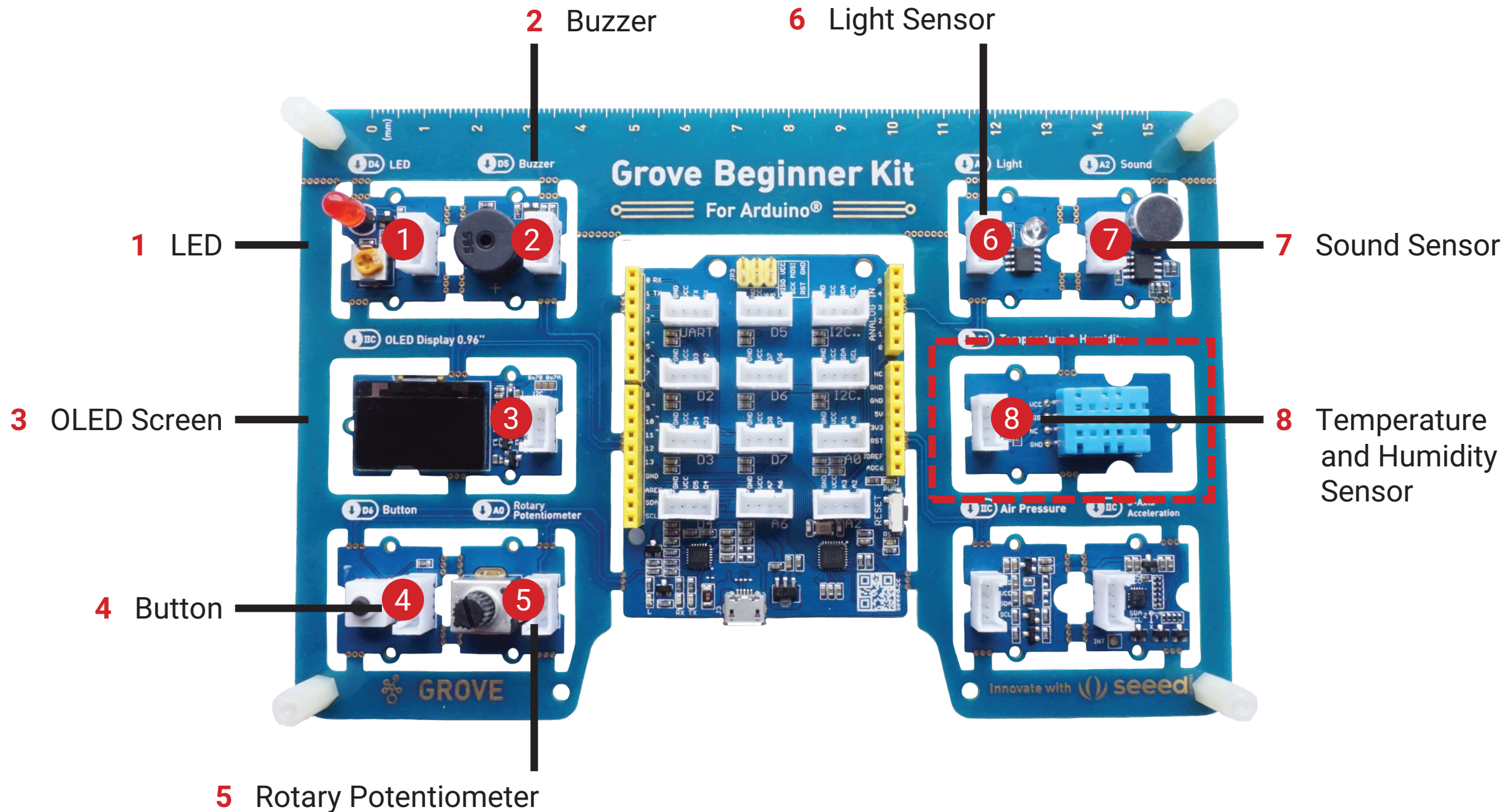


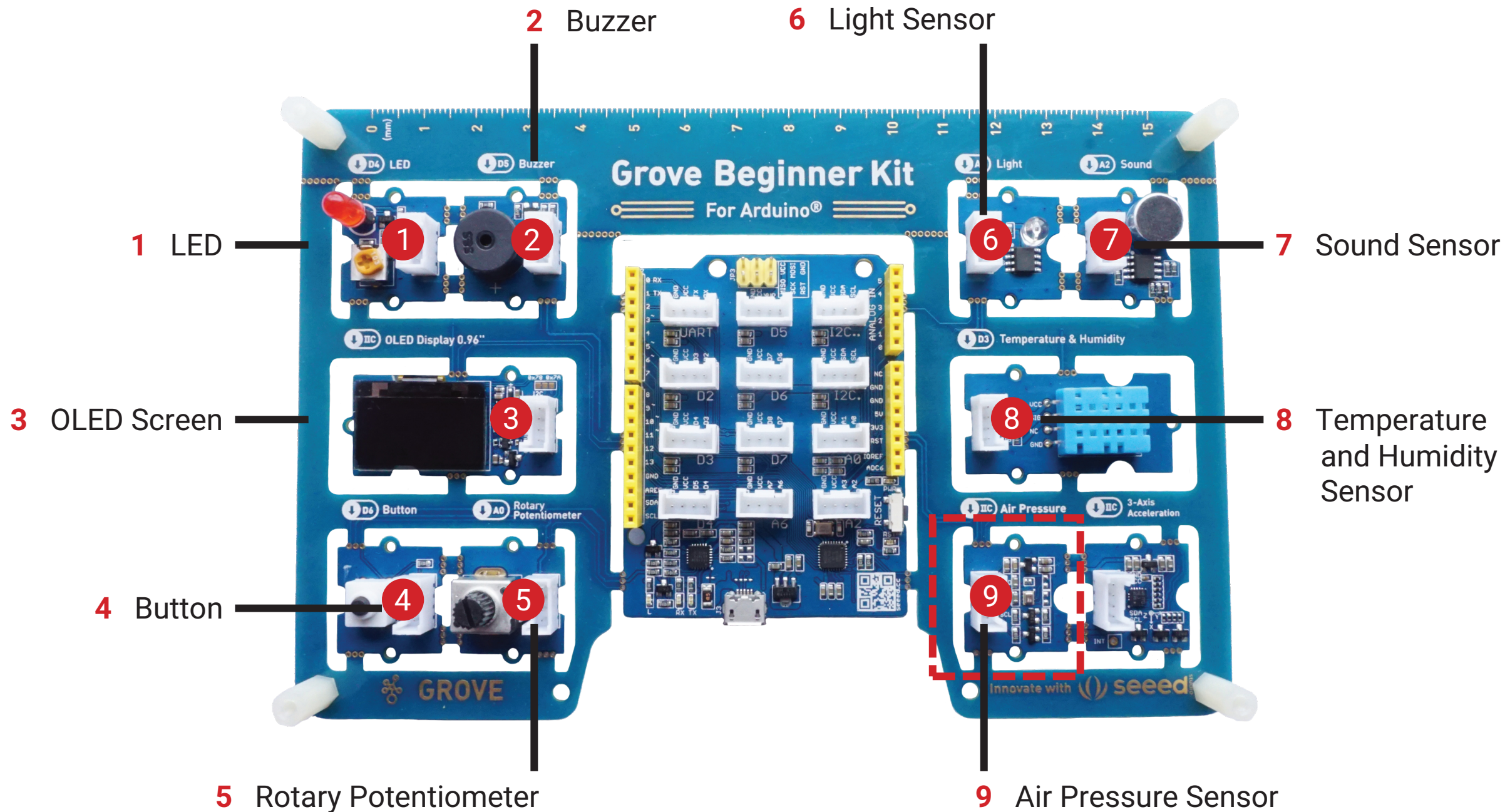


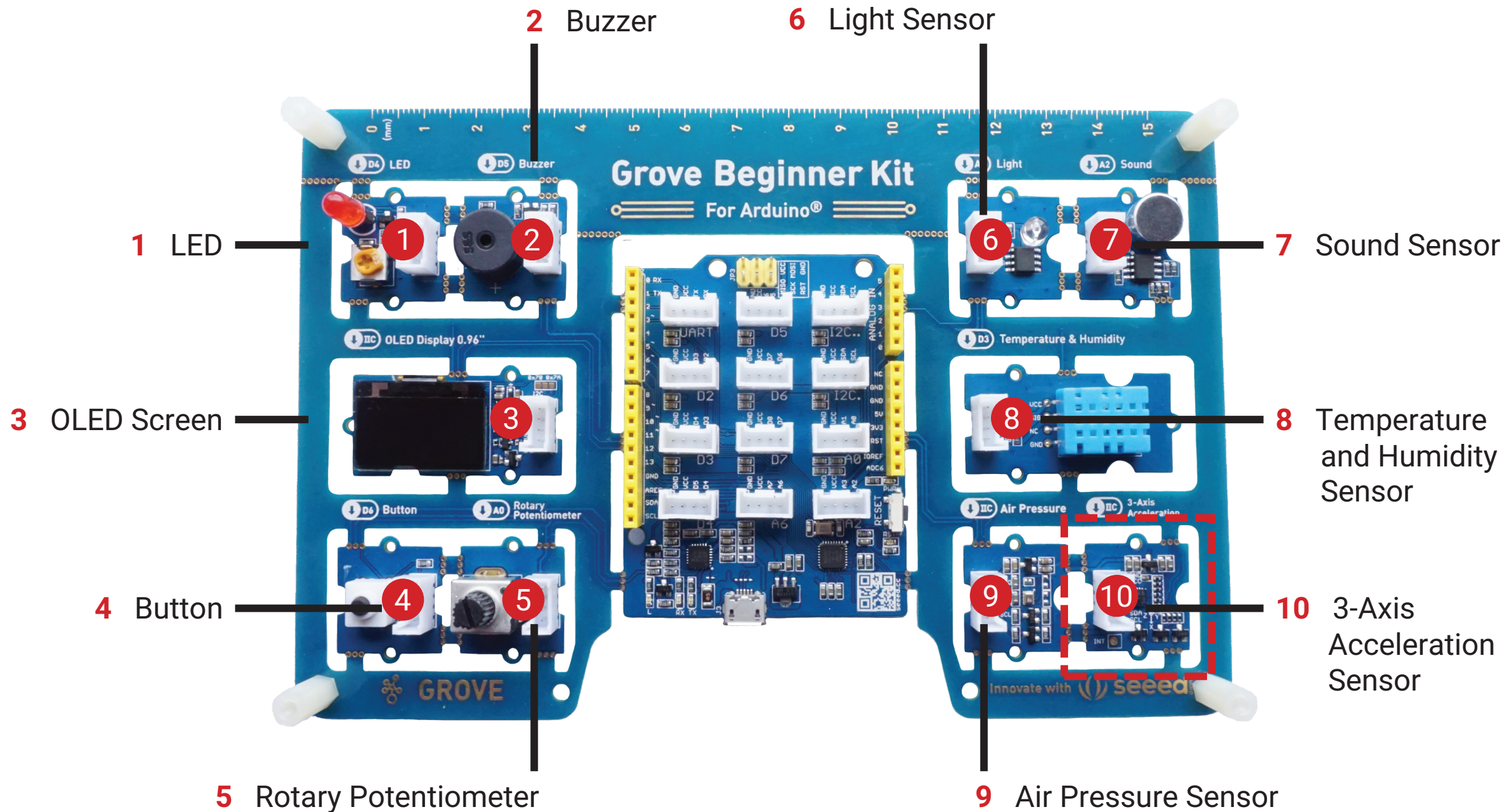


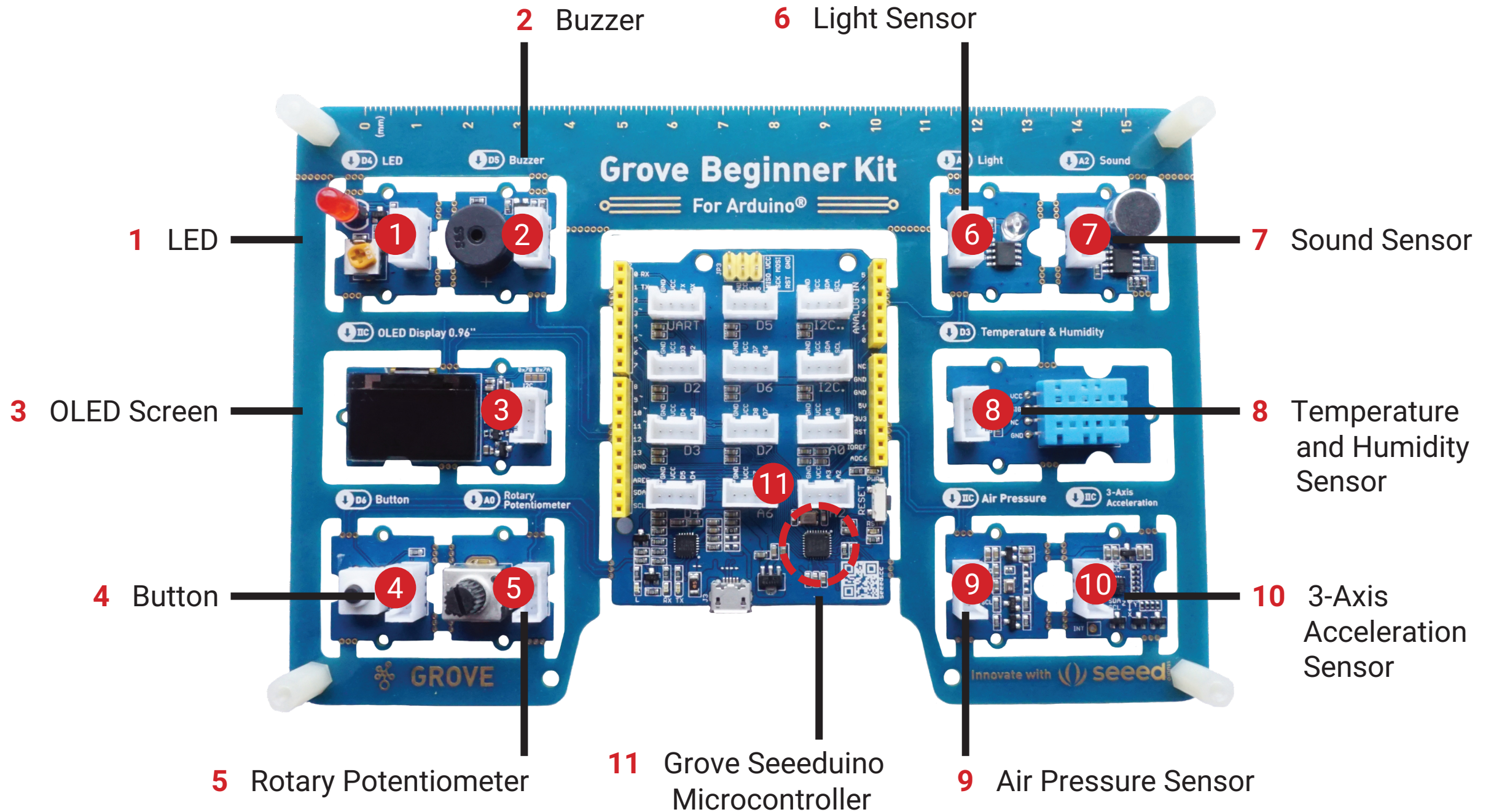


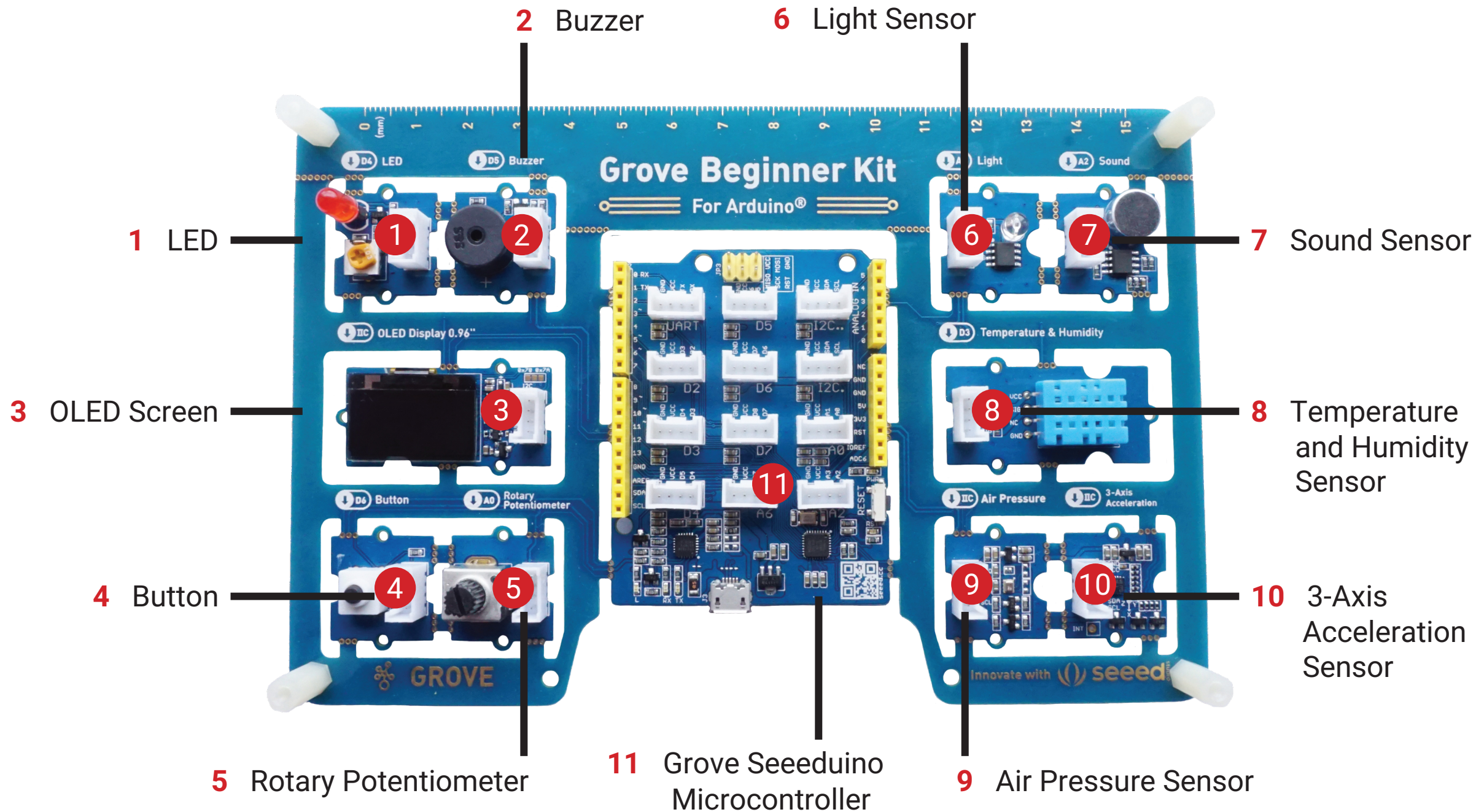


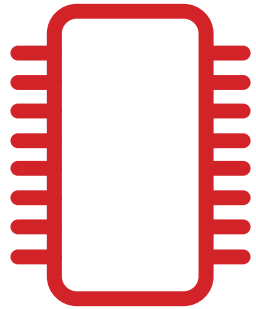




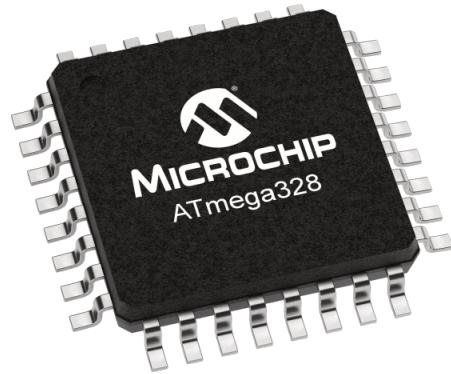








The Microcontroller



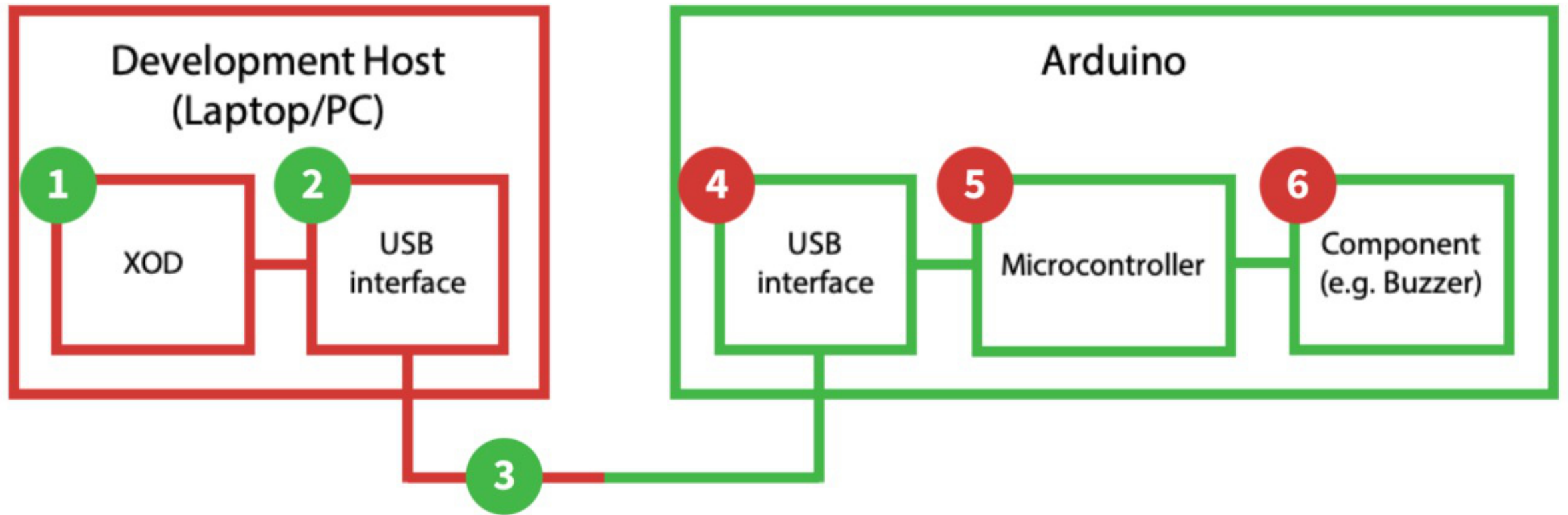
A0-A6 Analog

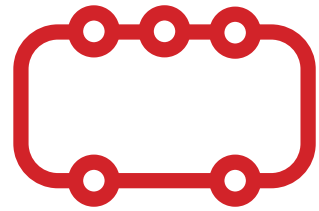
D0-D13 Digital

I2C I2C (require address)



PIN	DEVICE
A0	Rotary Potentiometer
A2	Sound Sensor
A6	Light Sensor
D3	Temperature and Humidity Sensor
D4	LED
D5	Buzzer
D6	Button
I2C (19h)	Three-Axis Accelerator
I2C (77h)	Air Pressure Sensor
I2C (3Ch)	OLED Screen





The XOD IDE

Project Browser

- welcome-to-xod
- 001-hello
- 002-simulate
- 003-inspector
- 004-patching

- awgrover/adafuitneopixel
- awgrover/conversions
- bradzilla84/neopixel
- bradzilla84/visi-genie-extra-library
- cesars/0-all-examples
- cesars/i2c-scanner

Inspector

clock

xod/core/clock

EN True

IVAL 1

RST Never

TICK pulse

Label

Description

001-hello x

Welcome to XOD, Maker!

In XOD, we do not use text to code; we use visual objects instead.

This large gray area with boxes is your program. It's called a **patch**. Patches are like documents or source files in other systems.

Several related patches form a **project**. Currently you are working on a project named welcome-to-xod.

Exercise

Let's learn how to navigate a project.

- On the left-hand side, you will find a list of patches grouped by a project or library name. The list is called a **Project Browser**. The first item in it is welcome-to-xod. Expand the project by clicking on it.
