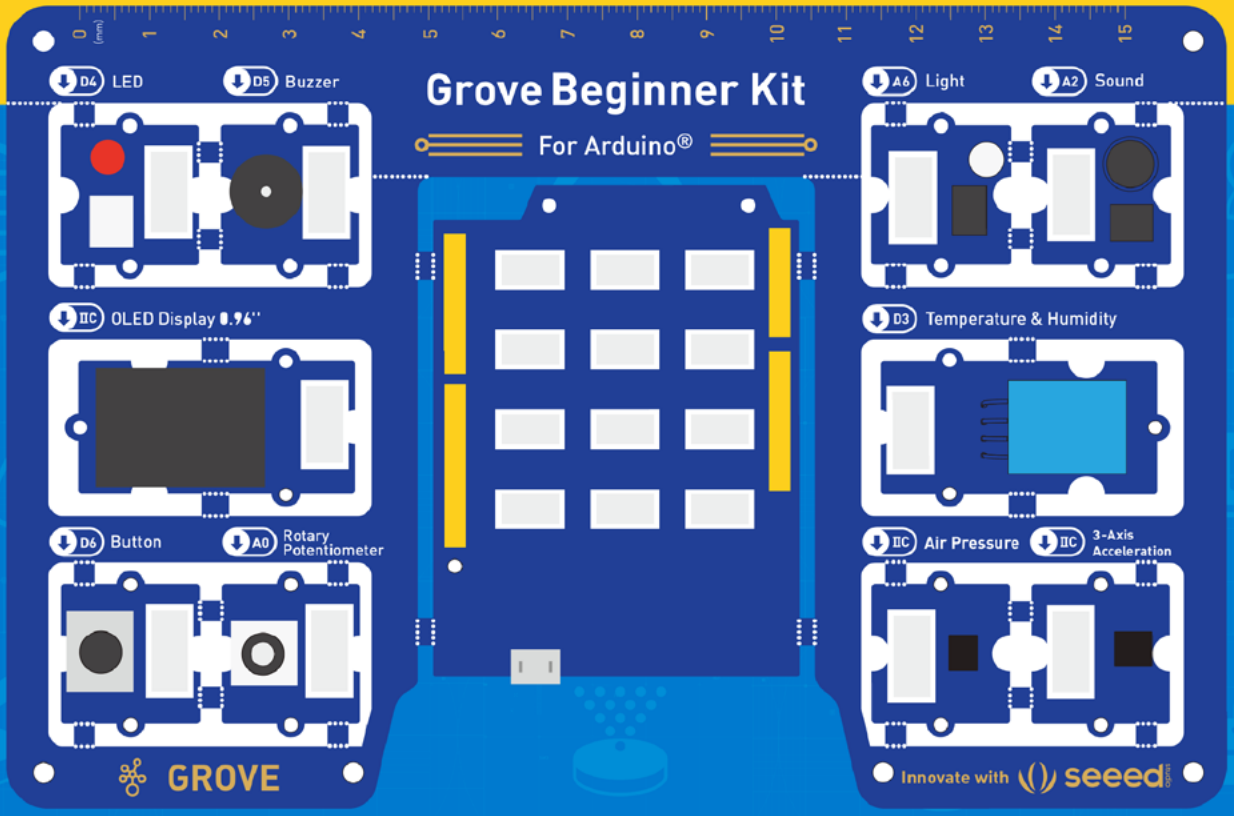
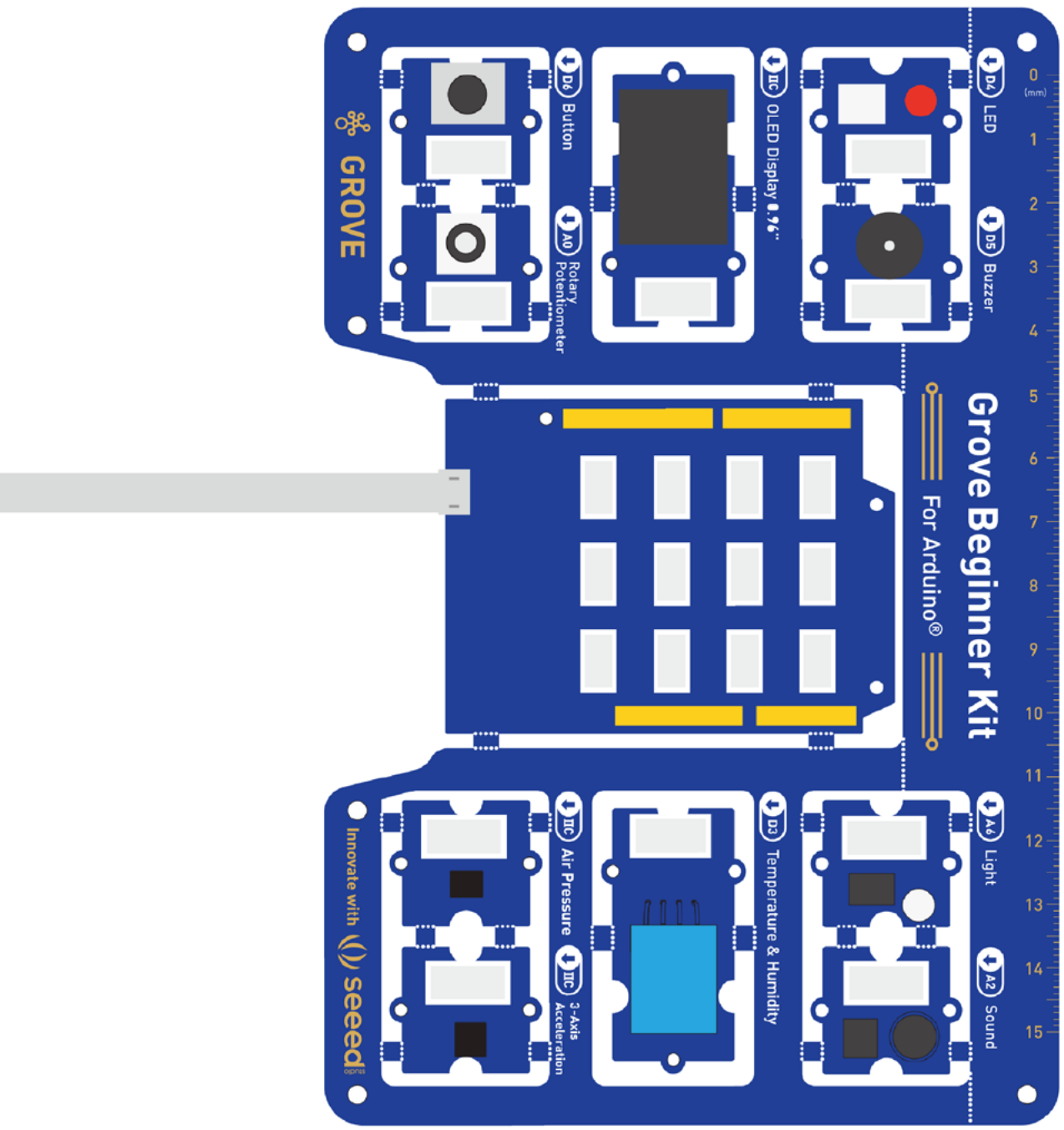
TinkerGen
STEM made simple

seeed studio

Grove Beginner Kit For Arduino

Codecraft Graphical Programming Course



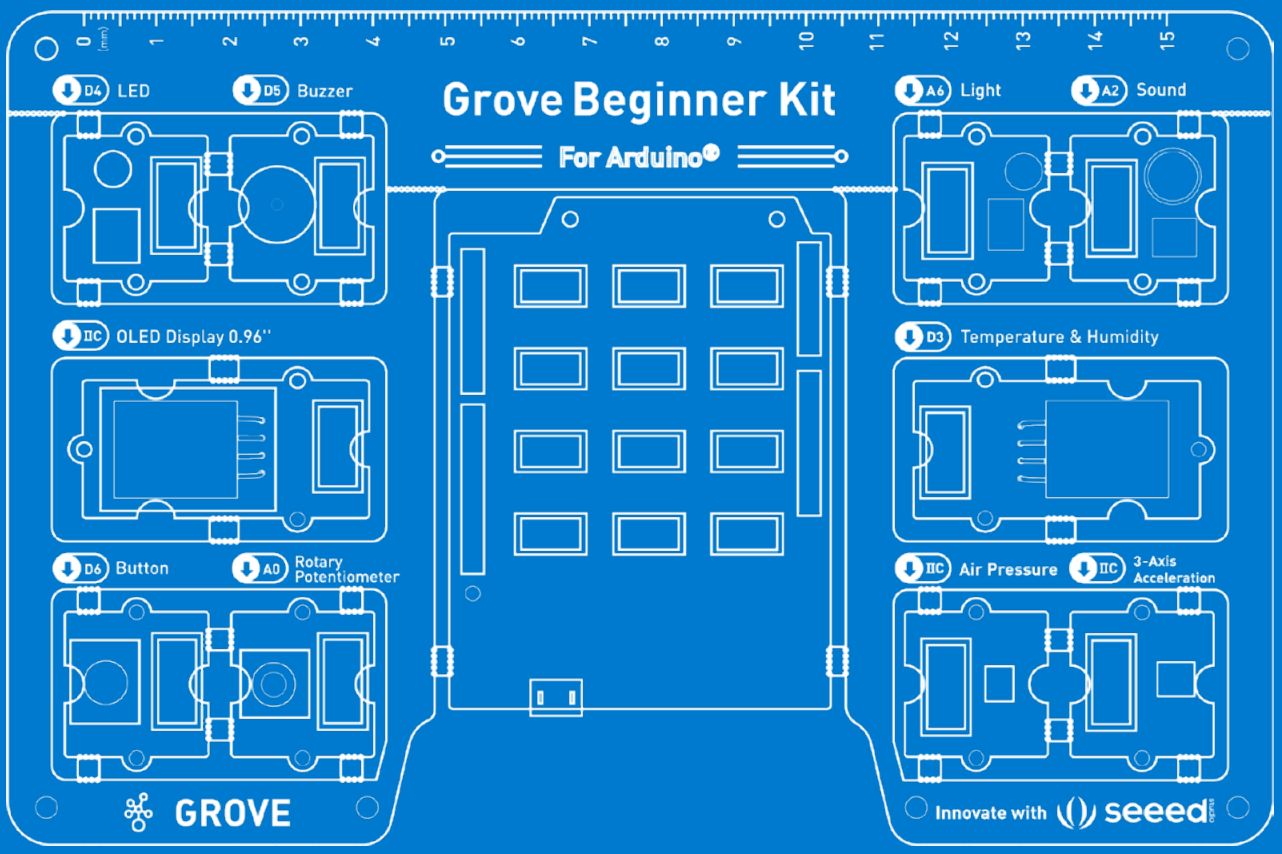


Proportion with real object 1:1

Contents

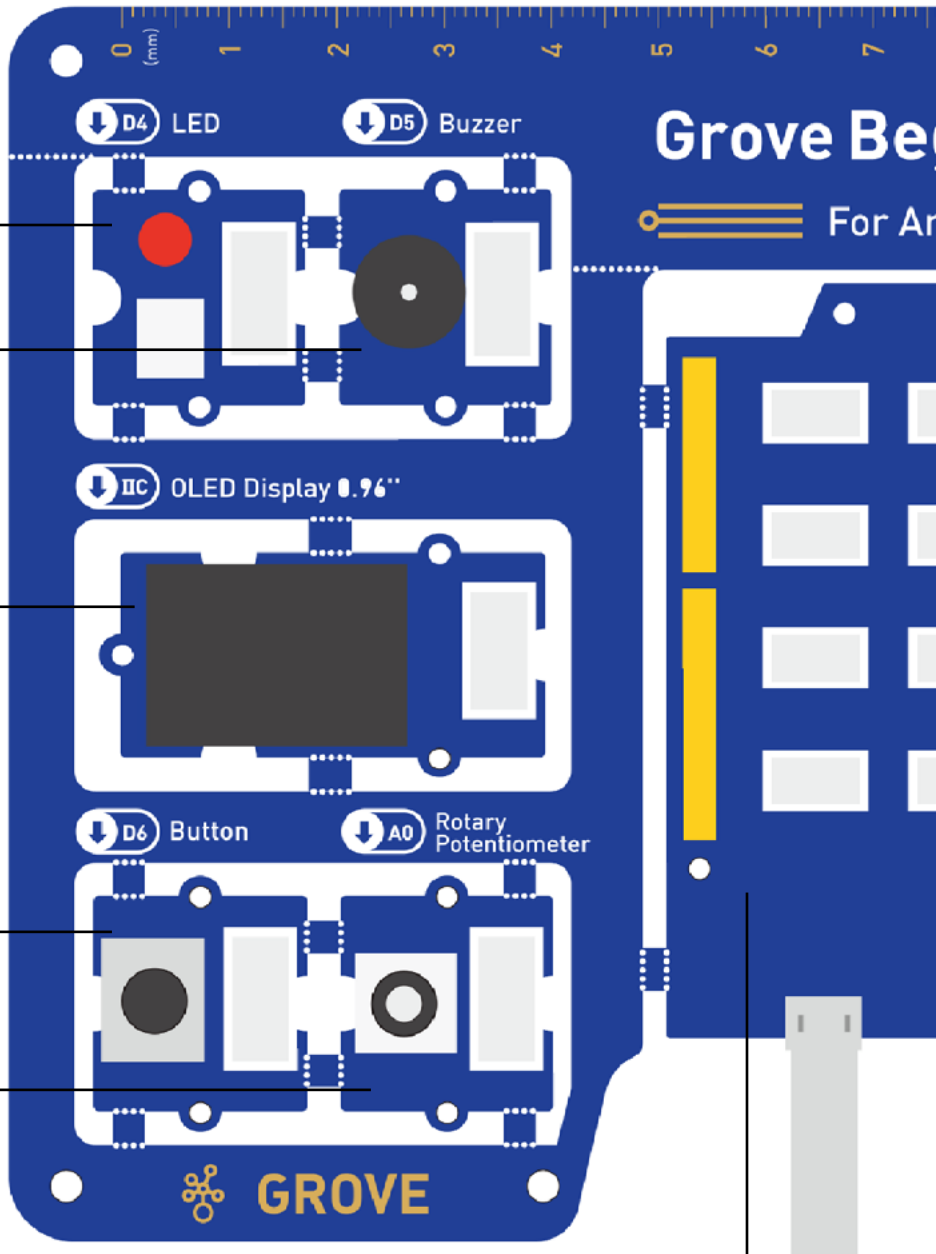
Preface

Welcome to Getting Started with Grove Beginner Kit!
Grove Beginner Kit for Arduino is one of the best Arduino Beginner Kit for beginners. It includes one Arduino compatible Board and 10 additional Arduino sensors and all in one-piece of PCB design. All the modules have been connected to the Seeeduino through the PCB stamp holes so no Grove cables are needed to connect. Of course, you can also take the modules out and use Grove cables to connect the modules. You can build any Arduino project you like with this Grove Beginner Kit For Arduino. This course will walk you through the basics of using Grove Beginner Kit with Codecraft, a graphical programming environment based on Scratch 3.0.



Hardware Overview

Dimensions - 17.69 * 11.64 * 1.88cm



Grove - LED: Simple LED module

Grove - Buzzer: Piezo Buzzer

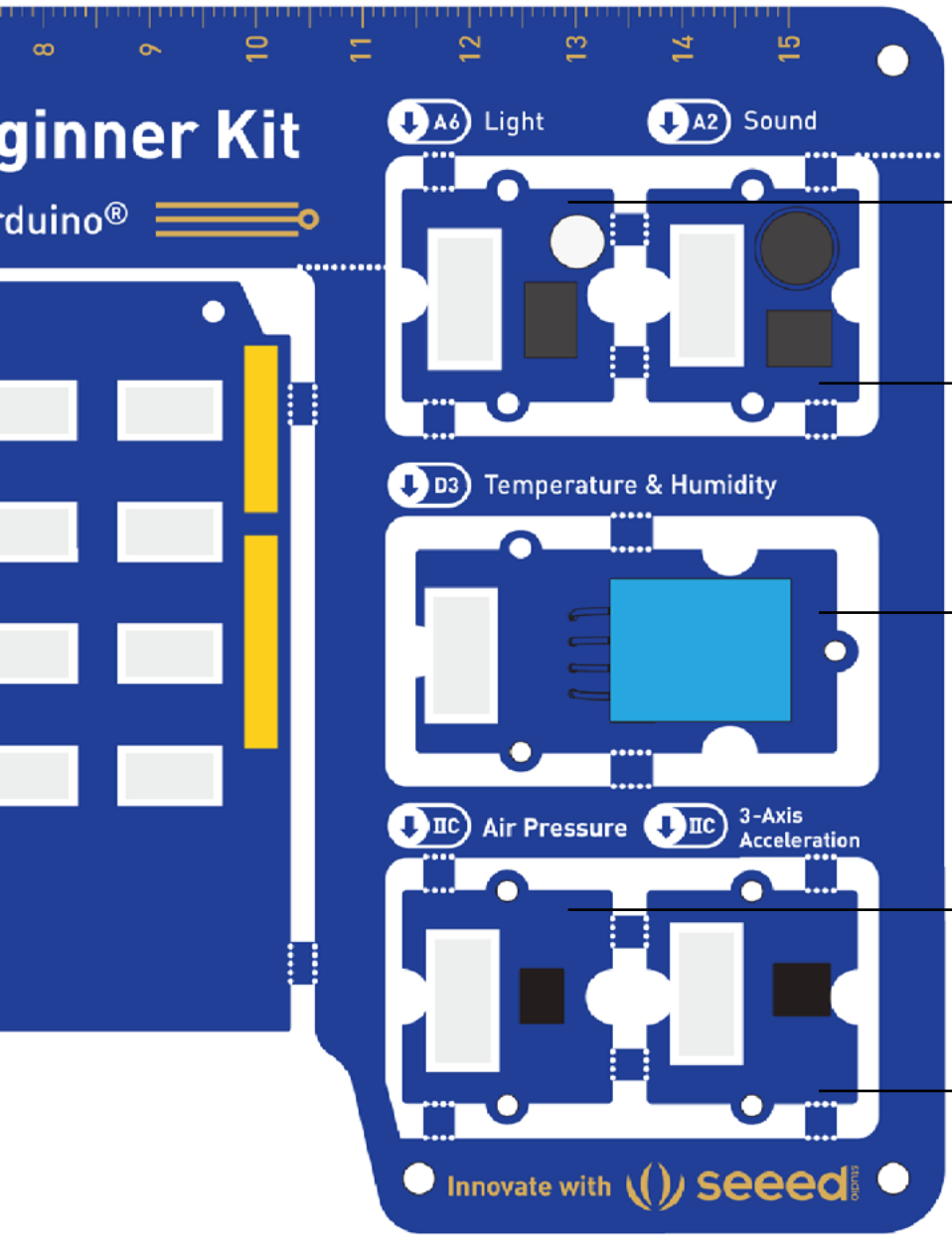
Grove - OLED Display 0.96": 128×64 dot resolution High brightness,self-emission and high contrast ratio Big screen on a compact design Low power consumption

Grove - Button: Momentary Push Button

Grove - Rotary Potentiometer: Adjustable Potentiometer

Seeeduino Lotus: Arduino Compatible Board with Grove Ports

For more details about the hardware, please visit the official document:
https://wiki.seeedstudio.com/Grove-Beginner-Kit-For-Arduino/



Grove - Light Sensor : Detects surrounding light intensity

Grove - Sound sensor: Detects surrounding sound intensity

Grove - Temperature & Humidity Sensor: Detects surrounding temperature and humidity values

Grove - Air Pressure Sensor: Detects surrounding atmospheric pressure

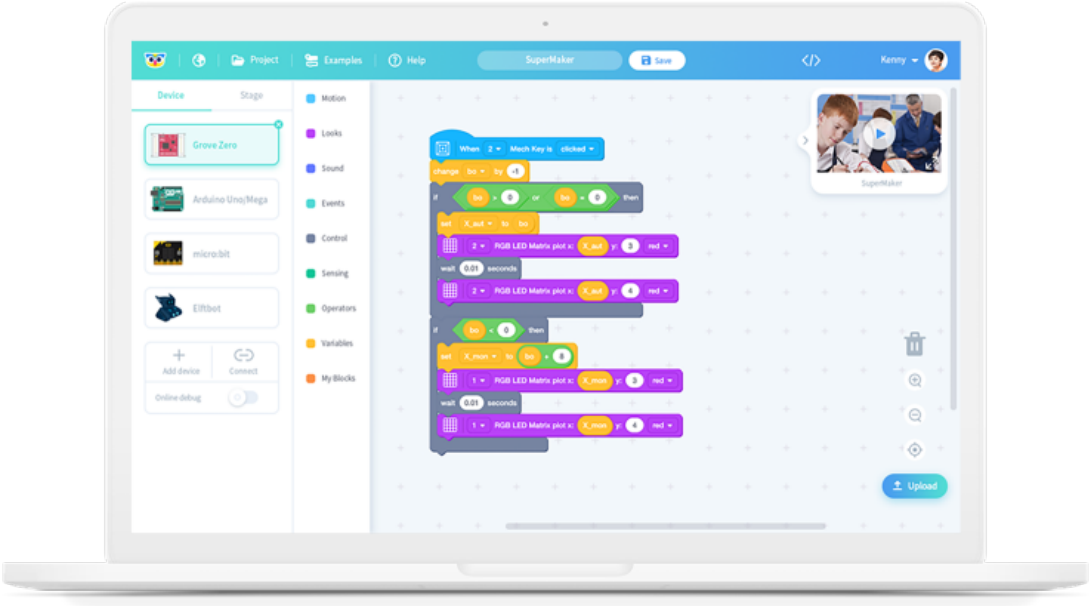
Grove - 3-Axis Accelerator: Detects object acceleration

Note

By default, Grove modules are connected to Seeeduino via PCB stamp holes. This means you don't need to use Grove cables to connect if not broken out. The default pins are as follow:

Modules	Interface	Pins/Address
LED	Digital	D4
Buzzer	Digital	D5
OLED Display 0.96"	I2C	I2C, 0x78(default)
Button	Digital	D6
Rotary Potentiometer	Analog	A0
Light Sensor	Analog	A6
Sound Sensor	Analog	A2
Temperature & Humidity Sensor	Digital	D3
Air Pressure Sensor	I2C	I2C, 0x77(default) / 0x76(optional)
3-Axis Accelerator	I2C	I2C, 0x19(default)

Prepare the programming environment



Codecraft is a graphical programming software suitable for kids ages 6-16, who are learning to program.

Codecraft is based on the Scratch 3.0 language and enables programming by simply "dragging and dropping" blocks. In addition to Scratch's ability to program interactive games or animations, Codecraft also supports a variety of common hardware devices, enabling hardware and software integration, which makes programming even more fun.

There are two ways to use Codecraft - from your web browser with Codecraft Web or if your computer is not connected to Internet, you can opt out for offline Codecraft PC Client.

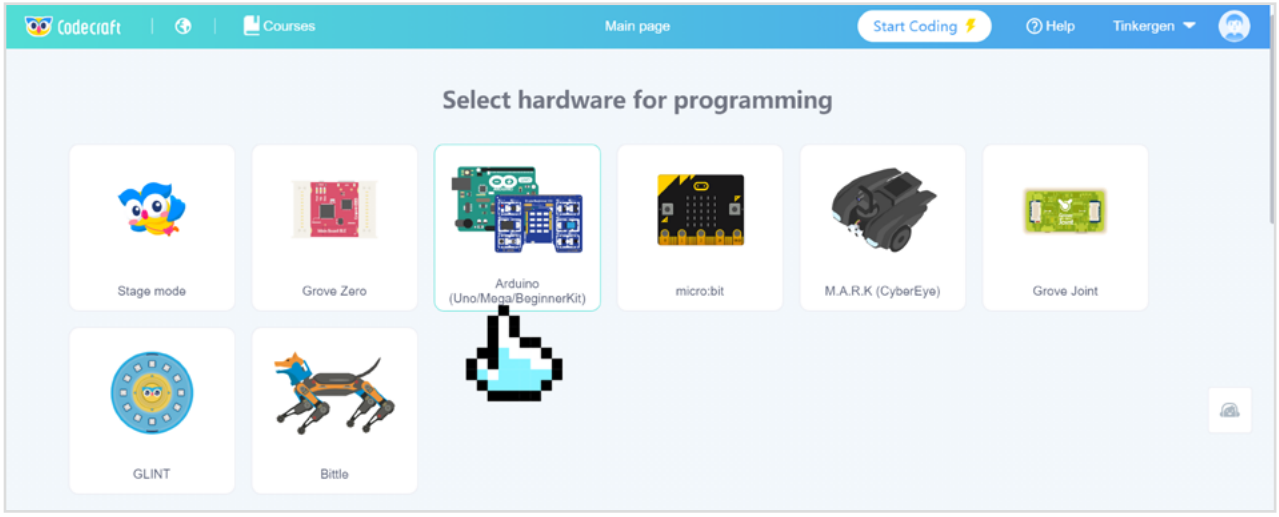
Codecraft Web

Codecraft is a web-based graphic programming software, and you can use it with a web browser, which provides a convenient and simple user experience.

Visit **ide.tinkergen.com** or click on the linkage below to start your creation with Codecraft now.

https://ide.tinkergen.com

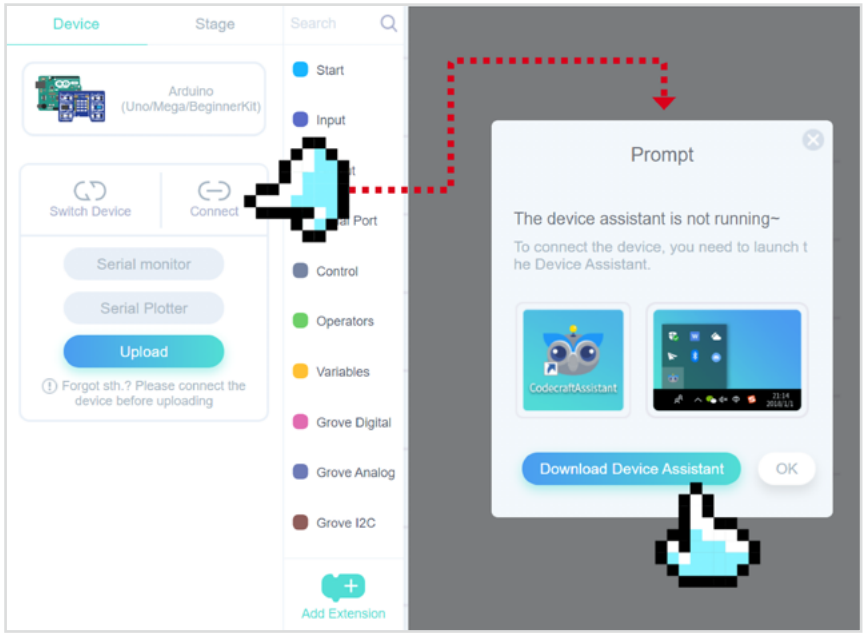
Enter the Codecraft main page and click Arduino (Uno/Mega/BeginnerKit) to create an Arduino hardware project.



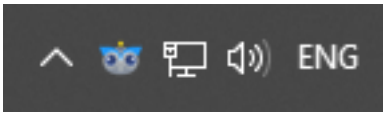
You need to download and install Codecraft Assistant for Device Mode

To program your hardware devices with Codecraft, you need to install **Codecraft Assistant** to connect your devices to Codecraft Web.

If you didn't have **Codecraft Assistant** installed, there will be a pop-up window reminding you to download and install **Codecraft Assistant** first when you click Connect.



Once Codecraft Assistant is installed, you will find the CC Assistant icon added to the desktop and the taskbar at the bottom right corner of the desktop.



If you have installed and Codecraft Assistant and run it, you can successfully connect devices in Codecraft Device list thereafter and no pop-up window will appear.

Codecraft PC Client

Operating Systems

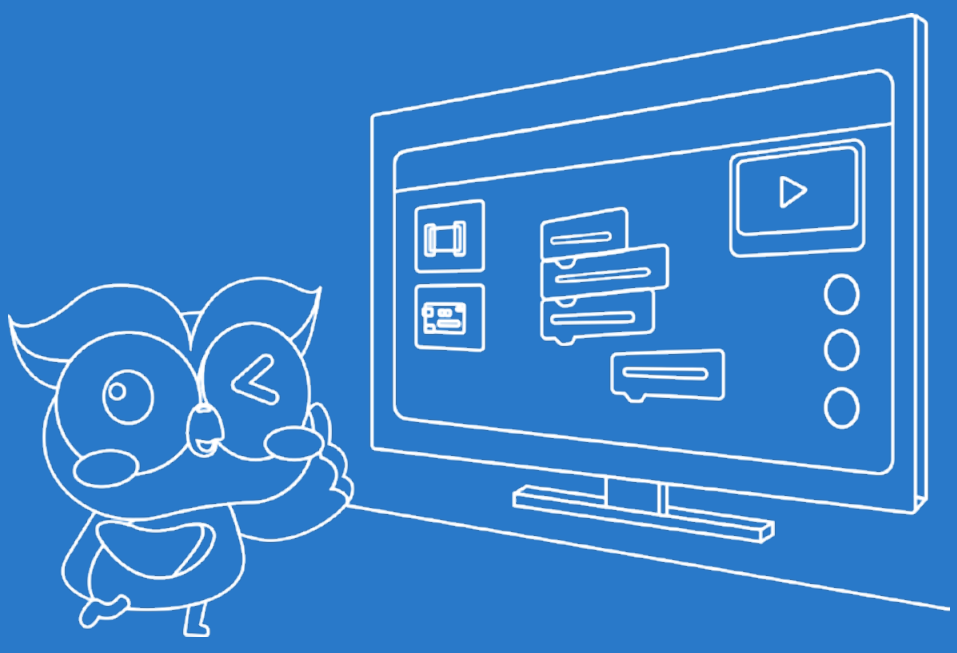
Windows 7 and Windows 10 with 32-bit and 64-bit systems
Mac OS 10.13.6 or above
Codecraft Installation Package Download URL
https://ide.tinkergen.com/download/en/

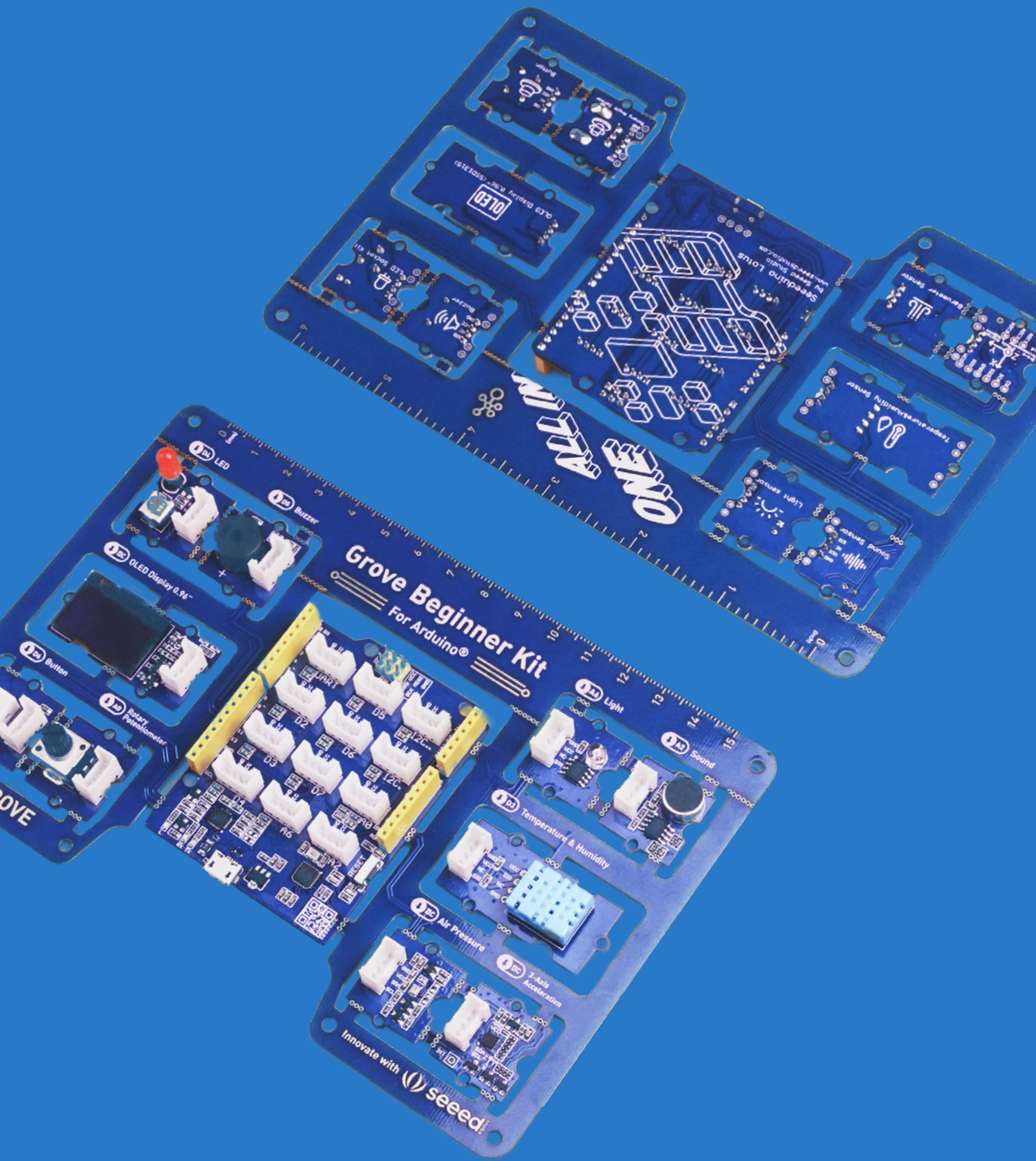
Windows Users Installation Instructions

Find the installation file in the download directory (file names vary with versions) and double click it to install the file.
After installation, Codecraft desktop client auto starts.
For more detailed instruction and information about using Codecraft on Mac/Linux, please visit our TinkerGen Help Center section about Codecraft:
https://www.yuque.com/tinkergen-help-en/codecraft
Once you have Codecraft installed and board at your hand, you are ready to start learning and exploring the possibilities of Grove Beginner Kit!

When the web version of Codecraft is connected to the starter kit, if the device assistant has been installed and it prompts you to install it, you can find and run the following icon on the computer desktop.