- As you can see, the tutorial consists of many patches. Right now, you are in the patch 001-hello. The next chapter of the tutorial is in the patch 002-simulate. Double-click it, and let's meet there!

Web hints

If anything goes wrong or you have no idea what to do, we have [hints for every patch](#) on the web.

Quick Help

clock

xod/core/clock

Outputs pulses at regular intervals

Inputs:

EN **boolean**

Is the clock enabled, i.e. produces ticks? At the moment when set to true, starts counting from scratch.

IVAL **number**

Tick interval in seconds

RST **pulse**

Resets current count, restarts clock with new interval

Outputs:

TICK **pulse**

Pulses on each time interval end

Deployment

1 Your Patch

001-hello

Welcome to XOD, Maker!

In XOD, we do not use text to code; we use visual objects instead.

This large gray area with boxes is your program. It's called a **patch**. Patches are like documents or source files in other systems.

Several related patches form a **project**. Currently you are working on a project named `welcome-to-xod`.

Exercise

Let's learn how to navigate a project.

1. On the left-hand side, you will find a list of patches grouped by a project or library name. The list is called a **Project Browser**. The first item in it is `welcome-to-xod`. Expand the project by clicking on it.
2. As you can see, the tutorial consists of many patches. Right now, you are in the patch `001-hello`. The next chapter of the tutorial is in the patch `002-simulate`. Double-click it, and let's meet there!

Web hints

If anything goes wrong or you have no idea what to do, we have [hints for every patch](#) on the web.

clock
xod/core/clock
Outputs pulses at regular intervals

Inputs:

- EN **boolean**
Is the clock enabled, i.e. produces ticks? At the moment when set to true, starts counting from scratch.
- IVAL **number**
Tick interval in seconds
- RST **pulse**
Resets current count, restarts clock with new interval

Outputs:

- TICK **pulse**
Pulses on each time interval end

count

STEP 1, INC, RST

watch

Inspector: EN True, IVAL 1, RST Never, TICK pulse

Deployment



New Patch Add Library

2 Project Browser: Buttons

1 Your Patch

The screenshot shows the XOD Maker interface. On the left is the **Project Browser** with a list of patches: 001-hello, 002-simulate, 003-inspector, and 004-patching. Below this is the **Inspector** for the selected 'clock' patch, showing parameters: EN (True), IVAL (1), RST (Never), and TICK (pulse). The main workspace shows a patch diagram with a 'clock' block and a 'count' block. A red circle with the number '1' highlights the 'clock' block. A red circle with the number '2' highlights the 'Project Browser' header. A 'watch' block is also visible in the diagram. The right sidebar contains a 'Quick Help' section for the 'clock' patch, detailing its inputs (EN: boolean, IVAL: number, RST: pulse) and outputs (TICK: pulse).

Welcome to XOD, Maker!

In XOD, we do not use text to code; we use visual objects instead.

This large gray area with boxes is your program. It's called a **patch**. Patches are like documents or source files in other systems.

Several related patches form a **project**. Currently you are working on a project named `welcome-to-xod`.

Exercise

Let's learn how to navigate a project.

1. On the left-hand side, you will find a list of patches grouped by a project or library name. The list is called a **Project Browser**. The first item in it is `welcome-to-xod`. Expand the project by clicking on it.
2. As you can see, the tutorial consists of many patches. Right now, you are in the patch `001-hello`. The next chapter of the tutorial is in the patch `002-simulate`. Double-click it, and let's meet there!

Web hints

If anything goes wrong or you have no idea what to do, we have [hints for every patch](#) on the web.



New Patch Add Library

2 Project Browser: Buttons

3 Project Browser: Project Patches

1 Your Patch

The screenshot shows the XOD interface with three red callouts: '2' points to the Project Browser buttons, '3' points to the project list, and '1' points to the patch diagram. The Project Browser shows a tree view with 'welcome-to-xod' expanded to show '001-hello', '002-simulate', '003-inspector', and '004-patching'. The Inspector shows the 'clock' patch with parameters: EN (True), IVAL (1), RST (Never), and TICK (pulse). The patch diagram shows a 'clock' patch connected to a 'count' patch, which is connected to a 'watch' patch. The 'clock' patch has inputs EN, IVAL, and RST, and an output TICK. The 'count' patch has inputs STEP, INC, and RST, and an output TICK. The 'watch' patch has an input TICK and an output pulse.

Project Browser

- welcome-to-xod
 - 001-hello
 - 002-simulate
 - 003-inspector
 - 004-patching
- awgrover/adafruitneopixel
- awgrover/conversions
- bradzilla84/neopixel
- bradzilla84/visi-genie-extra-library
- cesars/0-all-examples
- cesars/i2c-scanner

Inspector

clock
xod/core/clock

EN: True
IVAL: 1
RST: Never
TICK: pulse

Label:
Description:

Exercise

Let's learn how to navigate a project.

1. On the left-hand side, you will find a list of patches grouped by a project or library name. The list is called a **Project Browser**. The first item in it is `welcome-to-xod`. Expand the project by clicking on it.
2. As you can see, the tutorial consists of many patches. Right now, you are in the patch `001-hello`. The next chapter of the tutorial is in the patch `002-simulate`. Double-click it, and let's meet there!

Web hints

If anything goes wrong or you have no idea what to do, we have [hints for every patch](#) on the web.



New Patch Add Library

2 Project Browser: Buttons

3 Project Browser: Project Patches

4 Project Browser: Libraries

1 Your Patch

The screenshot shows the XOD IDE interface. On the left is the Project Browser with a list of patches and libraries. A red dashed box highlights the library section, and a red circle '4' is next to it. The main area shows a patch editor with a 'clock' component and a 'count' component. A red circle '1' is next to the 'clock' component. The right side shows the Inspector for the 'clock' component, with a red circle '2' next to the 'clock' component name. A red circle '3' is next to the '001-hello' patch in the Project Browser. The interface includes a 'Welcome to XOD, Maker!' message, an 'Exercise' section, and 'Web hints'.

Project Browser

- welcome-to-xod
- 001-hello
- 002-simulate
- 003-inspector
- 004-patches
- awgrover/adafruitneopixel
- awgrover/convers
- bradzilla84/neopixel
- bradzilla84/visi-genie-extra-library
- cesars/0-all-examples
- cesars/i2c-scanner

Inspector

clock

xod/core/clock

EN True

IVAL 1

RST Never

TICK pulse

Label

Description

Deployment

Quick Help

clock

xod/core/clock

Outputs pulses at regular intervals

Inputs:

EN **boolean**

Is the clock enabled, i.e. produces ticks? At the moment when set to true, starts counting from scratch.

IVAL **number**

Tick interval in seconds

RST **pulse**

Resets current count, restarts clock with new interval

Outputs:

TICK **pulse**

Pulses on each time interval end

Web hints

If anything goes wrong or you have no idea what to do, we have [hints for every patch](#) on the web.



New Patch Add Library

1 Your Patch

2 Project Browser: Buttons

3 Project Browser: Project Patches

4 Project Browser: Libraries

5 Inspector

The screenshot shows the XOD Maker interface with several numbered callouts:

- 2**: Project Browser Buttons (top left icons)
- 3**: Project Browser: Project Patches (list of patches on the left)
- 4**: Project Browser: Libraries (list of libraries on the left)
- 5**: Inspector (right side panel showing the 'clock' patch configuration)

The main workspace shows a patch diagram with a 'clock' block and a 'count' block. The 'clock' block has inputs for EN (boolean), IVAL (number), and RST (pulse), and an output for TICK (pulse). The 'count' block has inputs for STEP (pulse), INC (pulse), and RST (pulse), and an output for watch (pulse).

The Inspector shows the configuration for the 'clock' patch:

- EN: True
- IVAL:
- RST: Never
- TICK: pulse

The Inspector also shows the Label and Description fields.



New Patch Add Library

2 Project Browser: Buttons

3 Project Browser: Project Patches

4 Project Browser: Libraries

5 Inspector

1 Your Patch

The screenshot shows the XOD Maker interface with several numbered callouts:

- 2**: Project Browser Buttons (New Patch, Add Library, Filter, Menu)
- 3**: Project Browser: Project Patches (List of patches: 001-hello, 002-simulate, 003-inspector, 004-patching)
- 4**: Project Browser: Libraries (List of libraries: awgrover/adafruitneopixel, awgrover/convers, bradzilla84/neopixel, bradzilla84/visi-genie-extra-library, cesars/0-all-examples, cesars/i2c-scanner)
- 5**: Inspector (Properties for the 'clock' patch: EN: True, IVAL: 1, RST: Never, TICK: pulse)
- 1**: Your Patch (Main workspace showing a 'clock' patch connected to a 'count' patch, with a 'watch' component below)
- 6**: Quick Help (Panel on the right showing details for the 'clock' patch: xod/core/clock, Outputs pulses at regular intervals, Inputs: EN (boolean), IVAL (number), RST (pulse), Outputs: TICK (pulse))

The main workspace also contains a 'Welcome to XOD, Maker!' message and an 'Exercise' section with instructions on navigating the project.

6 Quick Help



New Patch Add Library

1 Your Patch

2 Project Browser: Buttons

3 Project Browser: Project Patches

4 Project Browser: Libraries

5 Inspector

The screenshot shows the XOD IDE interface with several numbered callouts:

- 1**: Points to the main workspace area containing a visual patch diagram with blocks like 'clock', 'count', and 'watch'.
- 2**: Points to the top navigation bar.
- 3**: Points to the Project Browser on the left side.
- 4**: Points to the Project Browser showing a list of libraries.
- 5**: Points to the Inspector panel on the left, showing the configuration for the 'clock' patch.
- 6**: Points to the Quick Help panel on the right, showing details for the 'clock' patch.
- 7**: Points to the bottom toolbar containing upload and debug buttons.

6 Quick Help

Upload And Debug

7 Upload Buttons





New Patch Add Library

1 Your Patch

2 Project Browser: Buttons

3 Project Browser: Project Patches

4 Project Browser: Libraries

5 Inspector

The screenshot shows the XOD Maker interface with several numbered callouts:

- 1**: Points to the main workspace area containing a visual patch diagram with blocks like 'clock', 'count', and 'watch'.
- 2**: Points to the top navigation bar.
- 3**: Points to the Project Browser on the left side.
- 4**: Points to the Project Browser's library list.
- 5**: Points to the Inspector panel on the left, showing the configuration for the 'clock' patch.
- 6**: Points to the Quick Help panel on the right side.
- 7**: Points to the bottom right corner of the workspace, where the upload and debug buttons are located.