If you have installed the PC version of Codecraft on your computer, you can find the icon below to run it.




Grove Beginner Kit
For Arduino®
Innovate with Seeed
LED
Buzzer
OLED Display 0.96"
Button
Rotary Potentiometer
Light
Sound
Temperature & Humidity
Air Pressure
3-Axis Accelerator
Seeeduino Lotus
Temperature&Humidity Sensor
Light sensor
Sound Sensor
OLED Display 0.96" (SSD1315)
Buzzer
Button
ONE

Grove Beginner Kit Course

Lesson 1 Blink - setup-loop, delay, digitalOutput

Lesson 2 The Lights Go Down - analogOutput, PWM

Lesson 3 Enter the Loop - WHILE/WHILE NOT/FOR, variables

Lesson 4 Under One Condition - IF-ELSE statements, digitalInput

Lesson 5 The Potentiometer Keeps on Turning - analogInput, map function

Lesson 6 Morse Code - Buzzer, Button

Lesson 7 Motion Picture - OLED display

Lesson 8 Direct Access - Serial Input/Output

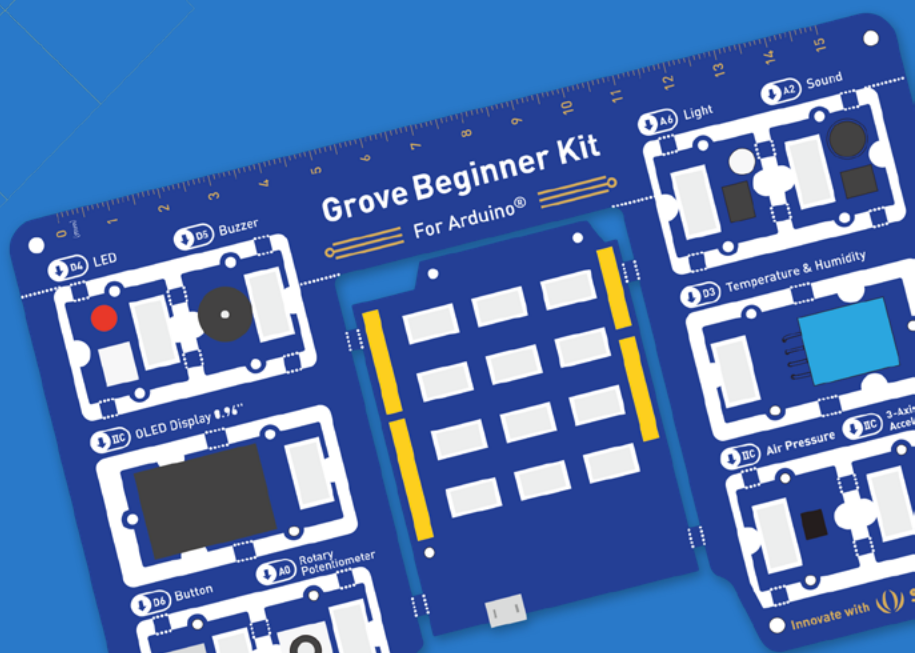
Lesson 9 See the Sound - Sound sensor

Lesson 10 Speed of Light - Light sensor

Lesson 11 Gaining Altitude - Air Pressure Sensor

Lesson 12 Rain and Shine - Temperature and Humidity sensor

Lesson 13 What Goes Around - 3-axis Accelerometer

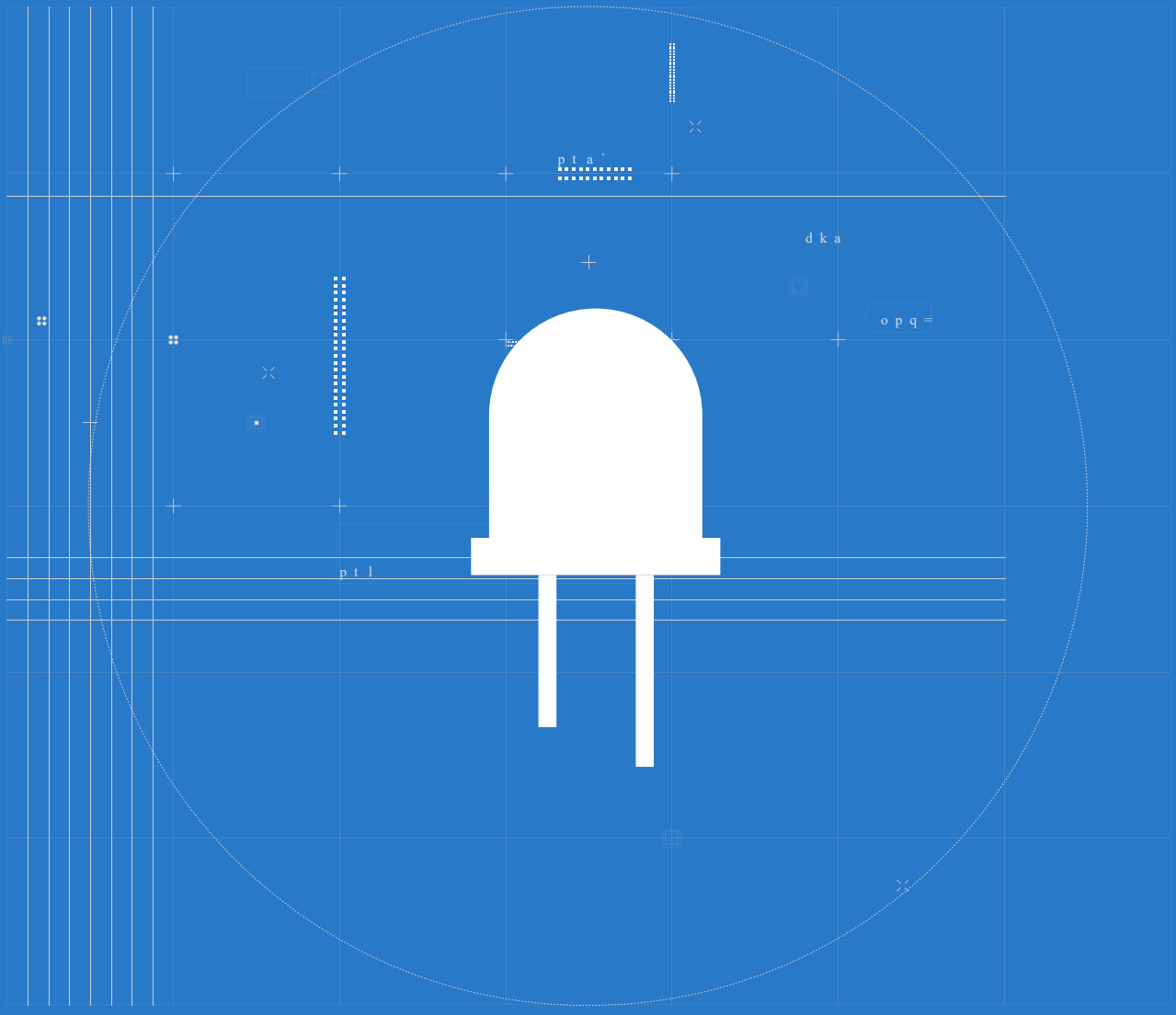



103
102
D4
LED

Lesson 1

Blink

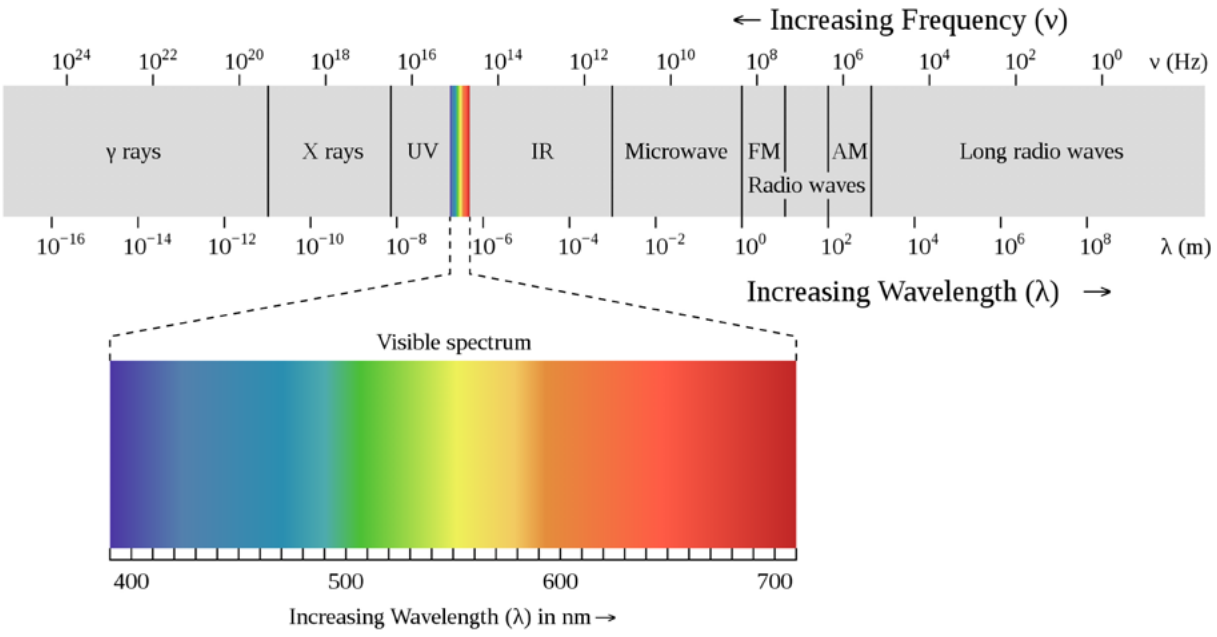
As you might have guessed the result of Blink program is blinking LED. Not very impressive? Rome wasn't built in a day and a program of a thousand megabytes begins with a single line. Blink program is Hello World equivalent for people learning programming on embedded systems. It is the first step towards understanding how computers sense and interact with real world, the logic and syntax needed to write more complex programs, that will allow us to implement our ideas in life. So, let us begin!



The Big Picture

Light

When we're very young, we have a very simple idea about light: the world is either light or dark and we can change from one to the other just by flicking a switch on the wall. But we soon learn that light is more complex than this. What we call light is actually a form of electromagnetic radiation with a wavelength which can be detected by the human eye. Electromagnetic radiation forms, some of which you might be familiar with, include X-rays, ultraviolet rays and radio waves. Wavelength is the size of a wave, which is the distance between any two corresponding points on successive waves, usually peak to peak or trough to trough.

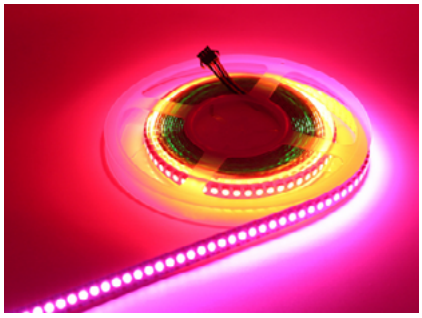


The world itself has no colors - it is our eyes and our brains have evolved this way, so that different wavelengths within visible light spectrum register as light of different colors by our eyes. This is also the reason animals see the world differently - their eyes have evolved in a different way.

Now when we have a rough idea of what light is, how can we "make" it? Well, there are multiple ways.

You could burn some wood and that would produce heat and light.

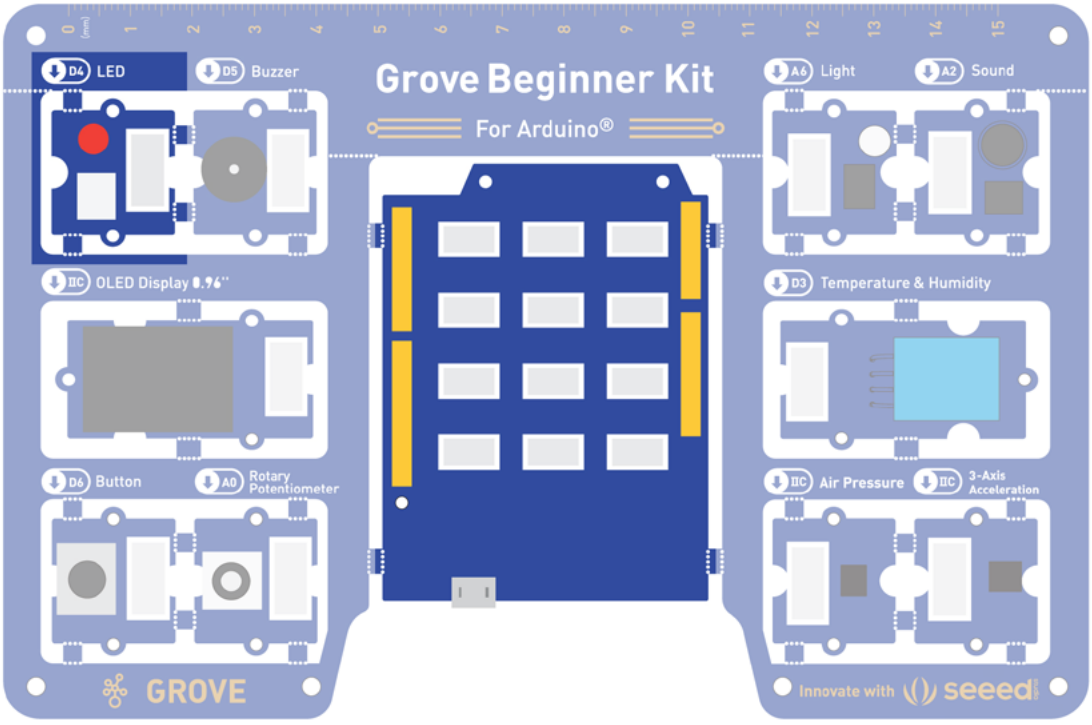
Or you could heat a metal sufficiently and then it would also start glowing - this is how the older incandescent light bulbs work.



The incandescent light bulbs are all gone now and replaced by more energy efficient illumination sources, such as fluorescent lights and LED lights.

LED module in Grove Beginner Kit

In our Grove Beginner Kit we have a LED module - LEDs contain semiconductor materials that are used as a light source.

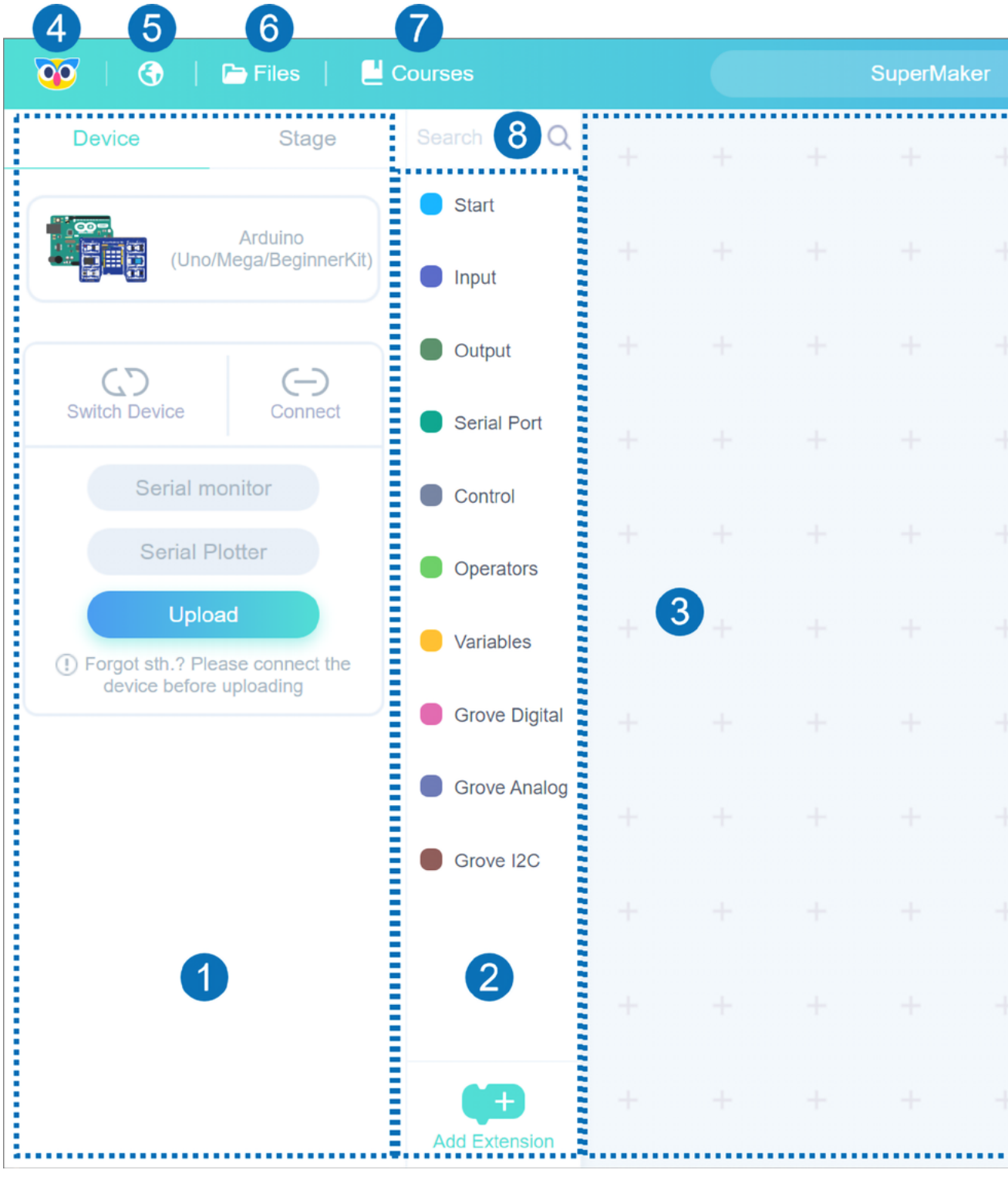


When a voltage is applied, the movement of electrons produces light. The process is called electroluminescence. Let's see how can we make that LED shine with a couple of blocks of code!