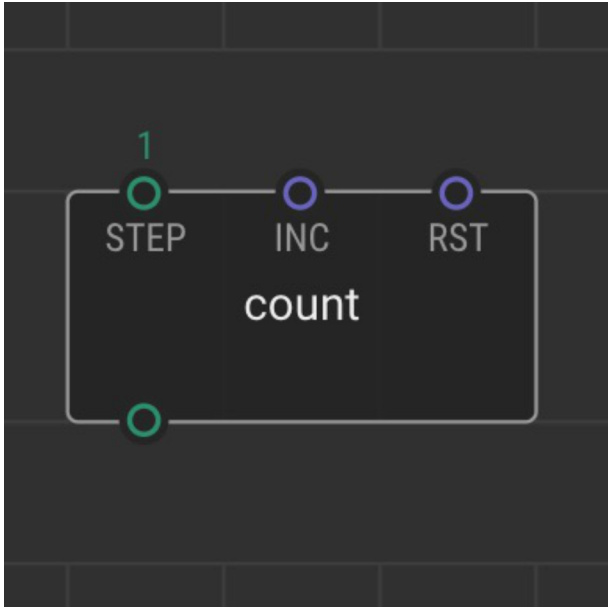
6 Quick Help

7 Upload Buttons

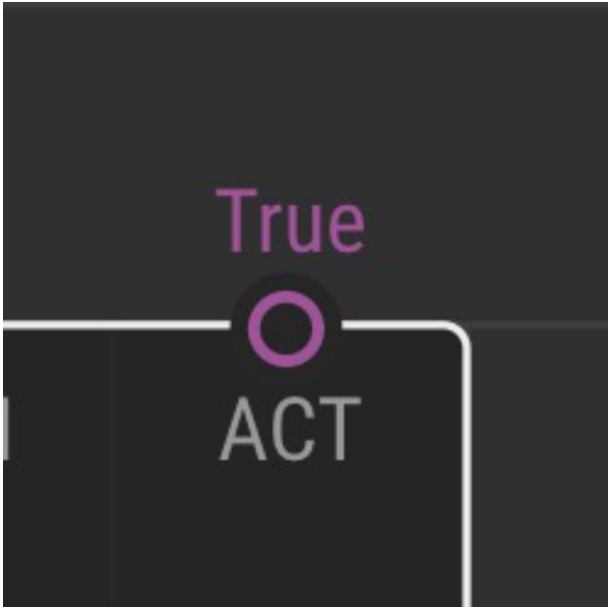
Upload And Debug



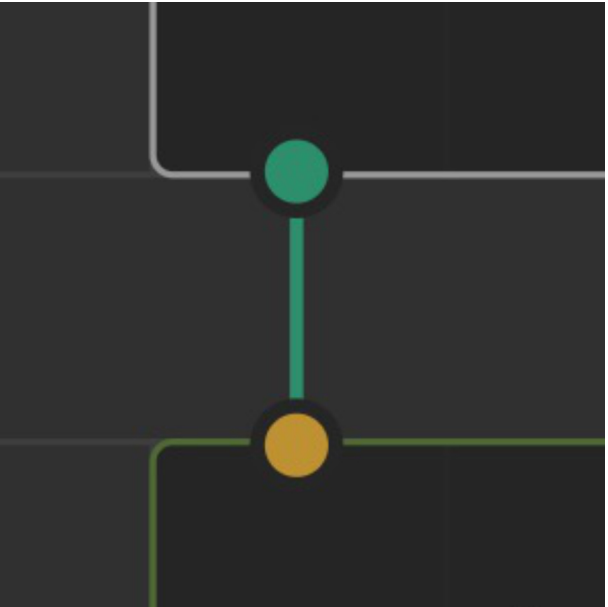
Nodes



Pins



Links



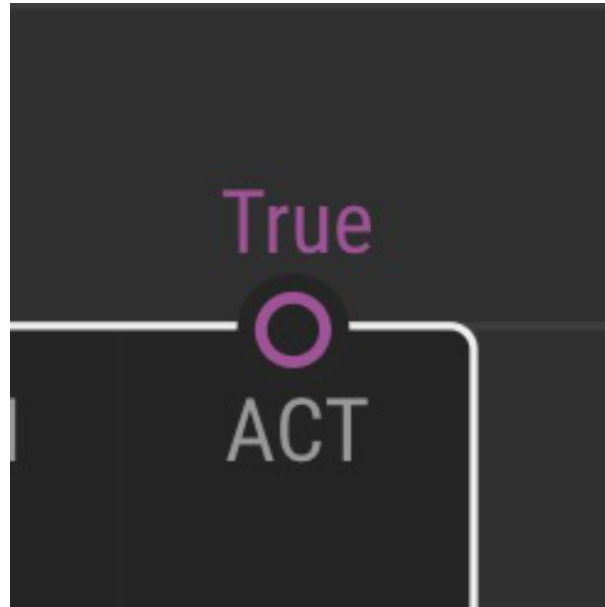
Pulse

String

Pins

Boolean

Port



Number

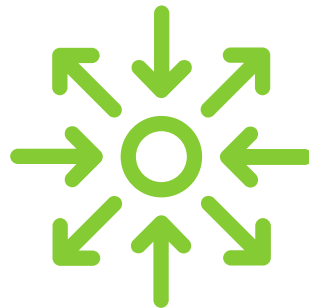
Byte

Break
30min

Break
20min



Testing Your Board



Inputs and Outputs

Breakout Groups – 35min

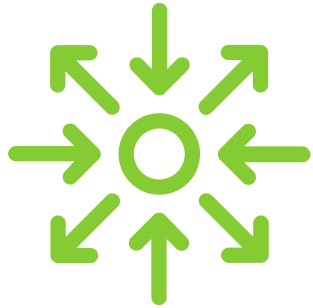
- 1 Introduce yourselves
- 2 Work together and see if each of you can complete the tasks
- 3 Step-by-step instructions are in the Guide (**p20-29**)
- 4 Use the 'Ask for Help' button if necessary



Testing Your Board

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



Inputs and Outputs

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

Congratulations!

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

Homework Challenge!

- 1 How can you expand on your simple programme?
- 2 Can you use the potentiometer to turn the buzzer on?
- 3 Can you make the buzzer turn on and the LED turn off when the button is pressed?
- 4 Can you get your light to flash?
- 5 Work through Lesson 3 in the Guide (**p32-45**)

Next Week

17:00 Welcome and Recap

17:05 Lesson 3: Explore XOD

Get to grips with some of the most useful nodes in XOD

17:25 Lesson 4: Building Devices (hands-on session)

Learn how to make your own XOD nodes and use the inbuilt OLED screen

17:55 Mini-Challenge (breakout groups)

What is the most interesting thing you can build with your Grove board

16:20 Round-up

Next Week

17:00 Welcome and Recap

17:05 Lesson 3: Explore XOD

Get to grips with some of the most useful nodes in XOD

17:25 Lesson 4: Building Devices (hands-on session)

Learn how to tidy and simplify complex programmes by building new nodes and using buses, then use these new skills to programme the inbuilt OLED display screen

16:25 Round-up

Today's Session

17:00 Welcome and Recap

17:05 Lesson 3: Explore XOD

Get to grips with some of the most useful nodes in XOD

17:25 Lesson 4: Building Devices (hands-on session)

Learn how to make your own XOD nodes and use the inbuilt OLED screen

17:55 Mini-Challenge (breakout groups)

What is the most interesting thing you can build with your Grove board?

16:20 Round-up

Today's Session

17:00 Welcome and Recap

17:05 Lesson 3: Explore XOD

Get to grips with some of the most useful nodes in XOD

17:25 Lesson 4: Building Devices (hands-on session)

Learn how to tidy and simplify complex programmes by building new nodes and using buses, then use these new skills to programme the inbuilt OLED display screen

16:25 Round-up

Last Week's Session

- 1** The Grove Board (**p6-7**)
- 2** The Microcontroller (**p8-11**)
- 3** The XOD IDE (**p12-15**)
- 4** Turned the LED on using the button (**p20-25**)
- 5** Controlled the buzzer using the button and potentiometer (**p26-29**)

Project Browser

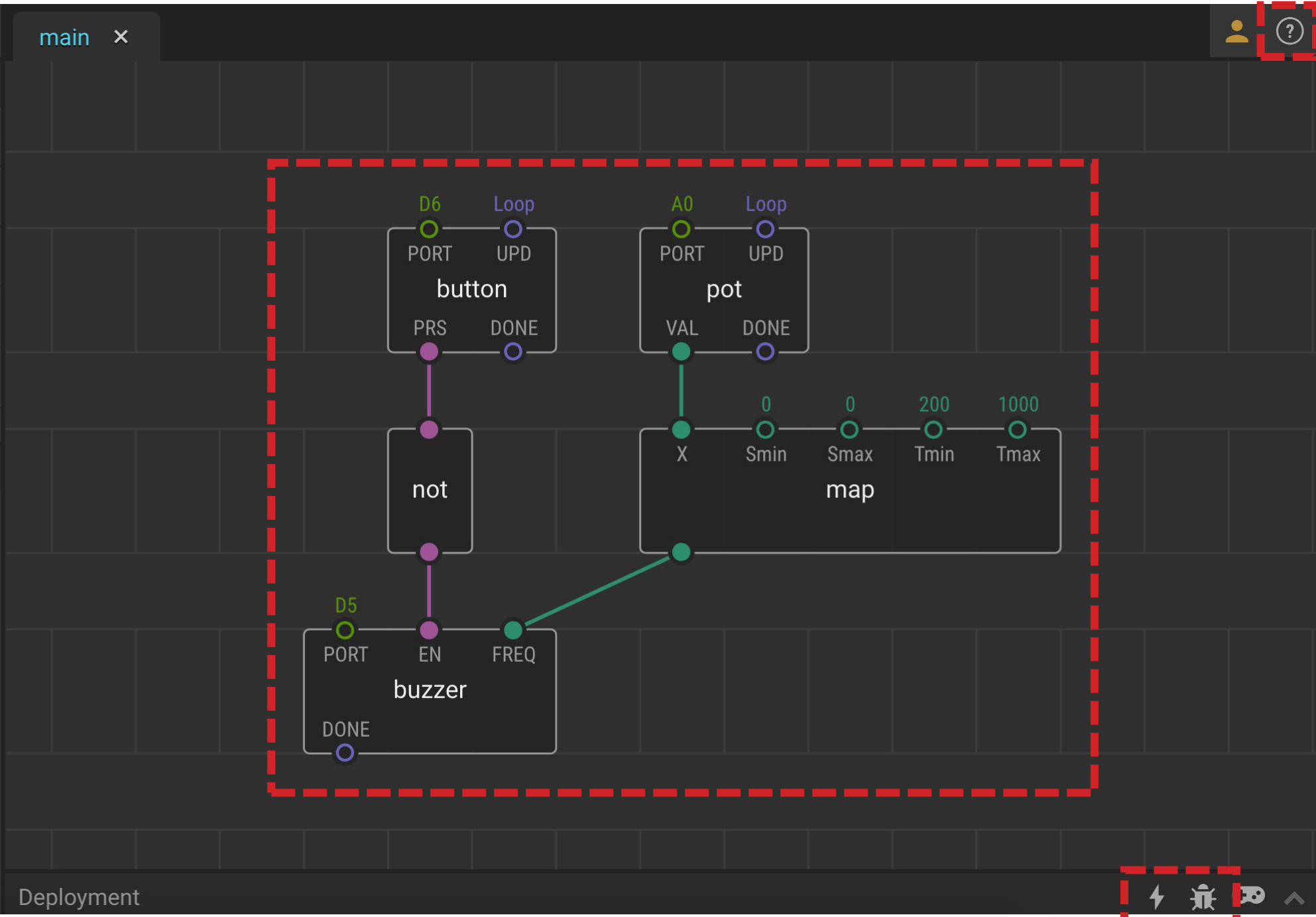
- My Project
- main
- awgrover/adafruitneopixel
- awgrover/conversions
- bradzilla84/neopixel
- bradzilla84/visi-genie-extra

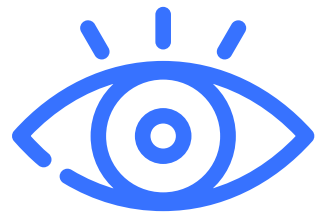
Inspector

main

Description

Deployment





Tweak and Watch Nodes

Boole

Pulse

String

Byte

Colour

Num

watch





main x



Project Browser



My Project

main

awgrover/adafruitneopixel

awgrover/conversions

bradzilla84/neopixel

bradzilla84/visi-genie-extra-

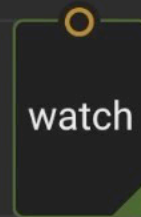
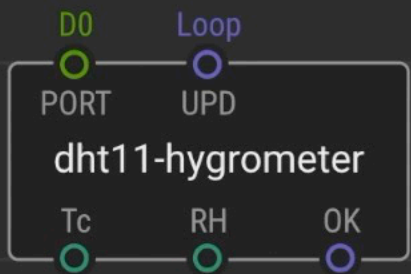
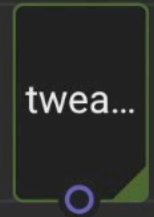
pector

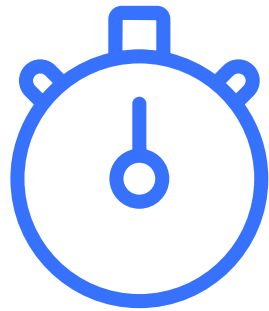
ain

scription

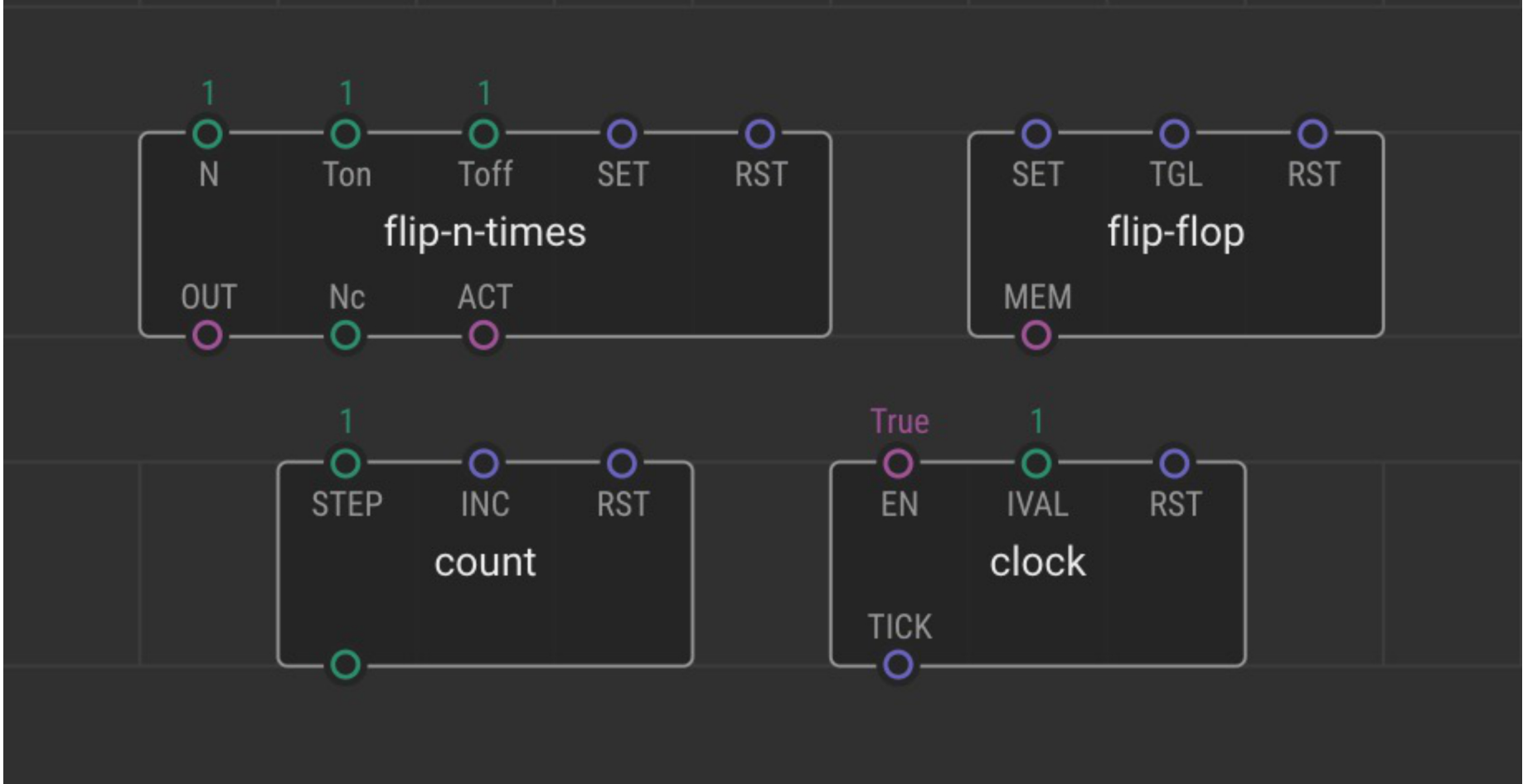
[Video Link](#)

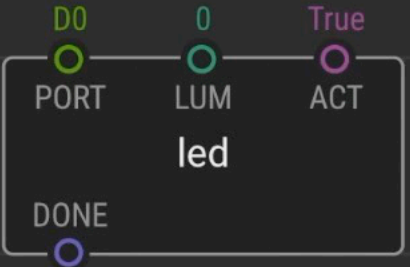
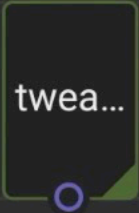
Deployment

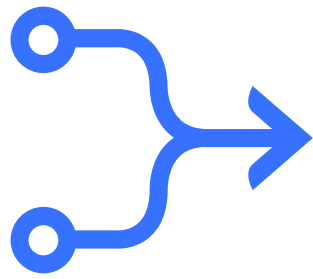




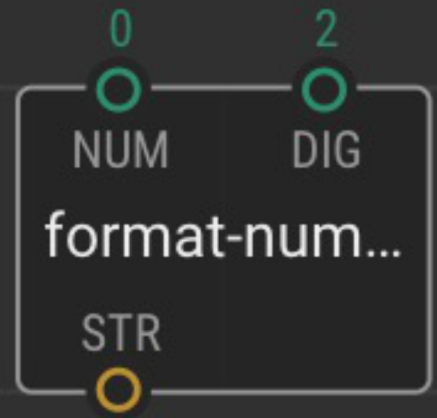
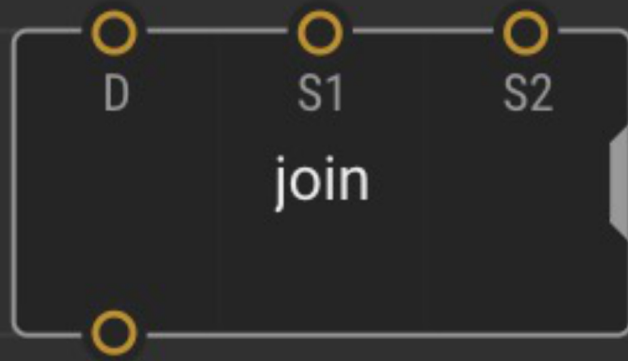
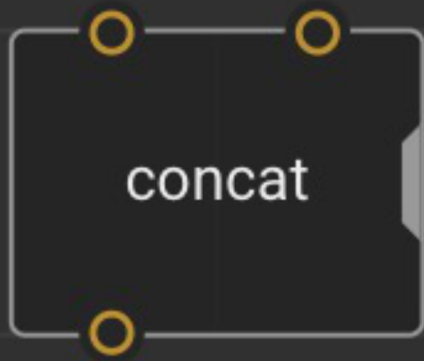
Flip, Clock and Count Nodes







Concat, Join and Format-Number Nodes

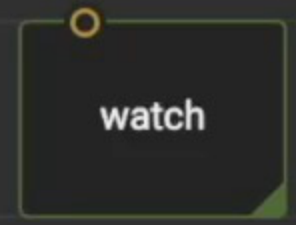
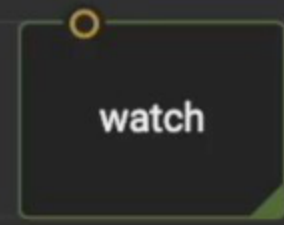
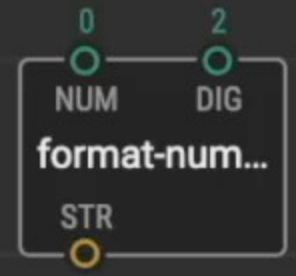
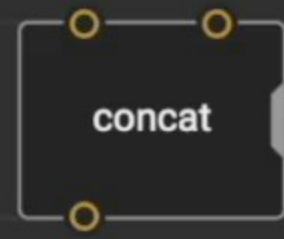
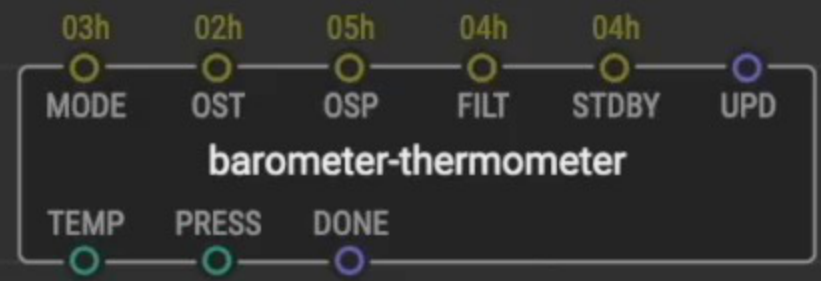


Project Browser

- My Project
 - main
- antoniorg/tcs34725
- awgrover/adafruitneopixel
- awgrover/conversions
 - data-to-pulse

Inspector

main x



main

Description

[Video Link](#)

Deployment

Lunch
60min

Afternoon Session

13:00 Lesson 4: Building Devices

Learn how to make more complex programmes in XOD using logic nodes, sequences and loops.

15:00 Break

15:30 Lesson 5: Next Steps

Learn how to expand your programming and hardware building capabilities to start building your own devices, and take a look at some previous projects.

16:25 Round-Up

Next Week's Session

13:00 Welcome, Recap & Lesson 4: Building Devices

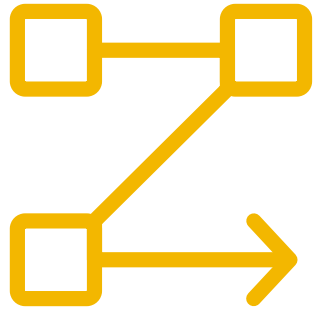
Learn how to make more complex programmes in XOD using logic nodes, sequences and loops.

14:00 Break

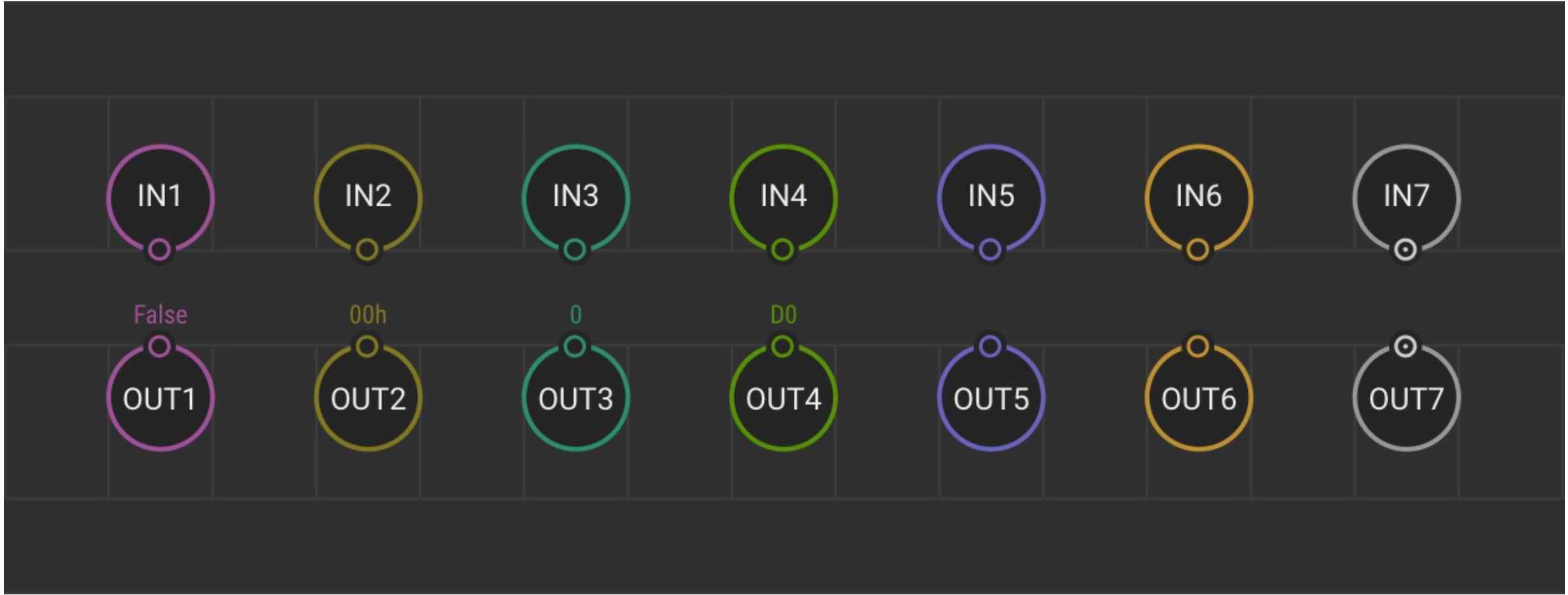
14:20 Lesson 4 cont. & Lesson 5: Next Steps

Learn how to expand your programming and hardware building capabilities to start building your own devices, and take a look at some previous projects.

15:55 Round-Up



Creating New Nodes



Try it Yourself – 15min

- 1 Work through Task 6 in groups
- 2 Step-by-step instructions are in the Guide (p50-54)
- 3 Ask if you need help



The Challenge!

Breakout Rooms and Idea Session

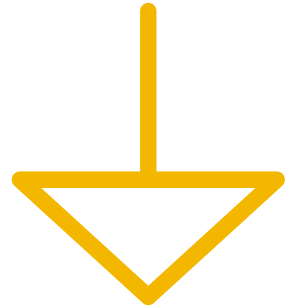
- 1 Meet your new group and introduce yourselves
- 2 Work through Task 6 (p50-54)
- 3 Discuss ideas – what is the most interesting thing you can do with the OLED screen and other onboard devices?
- 4 Make a list – what things would you need to be able to make a device of your choice?

Breakout Groups – 30min

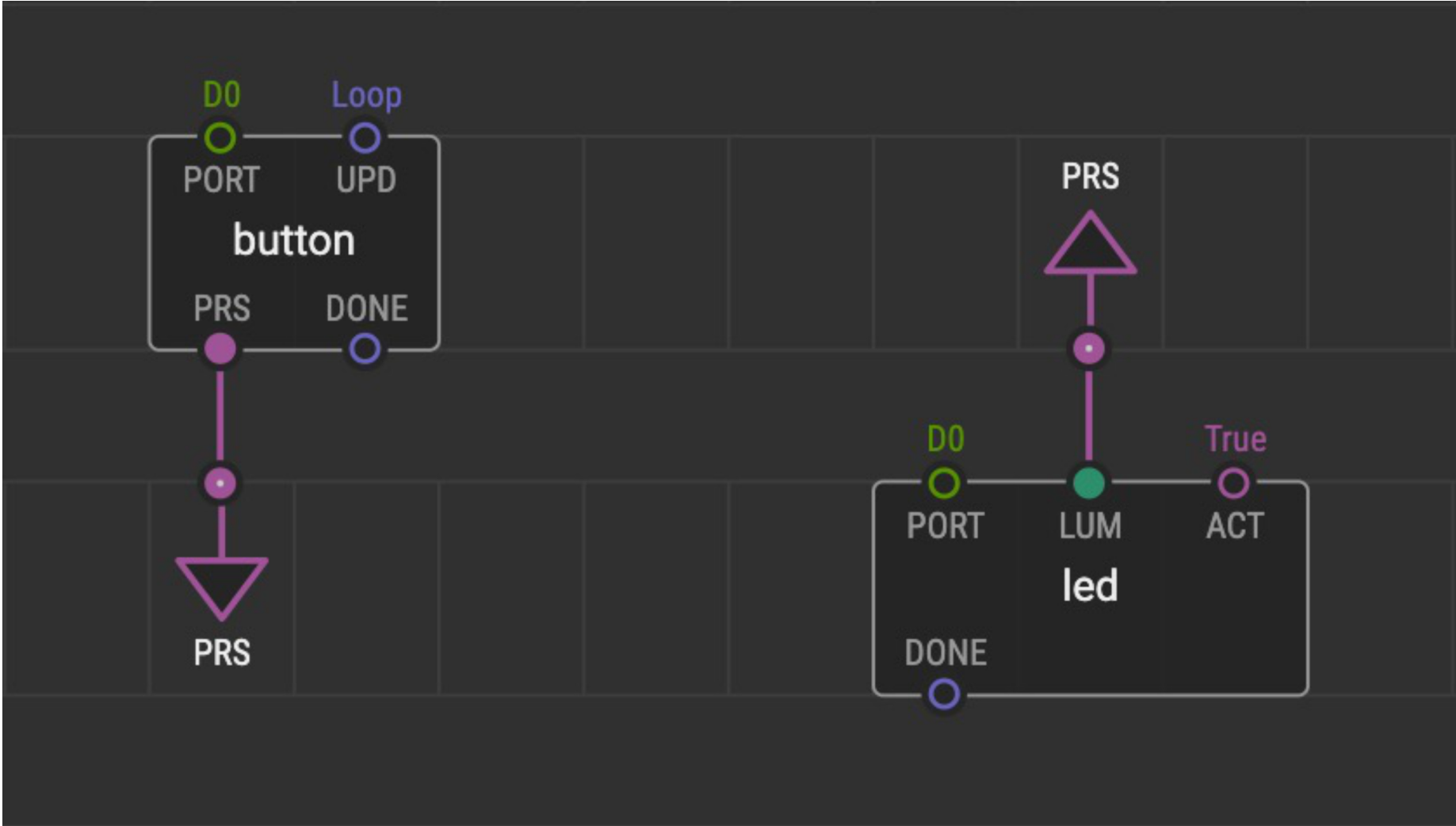
- 1 Meet your new group and introduce yourselves
- 2 Work through Task 6 together
- 3 Step-by-step instructions are in the Guide (**p50-54**)
- 4 Use the 'Ask for Help' button if necessary