Task : Write a program to make the LED start blinking

Step 1: Overview of Codecraft programming interface

After you open Codecraft, select Arduino Uno/Mega/Beginner Kit as your device and you will see the following interface.



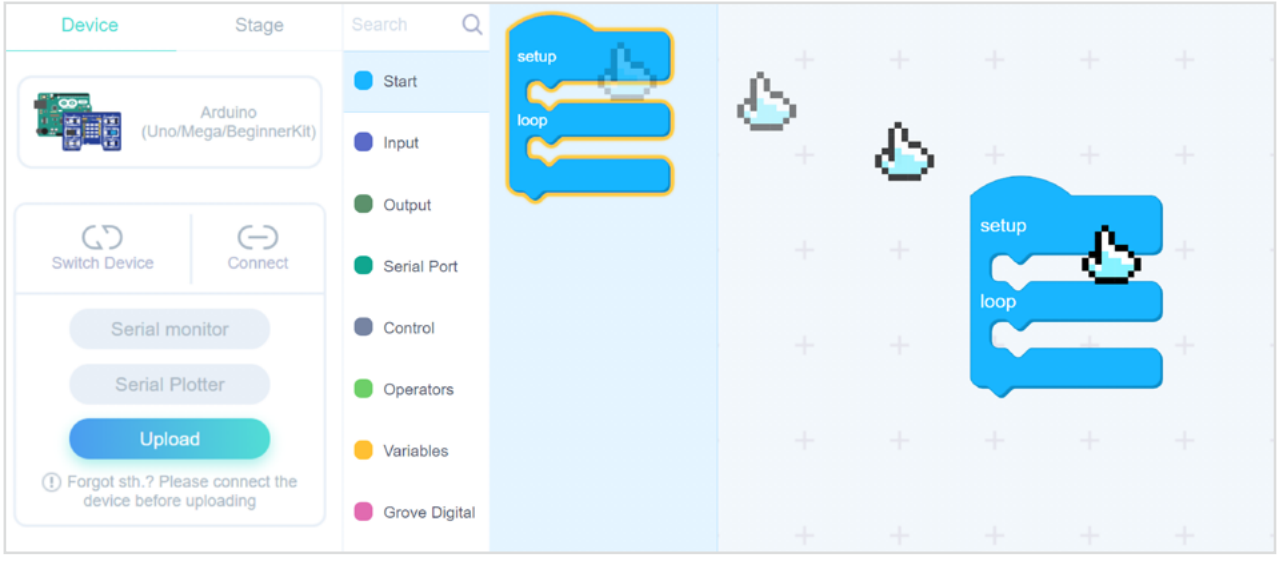
Visit **ide.tinkergen.com** or click on the linkage below to start your creation with Codecraft now.

https://ide.tinkergen.com

1. **Device/Stage Area**: You can choose stage/device programming here.
2. **Catalogue Area**: The categories are different in device/stage modes. You can select the blocks you need by category here.
3. **Scripts Area**: You can drag the blocks to Scripts area to make your programs.
4. **Return to Codecraft Home Page**: You can click on it to return back to the Codecraft Home Page. If you have not saved the current program, it will prompt you to save it.
5. **Language**: You can click on it, then language options appear. You can change languages here.
6. **Files**: You can create a new online project, or open a local project, or save your own projects to the cloud from computer following the on-screen instruction.
7. **Course**: Here you can check projects and course about Codecraft.
8. **Search**: Here you can search Blocks about Codecraft.
9. **Save Files Online**: You can modify your project names and save them to the cloud. (Codecraft should be connected to the internet and you should log
10. **Blocks/Codes**: You can toggle between blocks and code. Code language varies with hardware.
11. **Help**: We always appreciate your suggestions for Codecraft. We can't do better without your involvement.
12. **Login**: If you have not logged in, it will prompt you. If you are logged in, you can visit your cloud projects, account setting, my invitation code and log out.

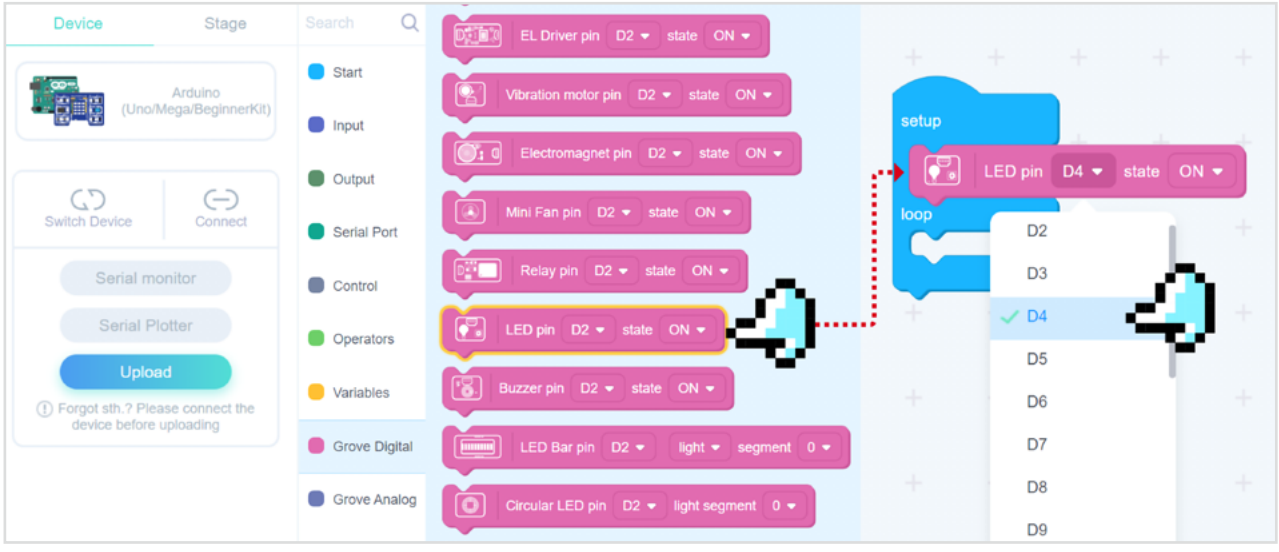
Step 2: Add setup and loop blocks

In the start column of the block classification area, you can see the setup and loop blocks. Every program we will make in this course consists of two parts - the setup and loop. The blocks within setup part are executed in sequence once, on the board startup (or after you press reset button). The blocks within loop part are executed in sequence and after the final block has finished executing, the board goes back to the first block and repeats the whole process - this is why it is called loop. Let's put this setup-loop block and upload our first program. To do that click on **Start** category then click on setup-loop block and drag it to the **Scripts area**. Once it's there, click on upload button choose the port to which the board is connected and press Upload.



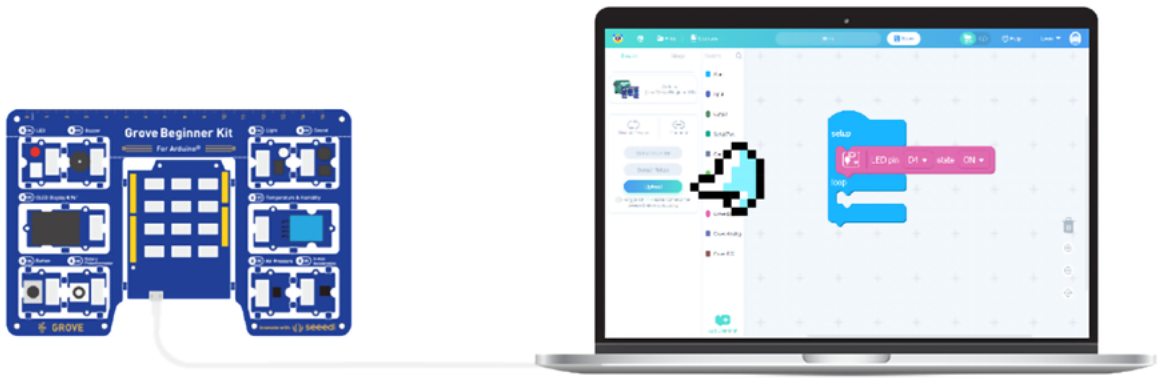
Step 3: Procedure for lighting the LED

Okay, now what? Our board doesn't seem to be doing ANYTHING. This is precisely because we instructed it do nothing. Let's add a block **LED pin ... state ...** to the setup part. Set the Pin number to **D4** - this is where our LED module is connected to the mainboard.

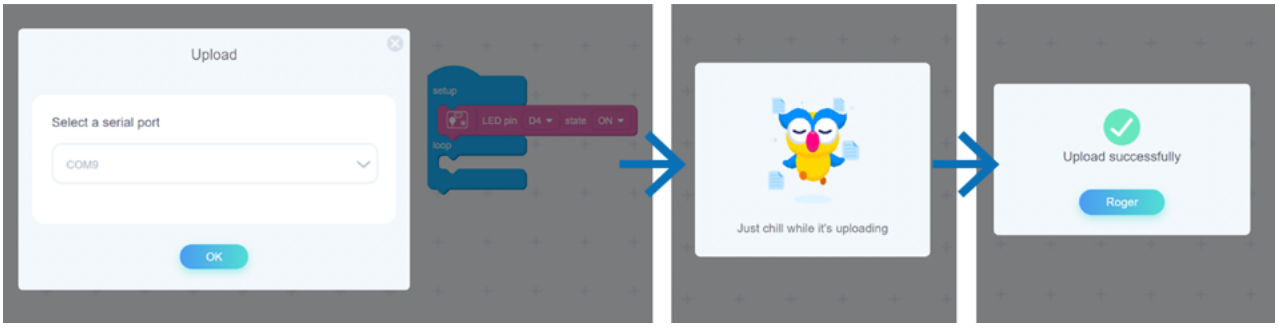


Step 4: Connect the Grove Beginner Kit

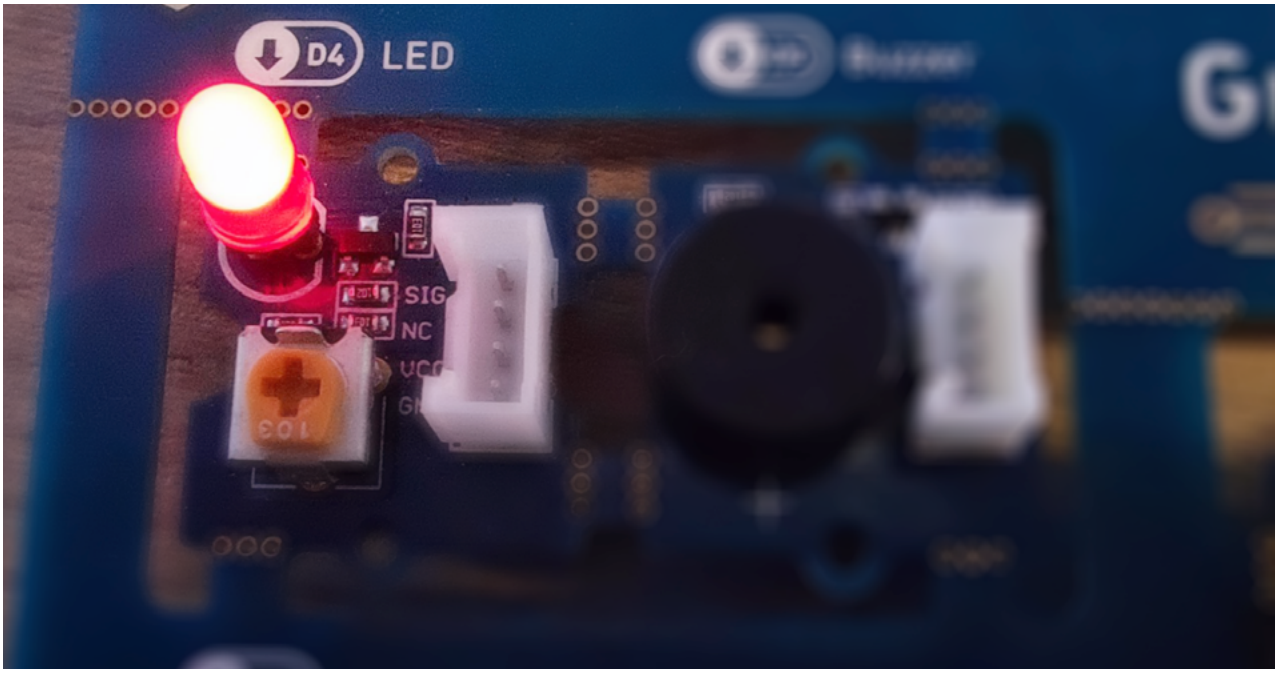
Connect the Grove Beginner Kit to the computer with a USB data cable, as shown in the figure below, and click the "**Upload**" button.



Upload until you see a successful prompt.

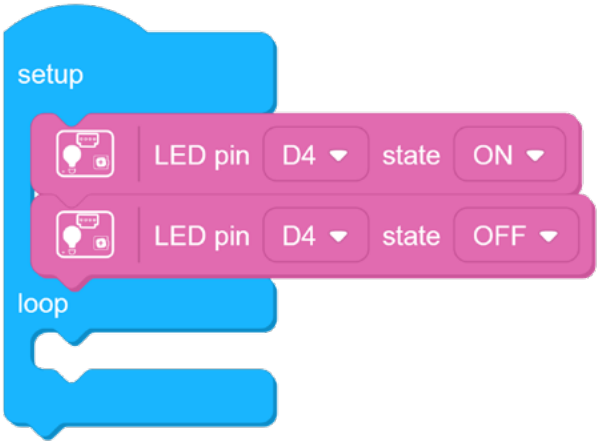


Let there be light! A 5V electric current, consisting of electrons, flows from our main board to LED and makes semiconductor material inside of it to glow.



Step 5: Light ON/OFF

Now if we wanted to switch off the LED, we could just change LED Pin D4 state On to LED Pin D4 state Off and re-upload the program. But doing it every time manually would be tiresome. Could we just place two blocks LED Pin D4 state On and LED Pin D4 state Off in a sequence? Of course.



Hm, that didn't work. Is your LED broken?

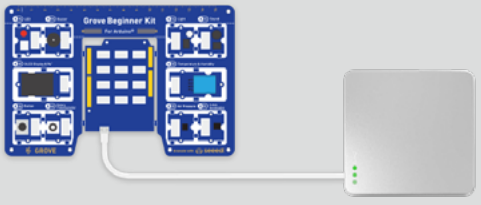
Not really. The microchip inside of our board did exactly what you instructed it to do - it closed the circuit allowing the electrons to flow and then opened it stopping the flow of electrons. It is just it happened way too fast for your eyes to see the difference. Let's add delay block, which you can find in Control category, to slow things down a bit. And here we have it! Our first blinking LED!