Try it Yourself – 15min

- 1 Work through Task 6 in groups
- 2 Step-by-step instructions are in the Guide (p50-54)
- 3 Ask if you need help



Using Buses



Try it Yourself – 5min

- 1 Modify your *write-text-to-oled* node so that the *ssd1306-oled-i2c-device* DEV pin is linked to each of the other DEV pins by a bus rather than by links.
- 2 There is an example of this on **p57** of the guide.
- 3 Ask if you need help

(Task 7 of the guide provides another task to practice this if you have time)

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

Break
30min

Homework Challenge!

- 1 How can you use this new node in other ways?
- 2 What is the most interesting device you can build using the OLED screen and onboard devices?
- 3 What other nodes might you want to build?
- 4 Work through Lesson 7 in the Guide (**p56-59**)

Next Week

17:00 Welcome and Recap

17:05 Lesson 4: Building Devices (hands-on session)

Learn how to make more complex programmes in XOD using logic nodes, sequences and loops.

16:05 Lesson 5: Next Steps

Learn how to expand your programming and hardware building capabilities to start building your own devices, and take a look at some previous projects.

16:25 Round-up

Today's Session

17:00 Welcome and Recap

17:05 Lesson 4: Building Devices (hands-on session)

Learn how to make more complex programmes in XOD using logic nodes, sequences and loops.

16:05 Lesson 5: Next Steps

Learn how to expand your programming and hardware building capabilities to start building your own devices, and take a look at some previous projects.

16:25 Round-up

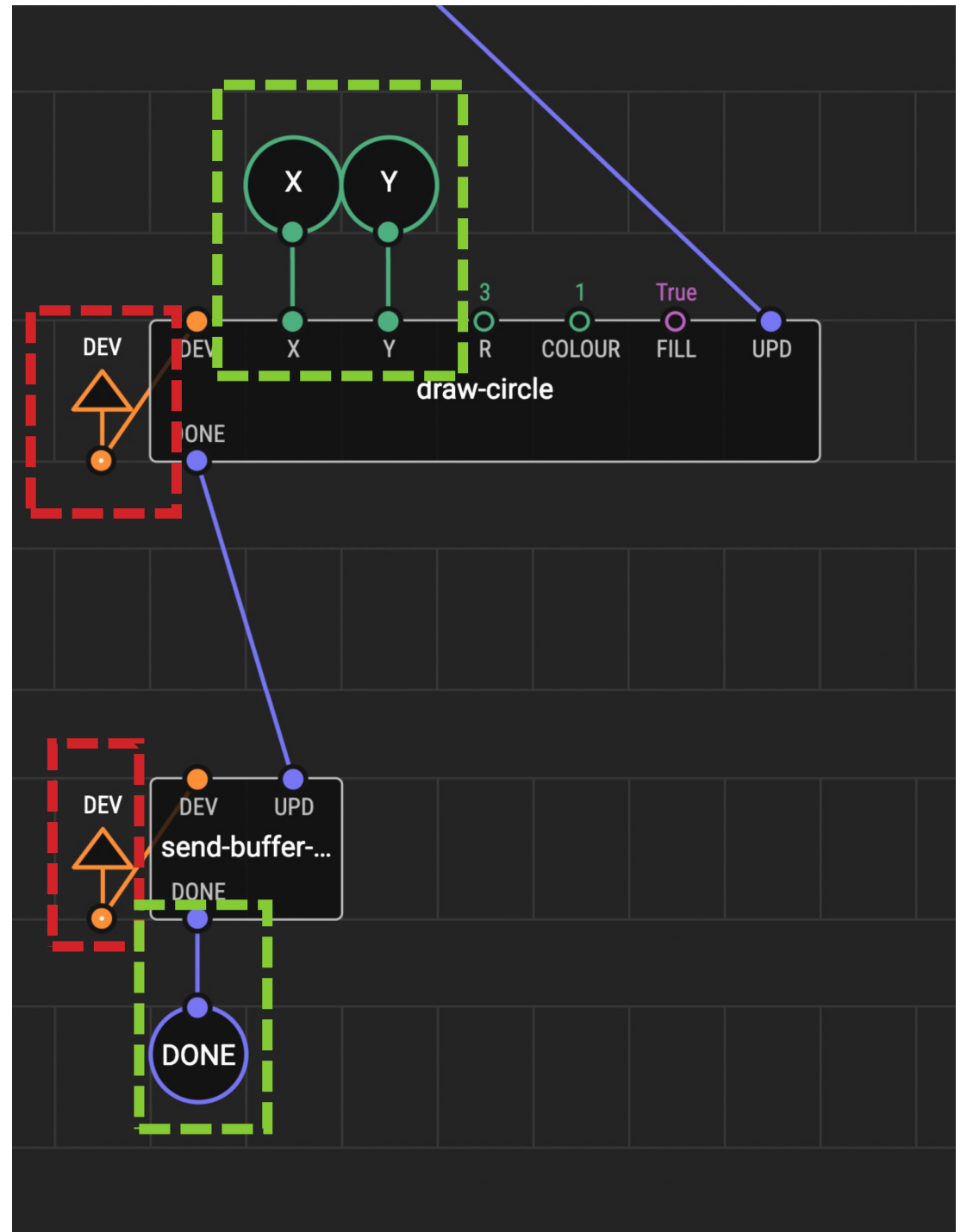
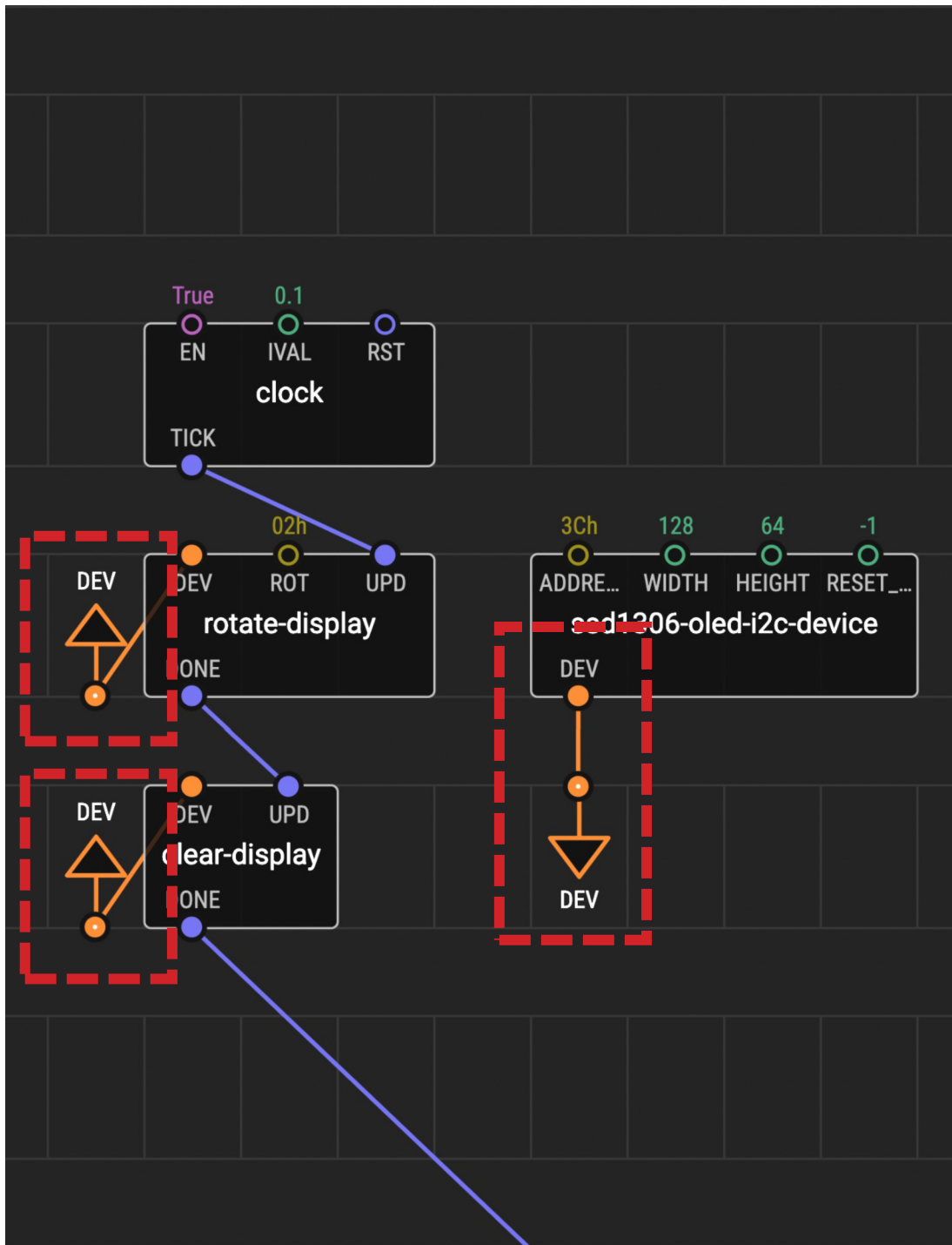
Last Week's Session

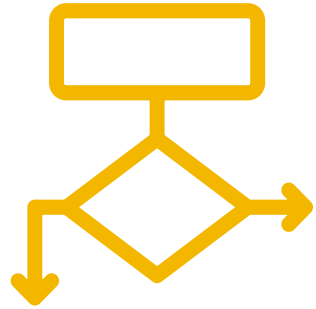
- 1 Learned about some useful nodes in XOD (**p31-45**)

Tweak, watch, flip, clock, count, concat, join and format-number

- 2 Made a new node to write text to the OLED screen (**p49-55**)

- 5 Learned how to use buses to simplify complex programmes (**p56-59**)

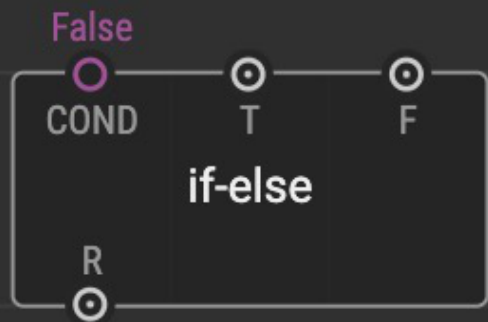
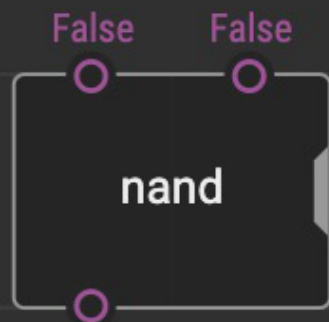
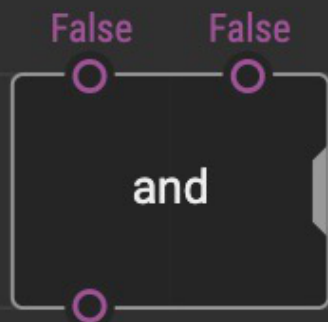
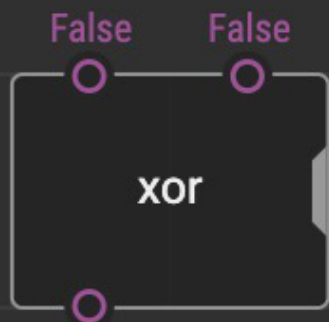
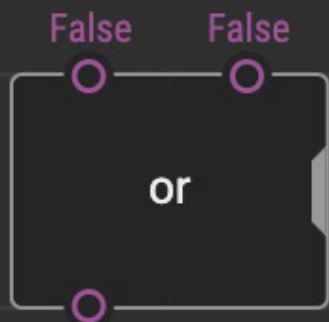
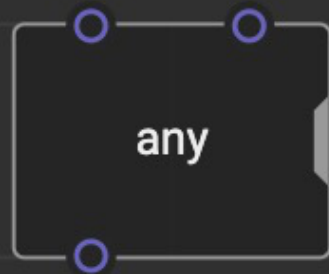
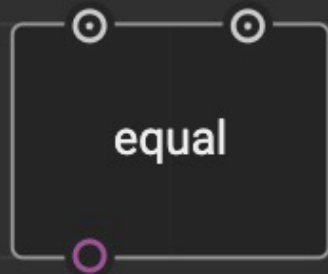
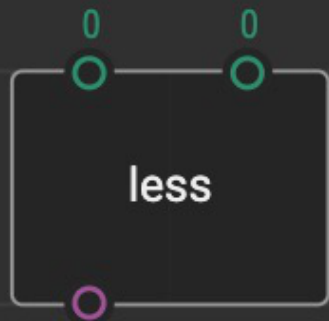
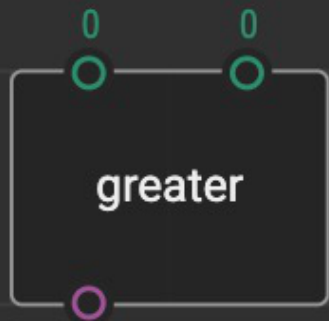


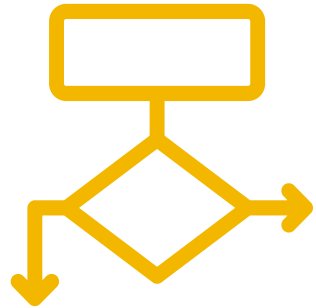


Logic Programmes

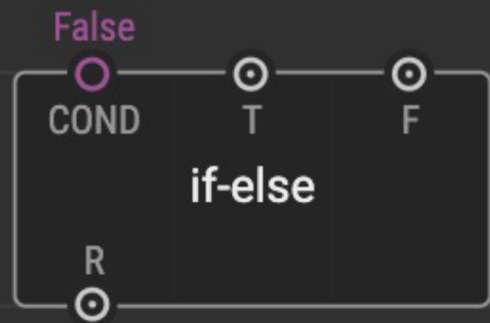
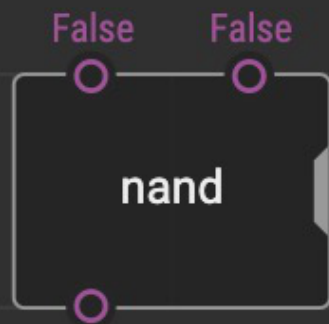
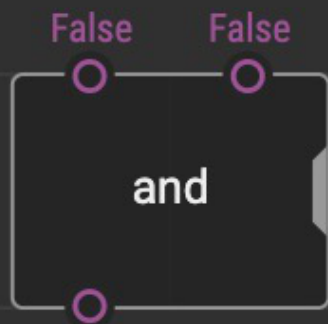
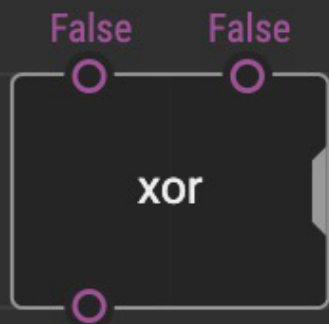
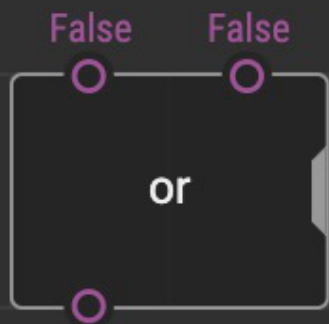
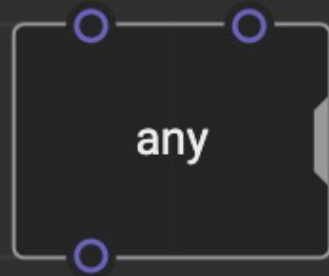
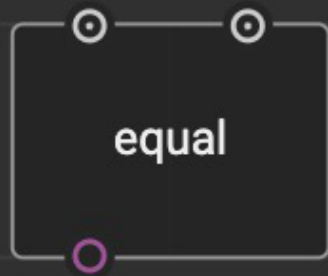
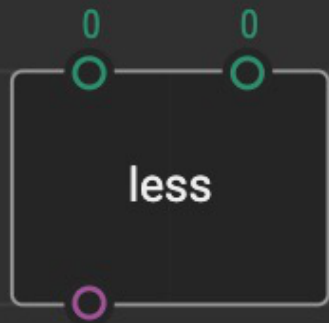
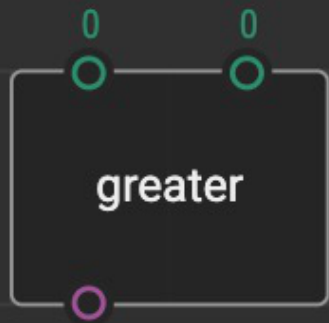


Sequences and Loops





Logic Programmes



Try it Yourself – 20min

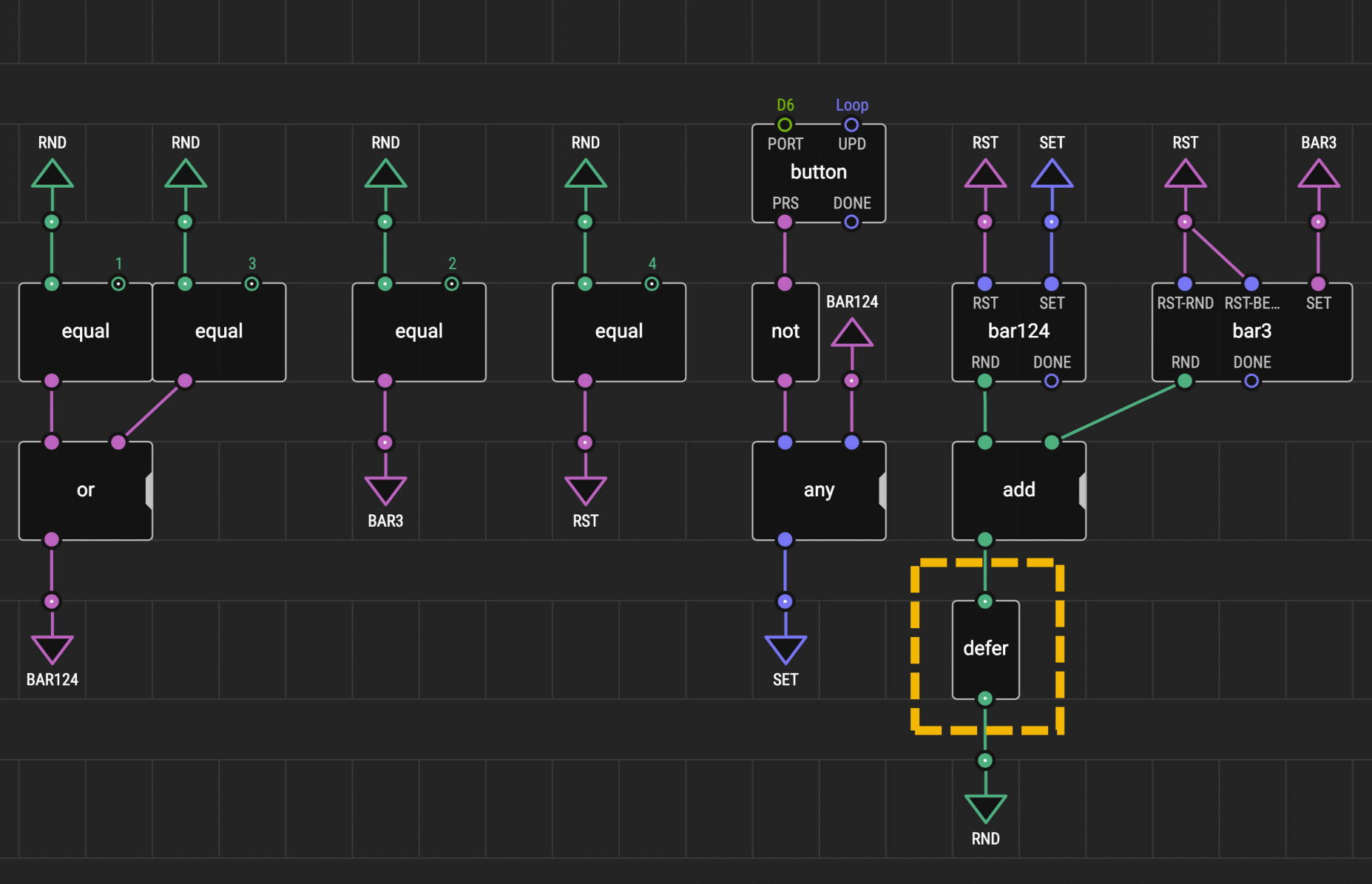
- 1 Work through Task 8 in groups
- 2 Step-by-step instructions are in the Guide (**p60-63**)
- 3 Ask if you need help

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



Sequences and Loops



Try it Yourself – 20min

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

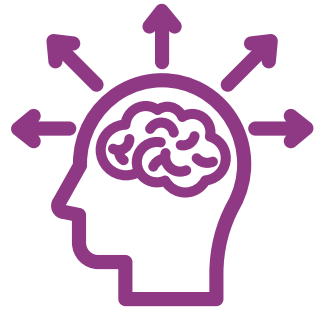
Try it Yourself – 15min

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

Breakout Groups – 35min

- 1 Meet your new group and introduce yourselves
- 2 Work through Tasks 8 and 9 together
- 3 Step-by-step instructions are in the Guide (**p60-70**)
- 4 Use the 'Ask for Help' button if necessary