Download the package from the attached online course, which can be found by number

1_4.cdc

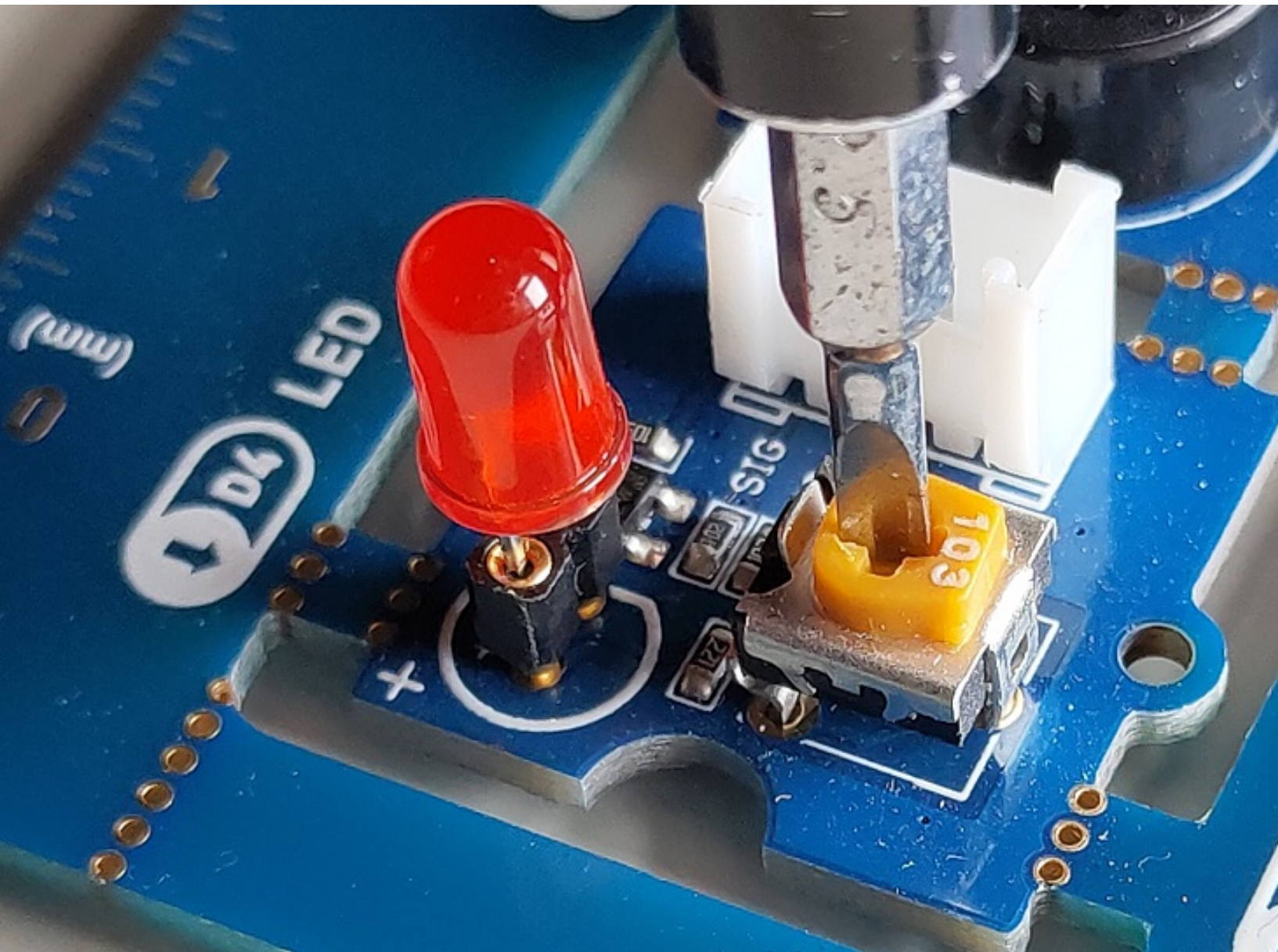
Note

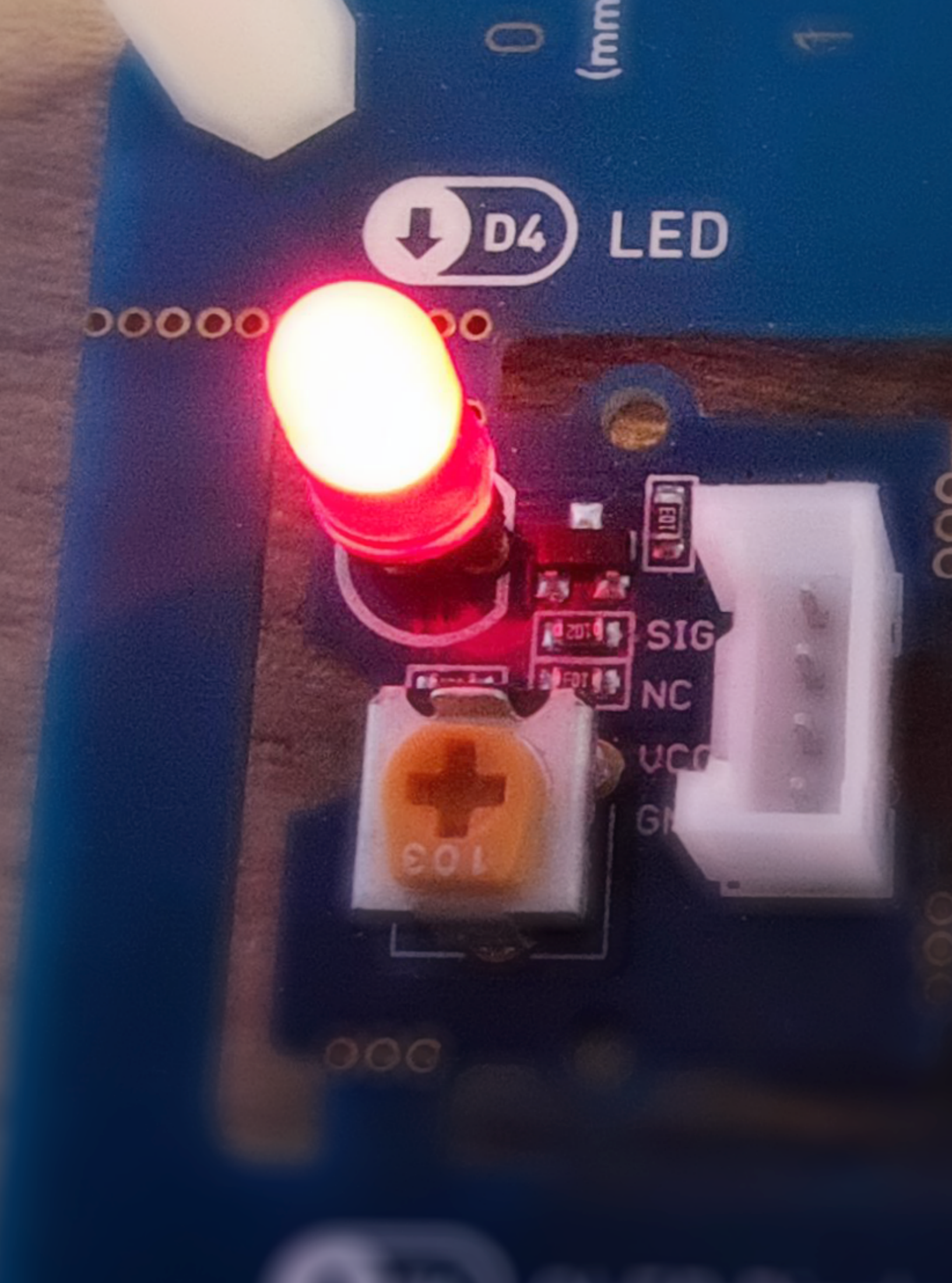
Our mainboard stores the program we uploaded in non-volatile memory, which means it stays there even after we turn off the power. That means that you don't need to upload the program from computer every time - after uploading the code you can provide power to the board with portable power supply (make sure that it is 5V!) and it will start executing the code you uploaded earlier.



⭐ Outside the Box

- Try moving the code from setup part to loop part. Observe the difference.
- Try removing delay blocks and moving the code from setup part to loop part. What is the result? Why do you think that happened?
- Try different delay values.
- Manually tweak the brightness of the LED by turning a small potentiometer on the LED module with a screwdriver.

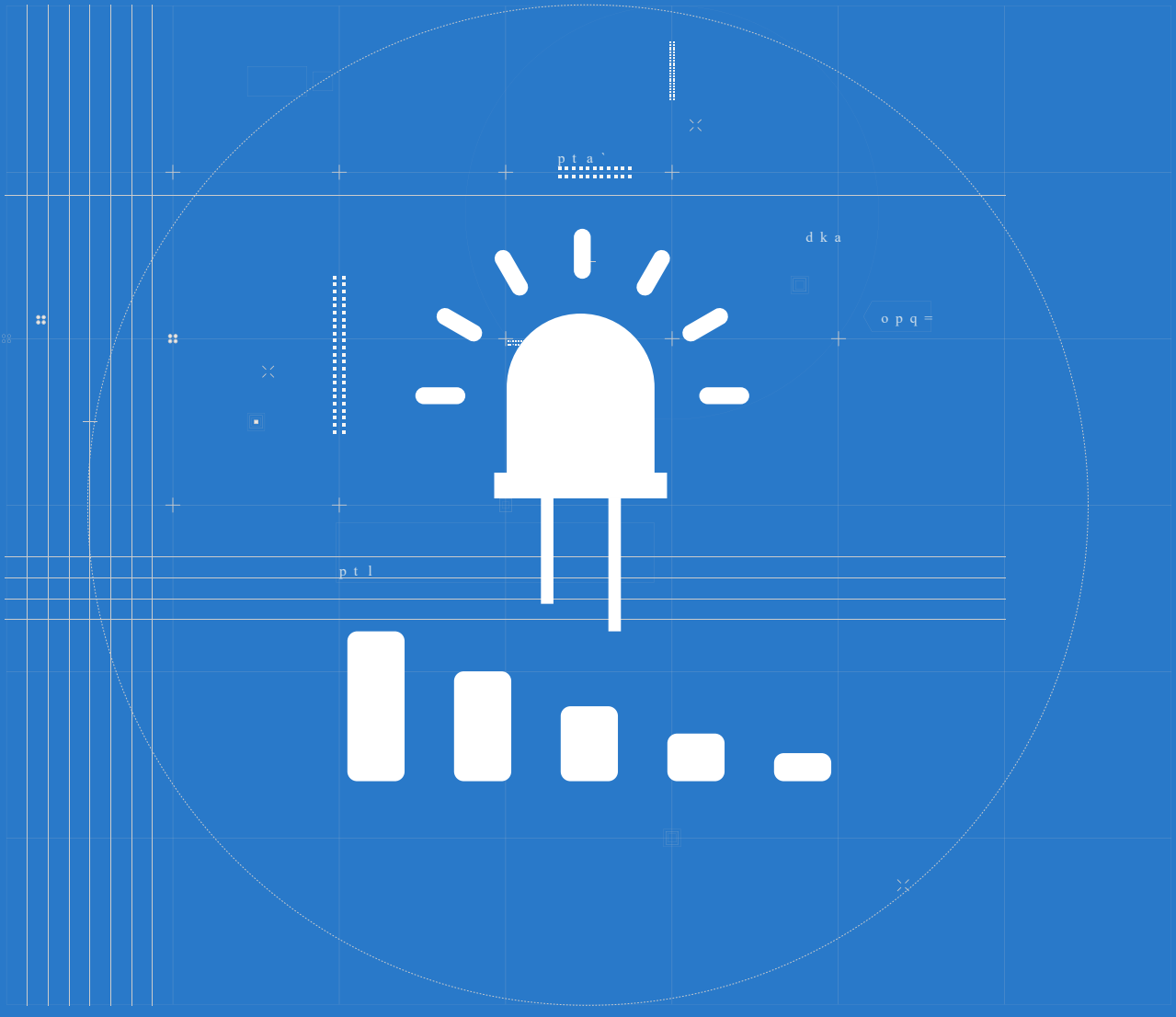



D4
LED
SIG
NC

Lesson 2

The Lights Go Down

That LED is shining brightly! In fact it is so bright, your eyes might hurt a bit when looking directly at it. In that case don't look directly at it. Problem solved. Or are there any other solutions? We need a way we can regulate the brightness of an LED - and by extension the same method allows us to control the speed of the motors, loudness of sound and do many other useful things.



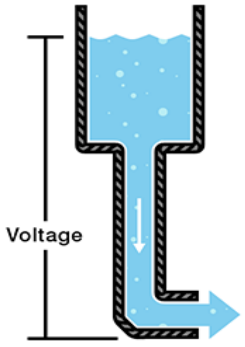
The Big Picture

Voltage, Resistance and Current

As we mentioned before, when a voltage is applied to LED, the movement of electrons produces light. Let's elaborate a bit on concepts of voltage and current here to understand how can we change the brightness of LED. Electricity is the movement of electrons. Electrons create charge, which we can harness to do work.

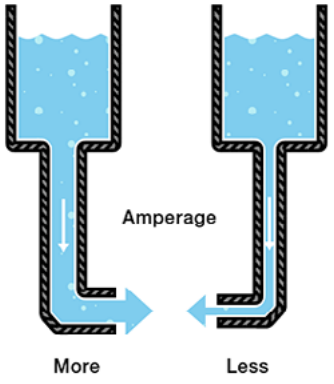
- Voltage is the difference in charge between two points.
- Current is the rate at which charge is flowing.
- Resistance is a material's tendency to resist the flow of charge (current).

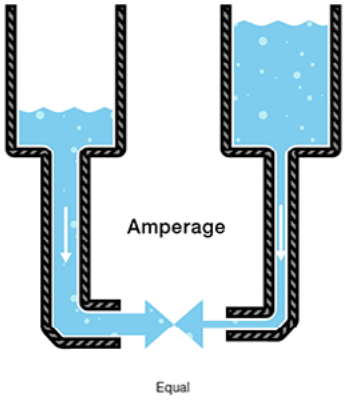
A good analogy for explaining these three concepts in simple terms is flow of water in a system. Imagine we have a water tank connected to a water pipe. The water from the tank will flow through the pipe and out of the pipe. The water pressure is "voltage" in this example, current is the flow of water.



What if we change the diameter of the pipe, but the amount of water stays the same?

The diameter of the pipe here is the resistance. Higher resistance of the material yields lower current, given the same voltage. We can increase the current by increasing voltage.





In the above case, despite the diameter of the pipes(resistance) is different , we're getting equal amounts of water from both pipes - by increasing water pressure(voltage) in the right tank and thus increasing the water flow speed(current). The mathematical representation of this is Ohm's Law, named after famous physicist Georg Ohm:

Voltage = Current × Resistance or

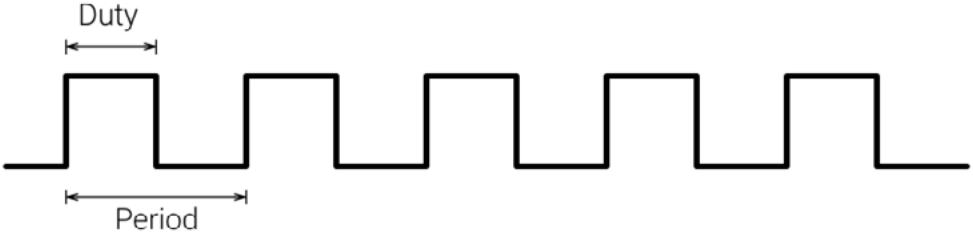
$$V = I * R$$

Georg Simon Ohm

Well, back to the task in hand - how to make the LED dimmer? We need to provide less electrons to LED - we could do it by

- increasing the resistance. But that would require us to change the properties of the wire and we are looking for solution that can be controlled from software.
- decreasing the voltage. Decreasing voltage will also decrease current, thus providing less electrons to LED. Less electrons means less light.

PWM

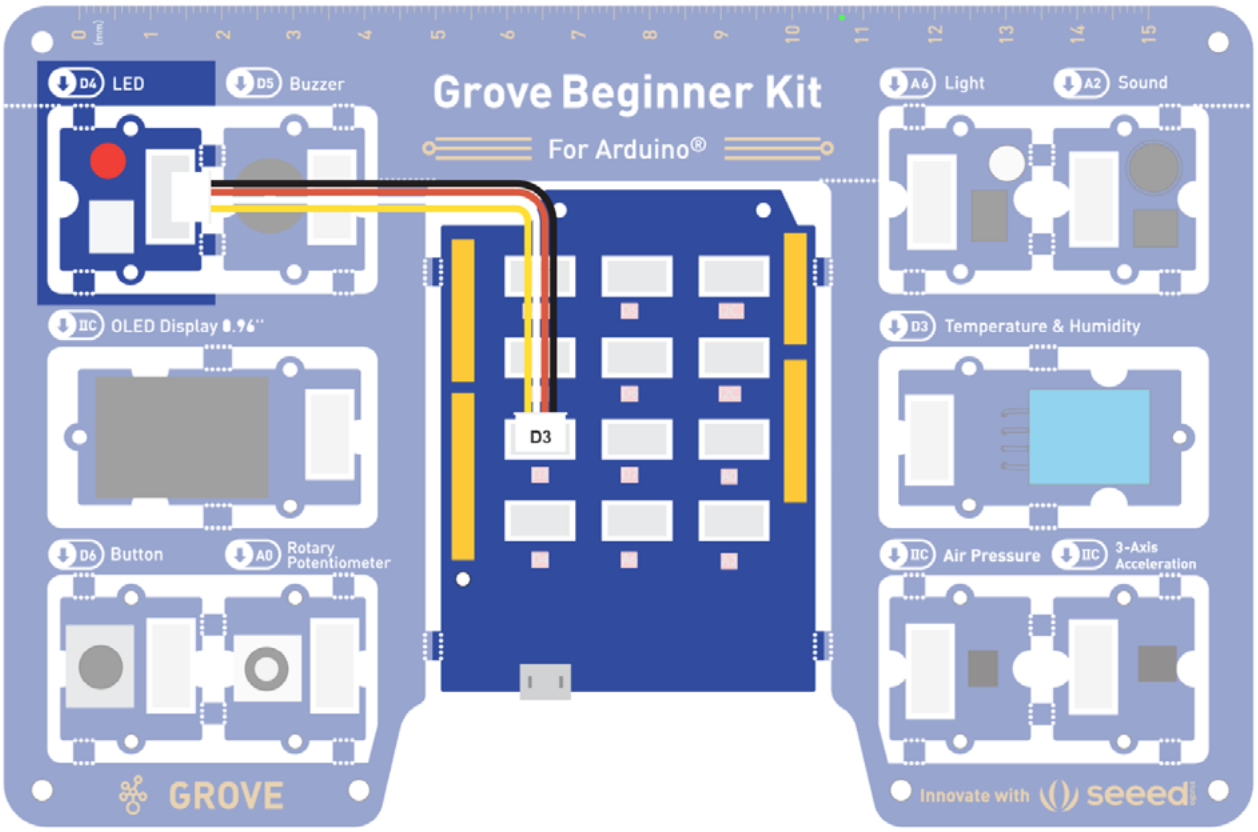


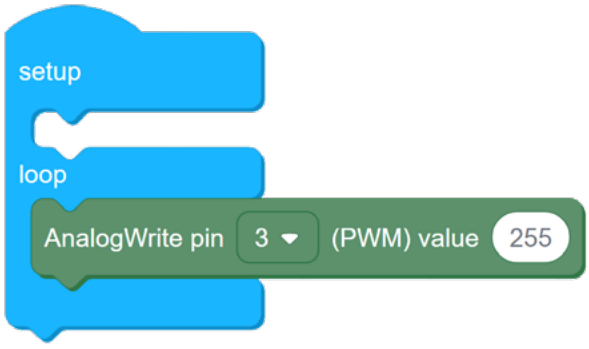
We can decrease voltage by using clever technique called **PWM** or **Pulse Width Modulation**, that allows us to provide different voltage from our control board. We will discuss in more details in later lessons how PWM works - for now let's consider it a magical black box. It is not magic of course, it is science.