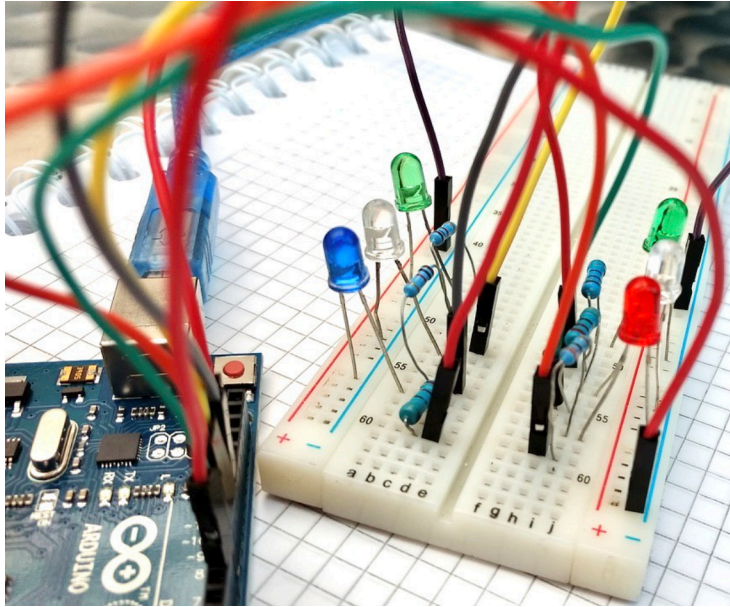
Break
30min



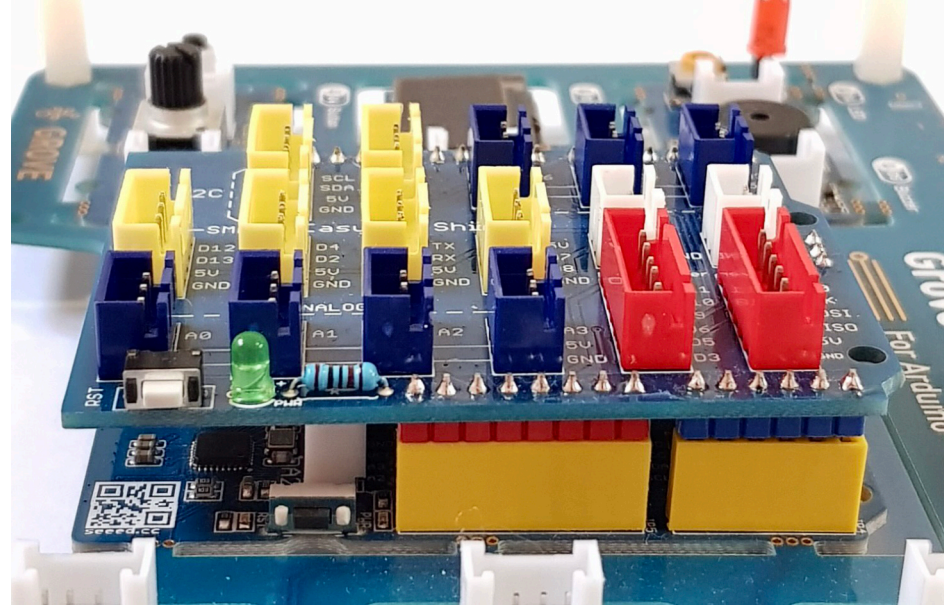
Expanding Your Capacity

Expanding Your Capacity

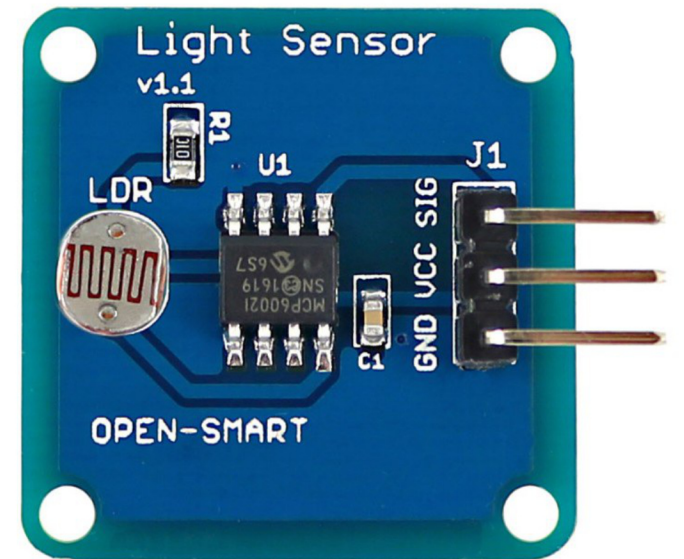
Wires



Shields



Breakout Boards



Plug-and-Play Components



www.seeedstudio.com
> Shop > Grove



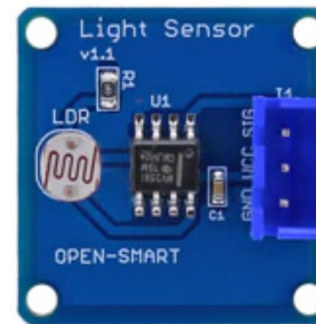
Plug directly into white sockets
on the board



www.m5stack.com
> Store > Unit



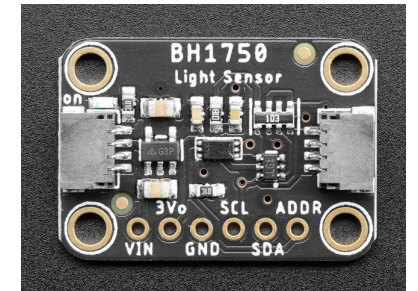
[open-smart](http://open-smart.aliexpress.com)
[.aliexpress.com](http://aliexpress.com)



Plug into Open Smart
Expansion Shield
(or use JST PH to
JST XH cables)



www.adafruit.com
> Products >
[STEMMA/STEMMA QT](#)

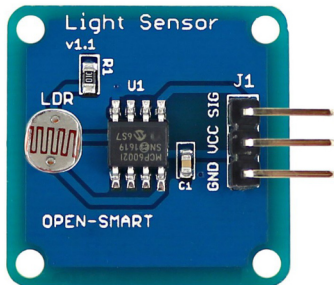


Plug directly
(STEMMA 4 pin)
Plug with JST PH to
JST SH cable
(STEMMA QT 4 pin)

Wired Breakout Boards

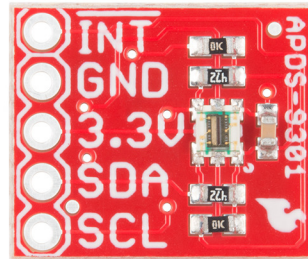
OPEN-SMART

[open-smart
.aliexpress.com](https://open-smart.aliexpress.com)



sparkfun[®]
ELECTRONICS

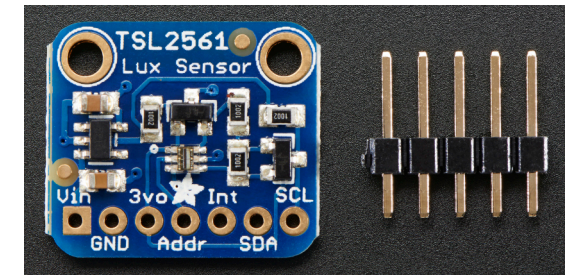
www.adafruit.com



Solder pins to board. Connect using Grove-to-female wires (make sure pin labels match up)

adafruit

www.adafruit.com



Finding XOD Nodes

Search using 'reference designator'
e.g. BMP280 (barometer) or SSD1306 (OLED screen)

www.xod.io/libs

forum.xod.io



Libraries

Sort by Updated ▾

emiliosancheza/bme280-sensor@1.0.1

First version for BME280 sensor. Based on Wayland BMP280-barometer
2020-09-18



wayland/bmp280-barometer@0.0.1

BMP280 barometric pressure and temperature sensor. Wraps
https://github.com/adafruit/Adafruit_BMP280_Library. Datasheet: https://aebst.resource.bosch.com/media/_tech/media/datasheets/BST-BMP280-DS001.pdf
2020-05-16



Do you want live notifications when people reply to your posts? [Enable Notifications](#)

7 results for **BMP280**

 Topics/posts ▾

[▶ Advanced filters](#)

Sort by



BMP280 to LCD 20x4

■ Hardware Interfacing Questions

Aug '20 - ...to XOD. I am trying to get bmp 280 to read and display temp, pressure on LCD but it will not keep reading "ERR". Maybe to do with the format of the **BMP280** output shows as number, I am guessing this is a INT. Please Help,



GY-BMP280 Temperature humidity and pressure sensor

Jan '20 - hi guys, does anyone know if this can be connected via xod and which one to use? thank you! /uploads/short-url/dk22h6ksZMKf1Coyt3GpxPJm8VG.jpeg 20200106_214337|281x500 20200106_214354 20200106_214409

Arduino IDE

Arduino provides its own free IDE software, which uses C++ coding language to programme the board.

www.arduino.cc/en/software



The screenshot shows the Arduino IDE interface with the 'Blink' example code loaded. The code is as follows:

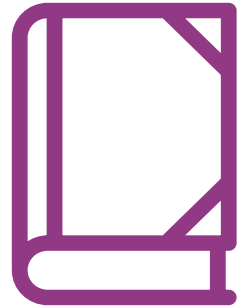
```
Blink | Arduino 1.8.5  
Blink §  
This example code is in the public domain.  
  
http://www.arduino.cc/en/Tutorial/Blink  
*/  
  
// the setup function runs once when you press reset or power the board  
void setup() {  
  // initialize digital pin LED_BUILTIN as an output.  
  pinMode(LED_BUILTIN, OUTPUT);  
}  
  
// the loop function runs over and over again forever  
void loop() {  
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000); // wait for a second  
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW  
  delay(1000); // wait for a second  
}
```

At the bottom of the window, it says '32' and 'Arduino/Genuino Uno on COM1'.

Converting Arduino libraries for use in XOD
bit.ly/arduino-to-xod

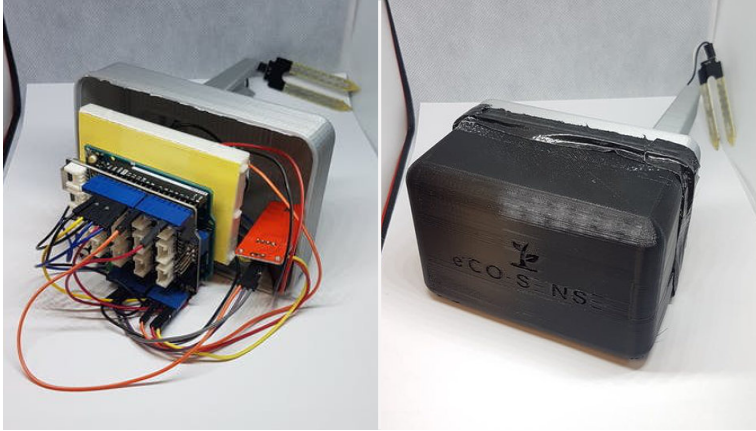
Combining XOD and Arduino IDE
XOD menu > Deploy > 'Show Code for Arduino'

More complex programming
www.arduino.cc/en/Tutorial/HomePage

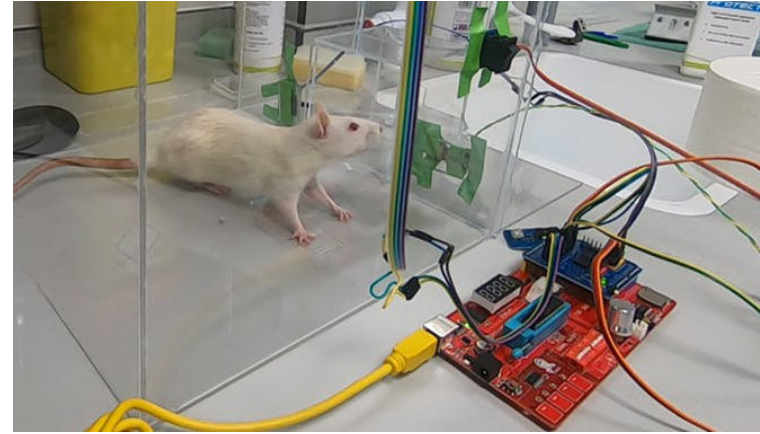


Case Studies

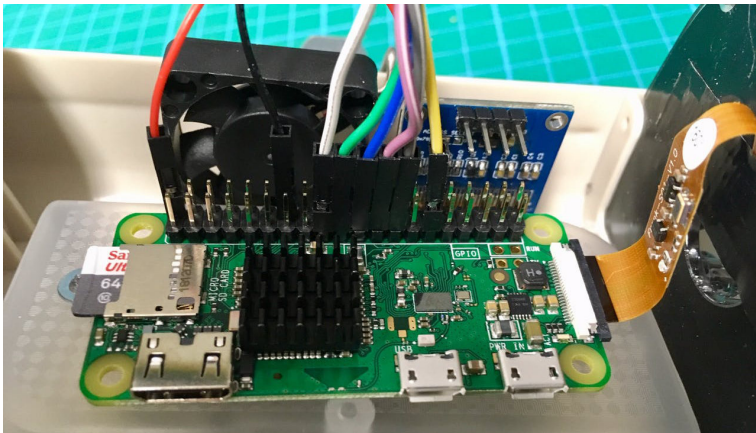
eCO-SENSE: Soil Sensors Powered by Plant Photosynthesis



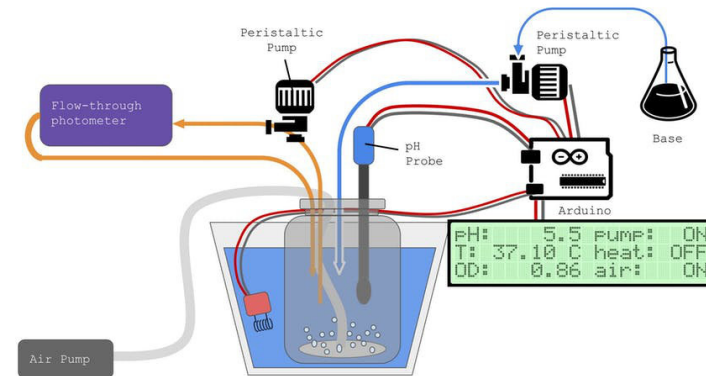
Behavioural Chamber to Evaluate Rodent Forelimb Grasping



Camera for Monitoring Plant Pollination Events



Open Source Microbial Bioreactor



Breakout Groups – 10min

- 1 In groups, pick and read through a case study (p80-83)
- 2 Discuss which case study you find most interesting
- 3 What instruments might be useful in your own research?
- 4 Think about how you might use what you've learned to build your own instrument – what additional hardware and programming skills would you need?

Discussion – 15min

- 1 Read through the case studies (p80-83)
- 2 Discuss in groups
- 3 Which of these devices is most relevant to your research?
- 4 What extra hardware or programming skills would you need to create one of these devices?



What would you build?

Discussion – 20min

- 1 What instruments would be useful in your own research?
- 2 How would you go about building such a device?
- 3 What additional hardware/programming would you need?
- 4 Do some research – has something like this already been done? Can you find the things you need?

Questions? Contact the Biomaker team: coordinator@synbio.cam.ac.uk

Discussion – 10min

- 1 What instruments would be useful in your own research?
- 2 How would you go about building such a device?
- 3 What additional hardware/programming would you need?
- 4 Do some research – has something like this already been done? Can you find the things you need?

Questions? Contact the Biomaker team: coordinator@synbio.cam.ac.uk

Thank You

More info:

www.biomaker.org