Task : Fading LED light

Step 1: Set LED pin PWM value to 255

Let start by adding setup-loop code block and placing **AnalogWrite pin 3 (PWM) value 255** from **Output** category to Codecraft **Script area**. You might recall that our LED is connected to pin 4 by default and that pin cannot use PWM to adjust the voltage. So we need to use Grove cable and connect pin 3 to Grove socket of LED module. Then upload the following code.

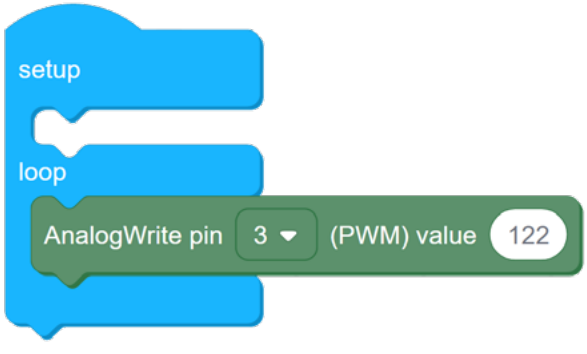




Step 2: Set LED pin PWM value to 122

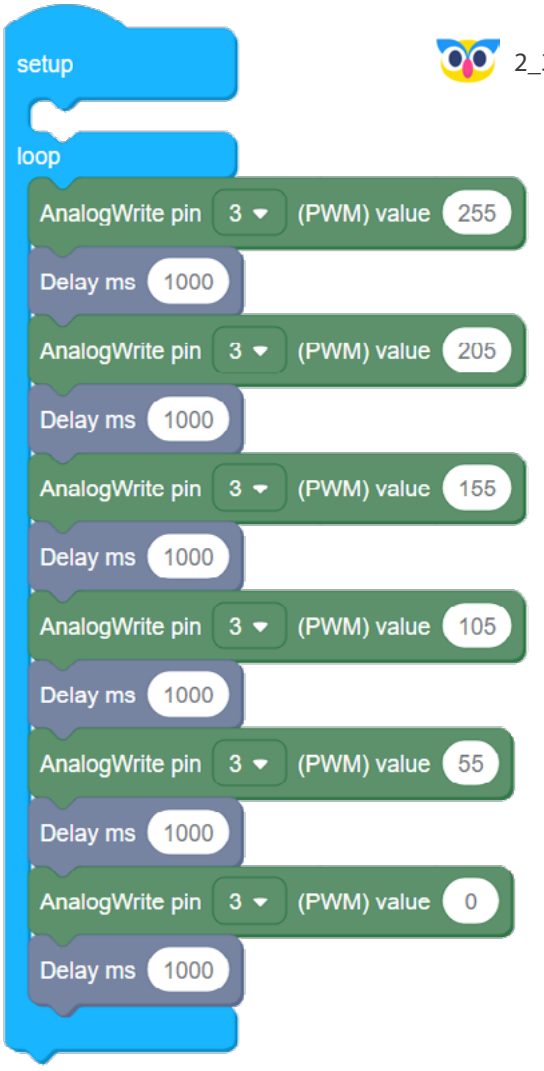
Hmm, not much difference from before… How about we change that value to something lower?

The LED now is supposed to be half as bright, because we are only providing half of maximum voltage to it. As you might have guessed there are 256 possible numbers you can enter as value, ranging from 0 to 255.



Step 3: Write simple code for LED fading from maximum brightness to zero

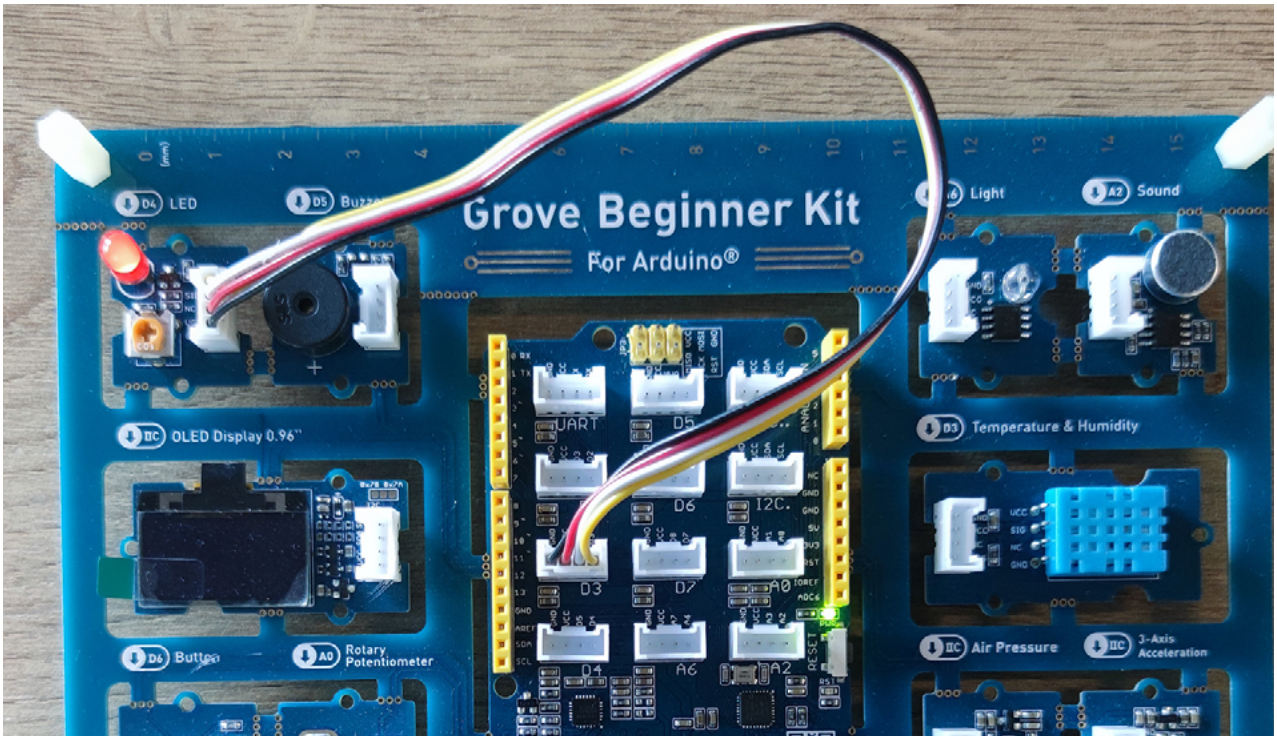
This code will make LED to change its brightness from 255(maximum brightness) to 0(off) over period of 6 seconds. In later lessons we will have a much better and concise way to write the same code.

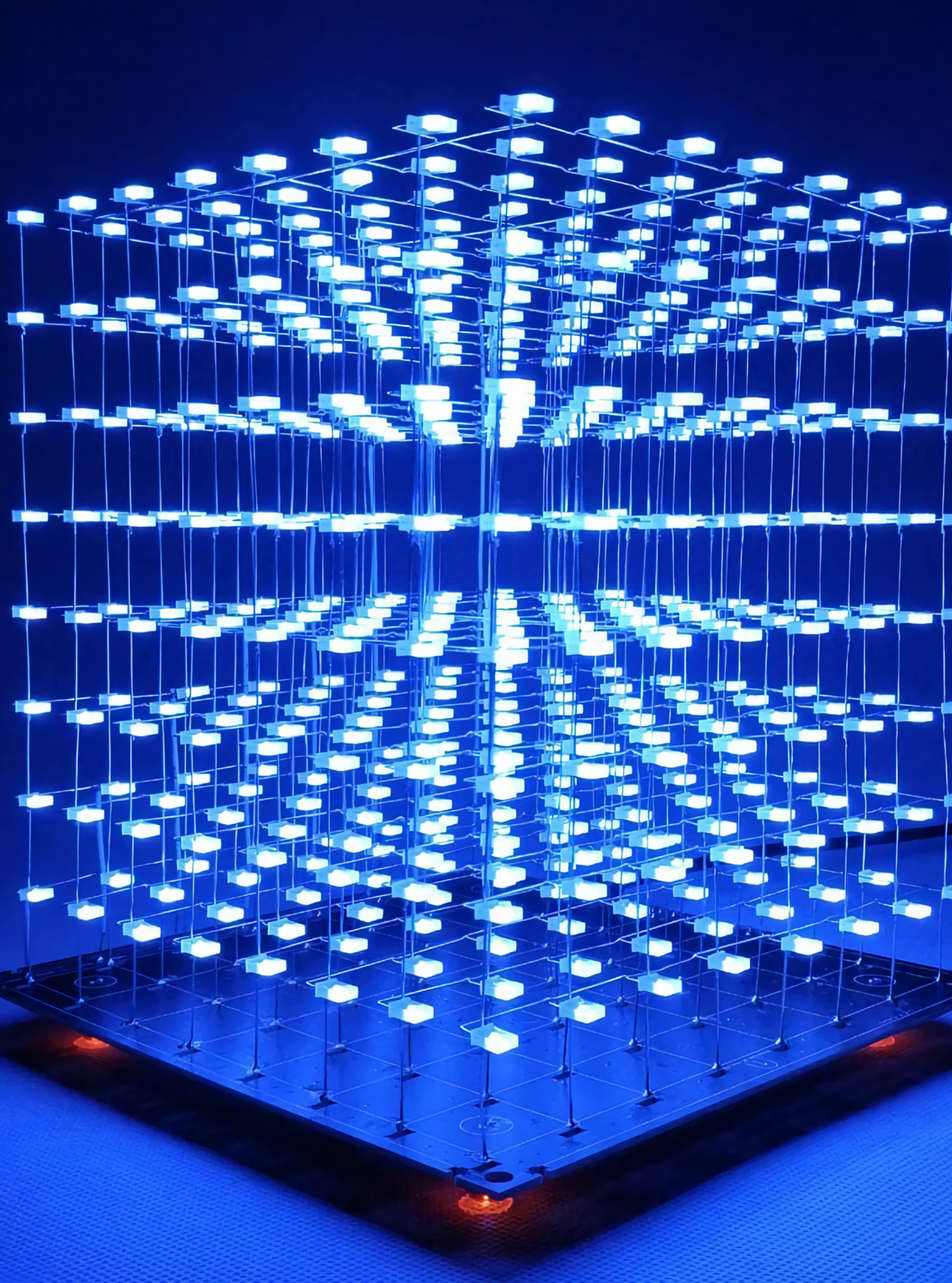


2_3.cdc

⭐ Outside the Box

- Try finding the minimum value when you still can see LED emit light
- Change the last code sample for LED to gradually fade (change brightness from highest to lowest)
- What effect will have entering numbers outside the range? Such as -1 or 256.

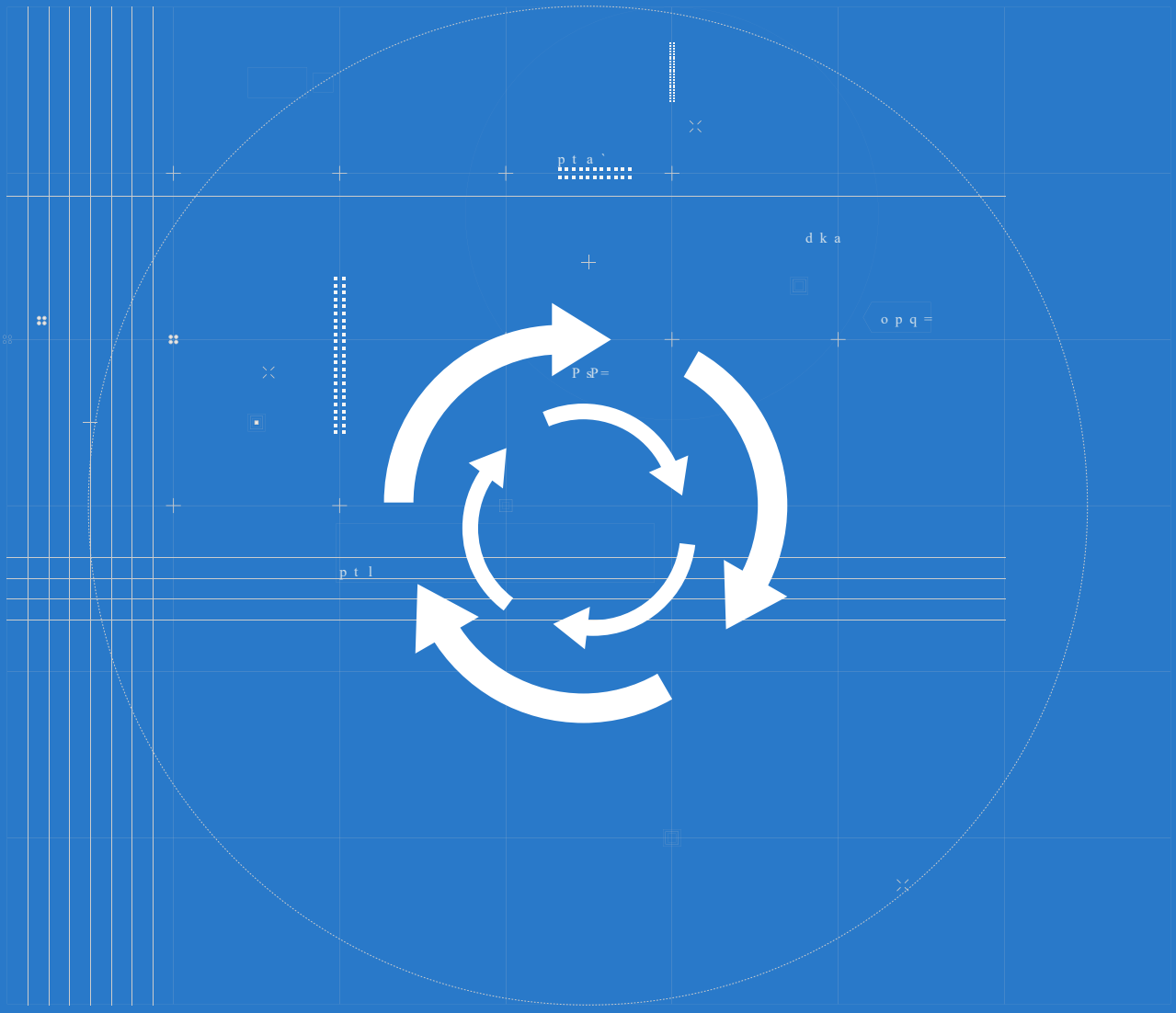




Lesson 3

Enter the Loop

In the last lesson we successfully experimented with LED light emitting different amount of light, which we adjusted with our code. The only thing that probably bothered you a little bit was the length of the code and the fact that we had to manually copy and paste code blocks each time just changing a few numbers. For this little LED dimmer, the process of copying and pasting code blocks was dull, but manageable.
For if we have an LED string with hundreds of LEDs?
And we want them to dim/shine brighter very gradually, increasing of decreasing brightness one percent at the time? Is there an easier way to do that?

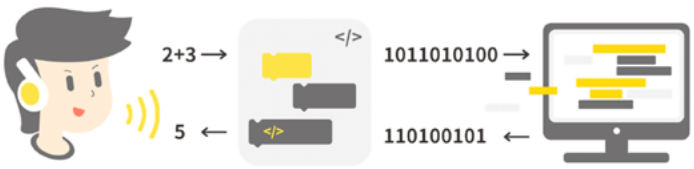


The Big Picture

How computers work

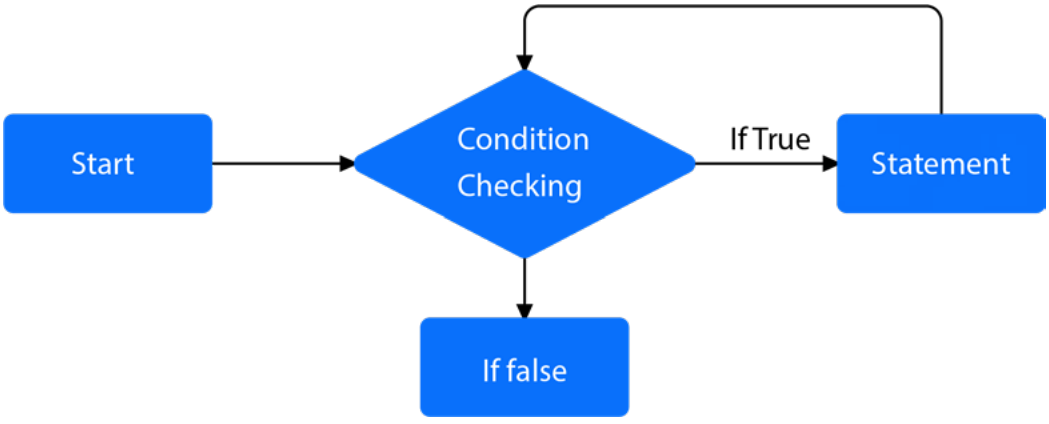
To solve this problem let's have a look at how the computers work.
The first computers were gigantic calculating machines and all they ever really did was "crunch numbers": solve lengthy, difficult, or tedious mathematical problems.

Today, computers work on a much wider variety of problems—but they are all still, essentially, calculations. So, in the way you can think of a programming language as a dictionary for translating between human readable words, that have meaning to you(such as "if", "else", "while", etc) to long strings of 0 and 1 that can be processed by computers. When you write a program, you are giving a set of commands to your computer and expecting a certain outcome based on those commands. But instead of speaking those instructions, you're writing instructions in a language that kind of resembles normal English, but with a few additional parameters and rules. Every computer program is a set of instructions; a sequence of short commands, one after another. It's about breaking up a complex task into a set of smaller, individual instructions and using a programming language to write those instructions.

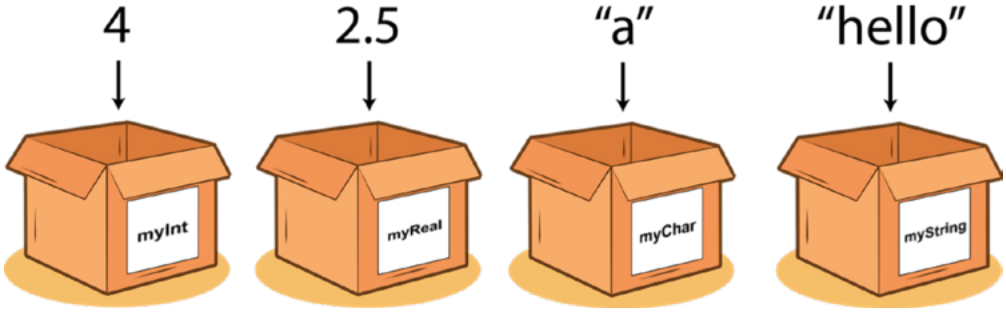


Loops

So, going back to our LED string with hundreds of LEDs problem. We want to use programming language to ask the computer to repeat certain parts of the code with some variations. We can use two important concepts in programming - a loop and a variable. A loop is a part of program that needs to be repeated until loop condition evaluates as False.



The simplest example of the loop is the infinite loop - it just goes on forever, since it's condition always evaluates as True.
In programming, a variable is a value that can change, depending on conditions or on information passed to the program. It is used for convenience of keeping track of important things during execution of program - similar how we use score in football game for example.

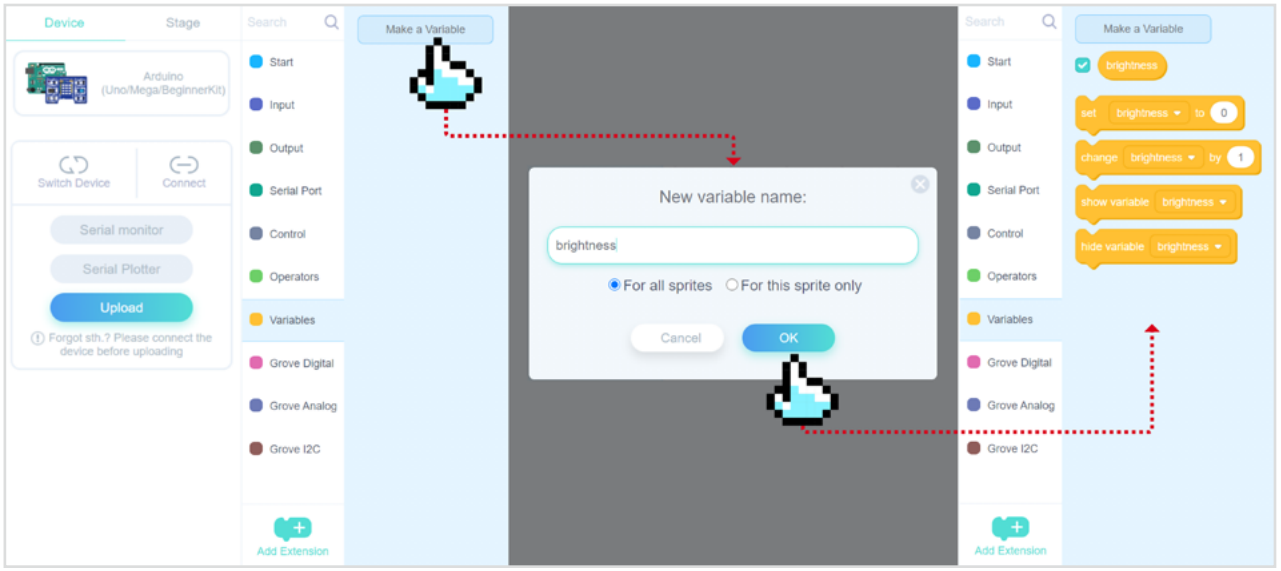


Let's see how can we use variable and loop to significantly shorten the program we made in the last lesson.

Task : Use loops and variables to write LED dimming programs

Step 1: Fading LED with variable and loop

Let's make the following program and upload it to the board to see the result. To make a variable click on **Variables** tab and then on **Make a Variable**.

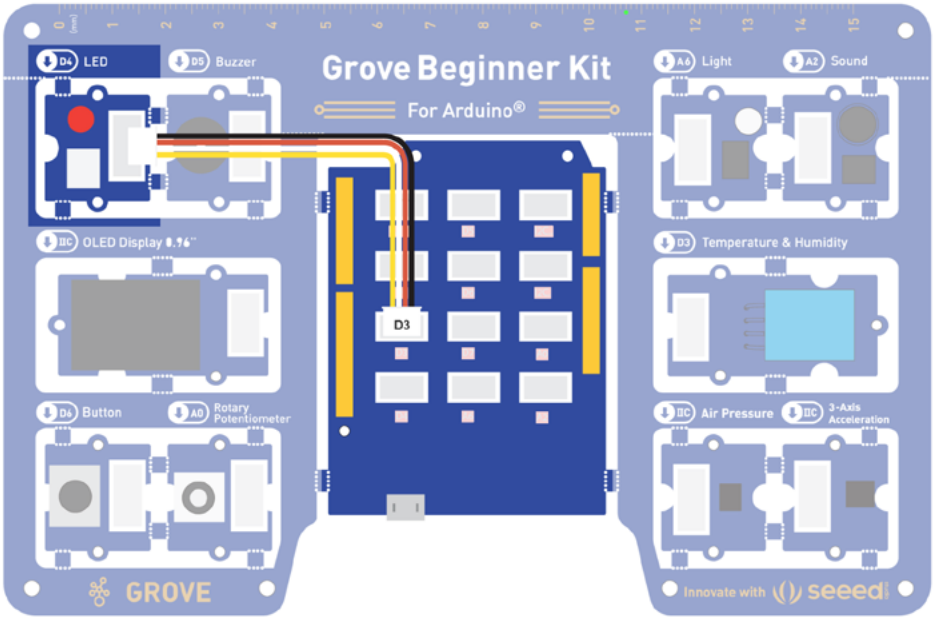


Give it a descriptive name, we have chosen **brightness**, but a variable can have any valid name - that will not influence the functionality.

Then make the following program(you can find the **EQUAL** block in the **Operators** category and **repeat until** block in **Control** category) and upload it to the board:

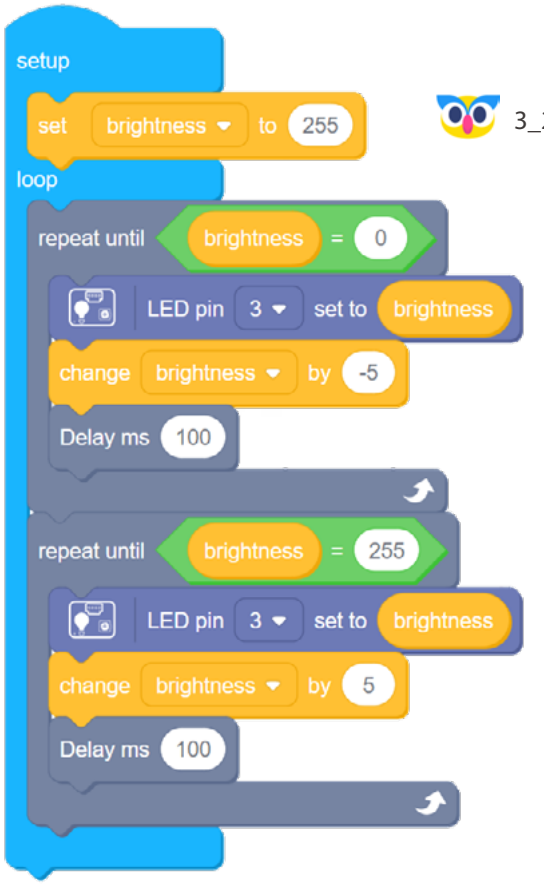
3_1.cdc

Remember, that LED module still needs to be connected to Pin 3 with Grove cable, since Pin 4 cannot.



Step 2: Breathing LED

We can see LED starting out with maximum brightness and then gradually fading completely only to light up a second later. We have two loops in the code - one is the main loop. In earlier lessons we mentioned this loop is a necessary part of any program for Grove Beginner Kit. The second loop is defined with repeat until block. We tell the microcontroller to repeat the commands in that block until variable brightness is 0.



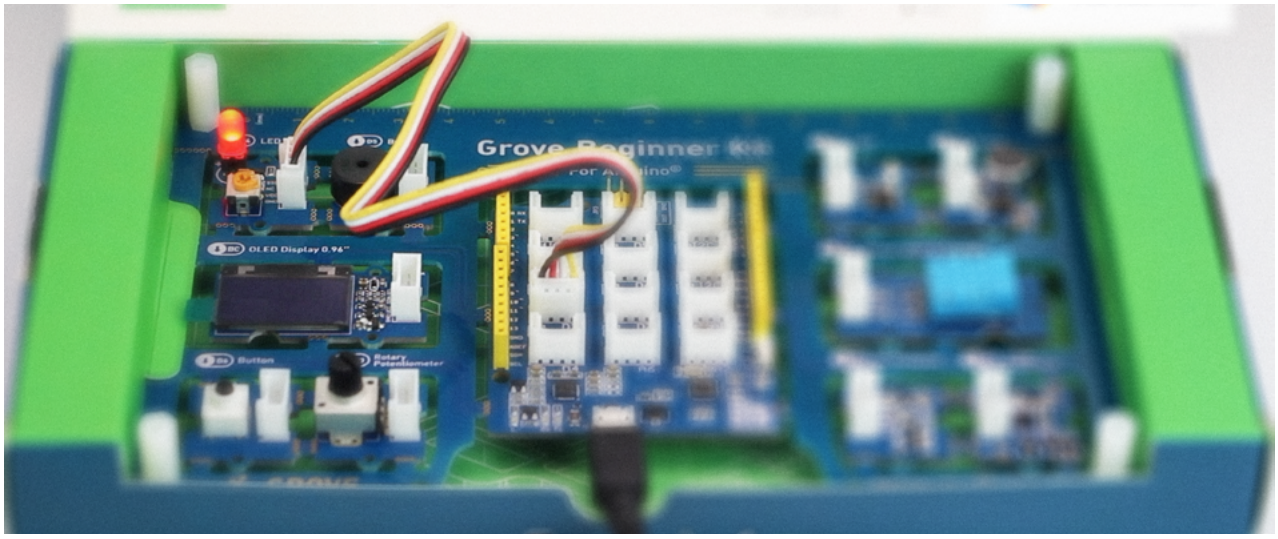
3_2.cdc

Step 3: Note about EQUAL operator

This code makes LED to fade and then gradually light up. Now, let's try speeding up the process a bit by subtracting and adding 10 instead of 5 to brightness variable.

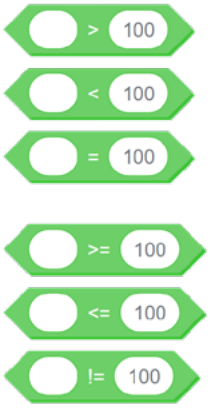
Oh-oh. Something went wrong - the LED fades properly, but then instead of gradually rising brightness it intermediately goes to the maximum brightness. Remember that computers execute instructions exactly according to the code you wrote. In our program we subtract 10 from brightness variable until it equals 0. But, if each iteration we subtract 10 from 255, it will never actually reach 0… If you want to do it with pen and paper, the values will go 15 5 -5. And since -5 and all the following values(-15, -25, etc.) do not equal 0, the code in the loop will never stop executing. The computer does exactly what we commanded it to do, but the result it not what we want.

3_3.cdc

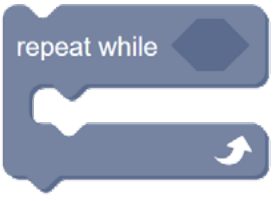


⭐ Outside the Box

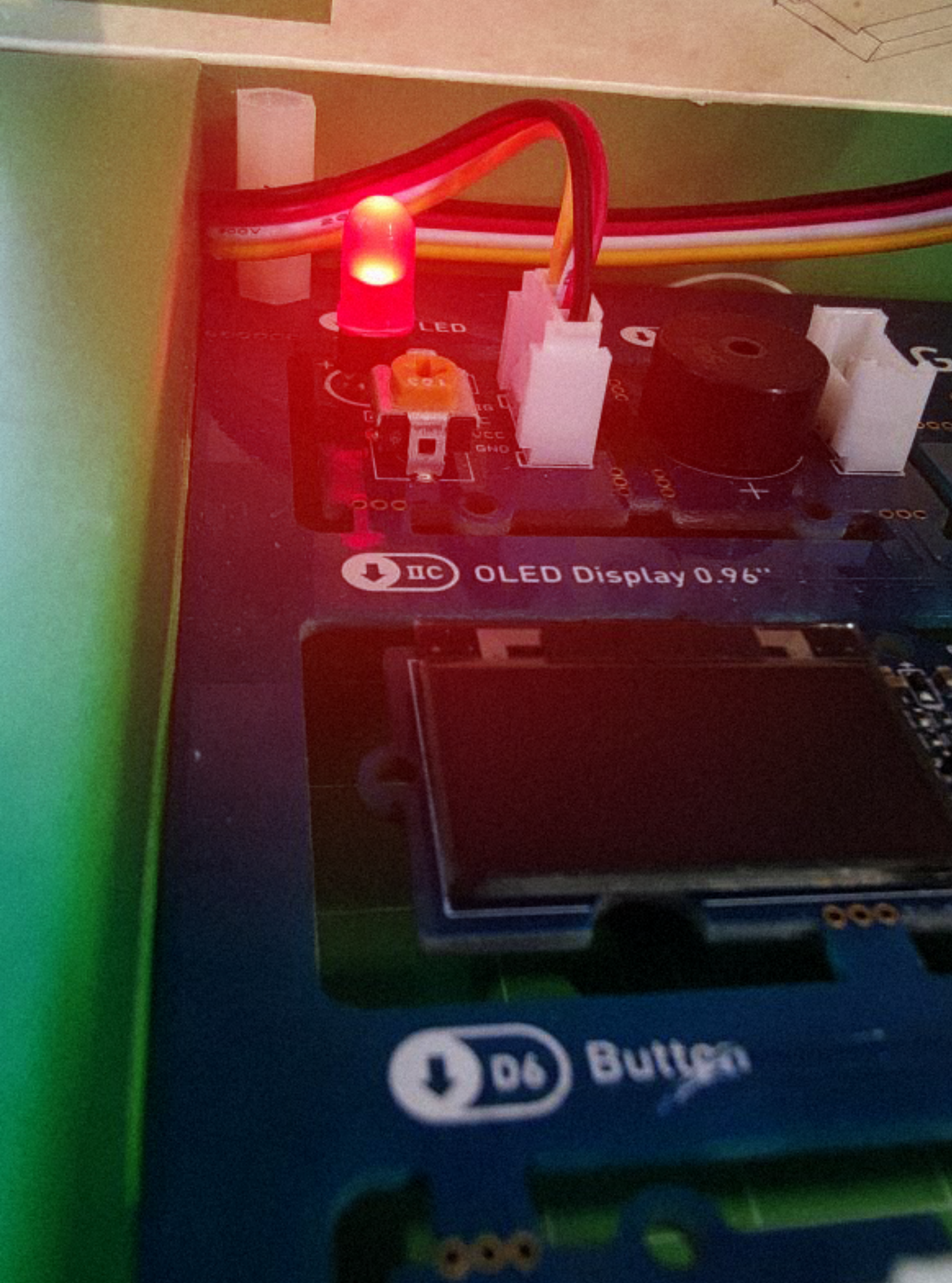
- Fix the example 3 by changing operator from EQUAL to LESS/GREATER.



- Try rewriting the example 3 using repeat while code block.



- Think of a way we can use repeat block to avoid copying and pasting repeat until block two times.

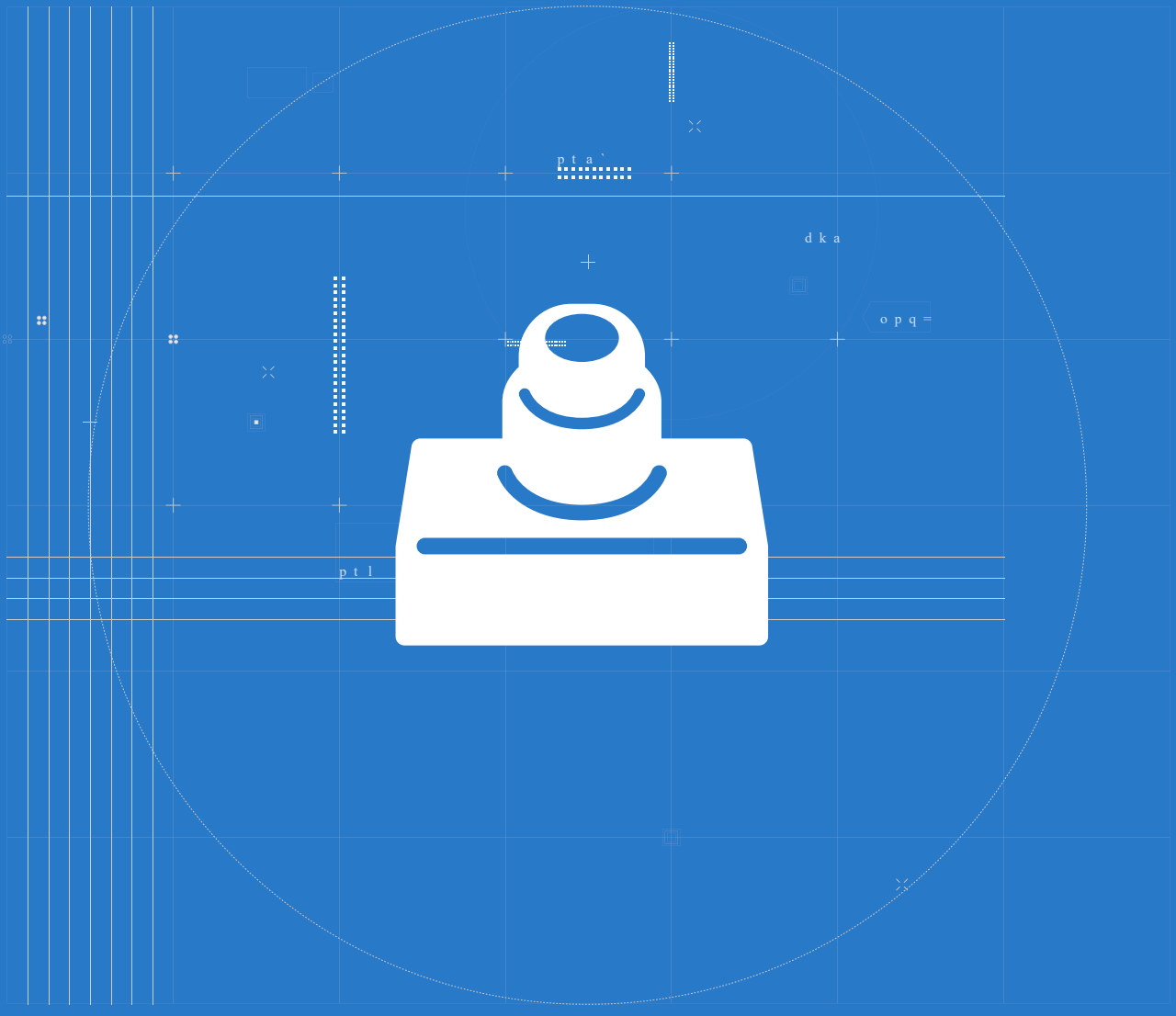

LED
GND
OLED Display 0.96"
D6
Button


D6
A0
GROVE

Lesson 4

Under One Condition

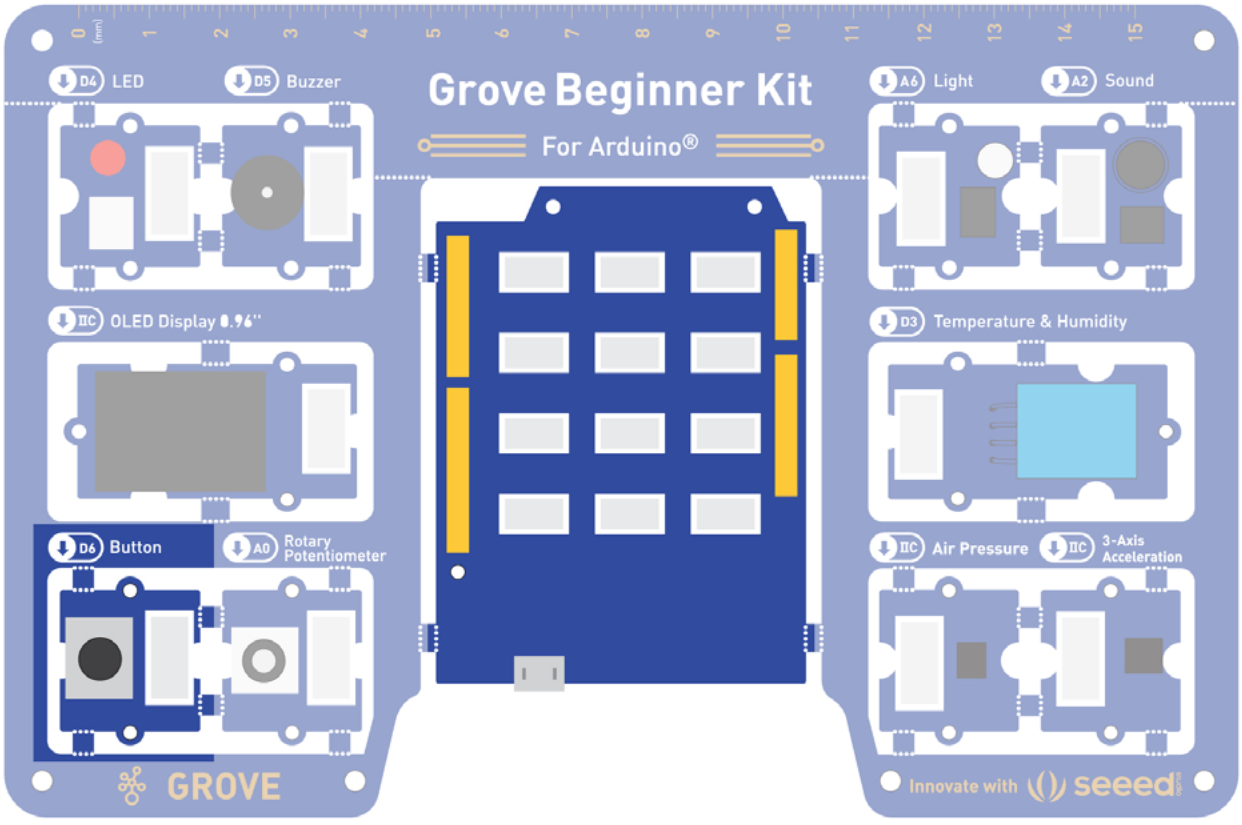
In the olden days robots were dumb machines, that can only do a pre-specified repetitive task and even slightest change in environment would completely break their routine and make them completely useless. Since then a lot of improvements has been made and we're going along the same path while exploring Grove Beginner Kit - in past few lessons the programs we wrote were rigid and we had no way to actually interact with the hardware. It is time to change it.



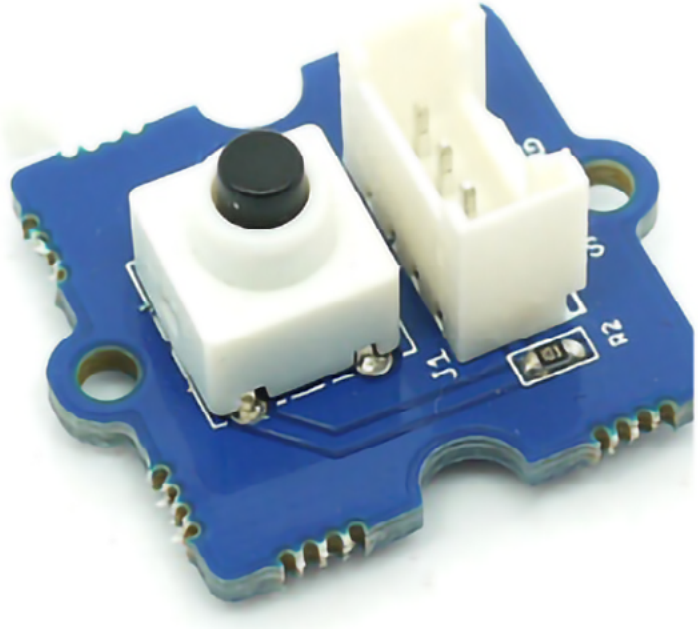
The Big Picture

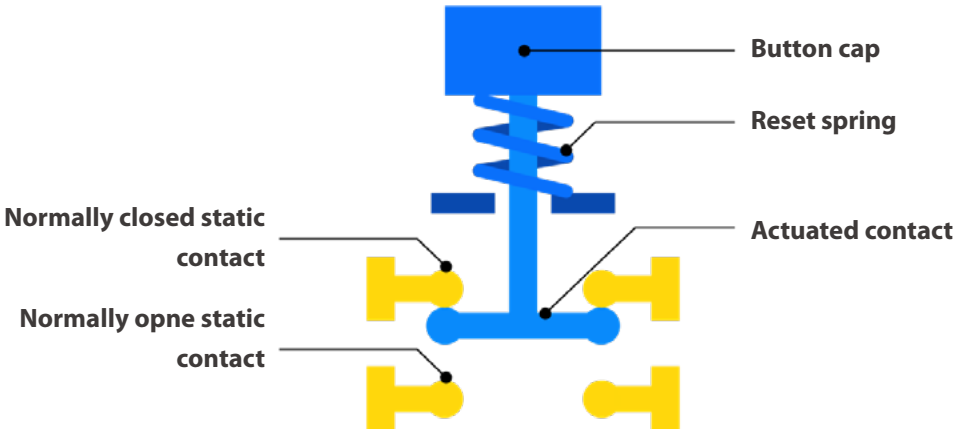
How do buttons work?

In this lesson we're going to use a new module to interact with our Grove Beginner Kit - a button.

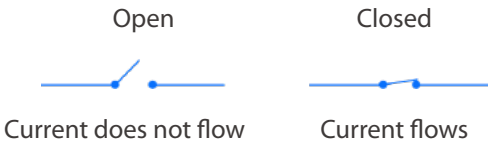


Button is a simple on/switch, which has mechanical structure that allows it to go back to the default(off) state. Of course, as we will see in almost every lesson in this course, even the things that seem simple on the outside are actually quite complex on the inside.



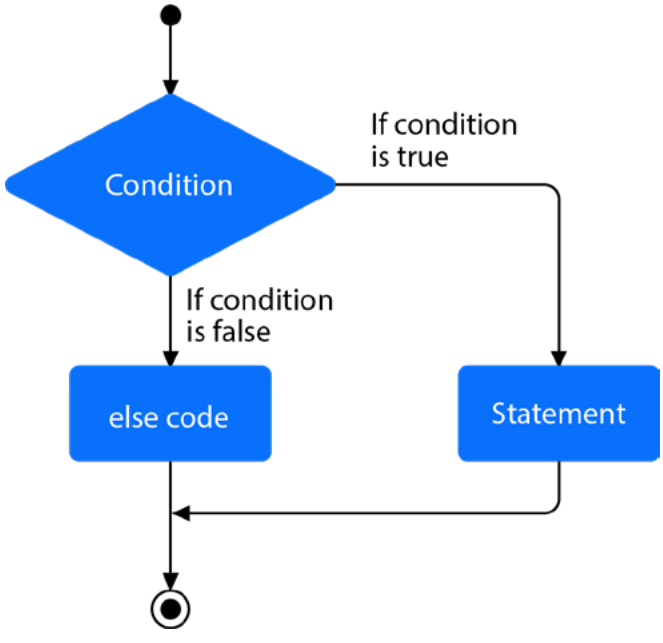


How does a button work? Or to take a larger view on things, how do on/off switches work in general? A switch is a component which controls the open-ness or closed-ness of an electric circuit. In the on state, circuit is closed and the electric current can flow inside the wires to its destination. In the off state, the circuit is open and the current is prevented from flowing, for example because there is a gap between wires. Switches are critical components in any circuit which requires user interaction or control.



Conditional statements

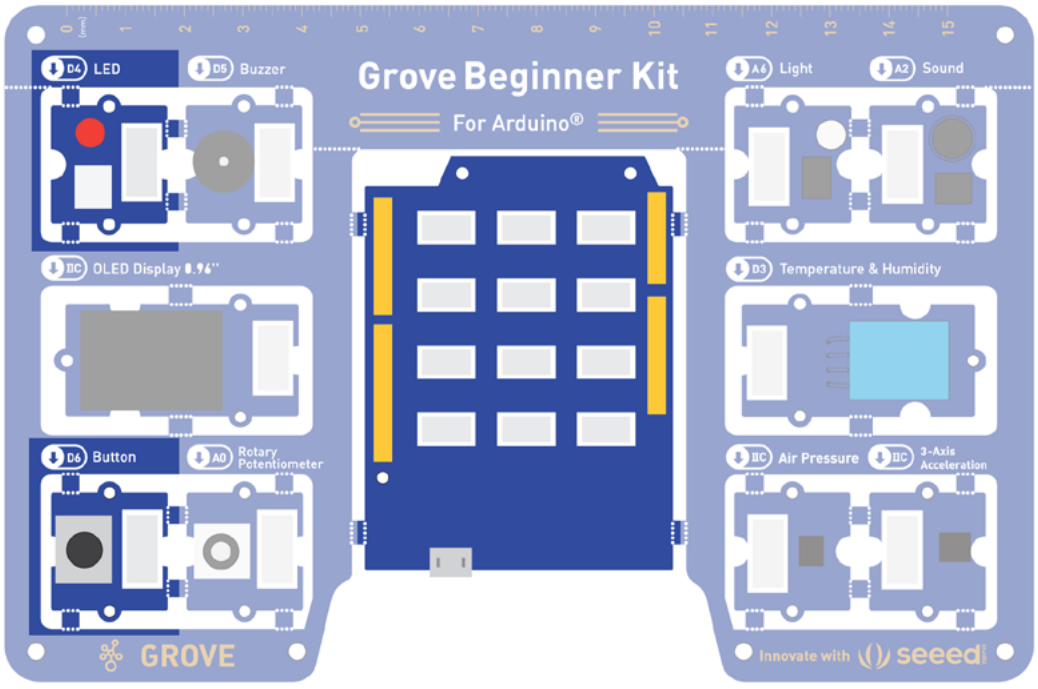
Closing and opening a physical circuit is the first step. The second step is to make the micro-controller act accordingly according to each case. We will use IF statement for that. IF statement in code is very much like IF statement in real life - you can think of many examples for it. Such as, "IF you finish your homework, you go play computer games". Us, humans, naturally understand the second, implicit part of IF statement most of the time - in this case that would be "or else, you cannot go play computer games". Computers on the other hand need this part explicitly defined with ELSE statement.



The code within IF statement block is executed if condition evaluates as true(button pressed, homework done, temperature is higher than 30 degrees, etc), the code within ELSE block is executed when condition evaluates as false(button not pressed, homework is not done, temperature is not higher than 30 degrees). Paraphrasing a saying by Linus Torvalds, talk is cheap, let's have a look at the code!

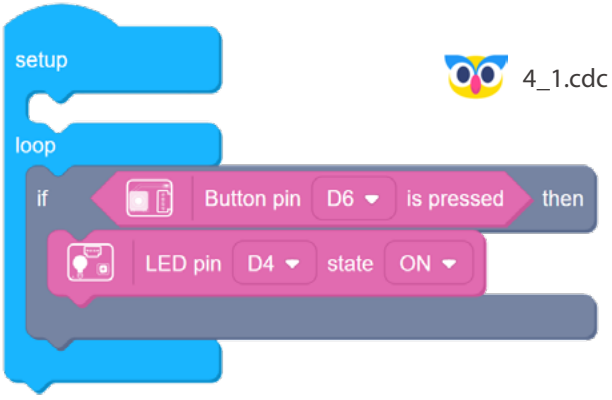
Task: Control LED light with Button module

The hardware modules used in this lesson.



Step 1: Use the button to turn the LED on

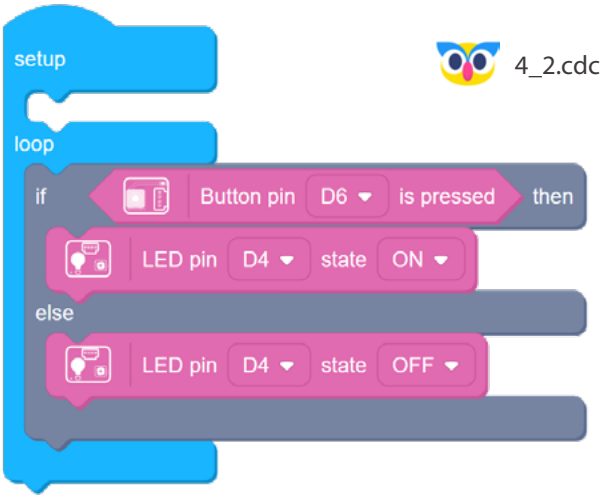
You can find IF statements in Control category and Button block in Grove Digital category. You can notice that Button code block has triangular shaped edges - it means that this block has two possible outputs: True or False.



4_1.cdc

Step 2: Use the button to turn the LED on and off

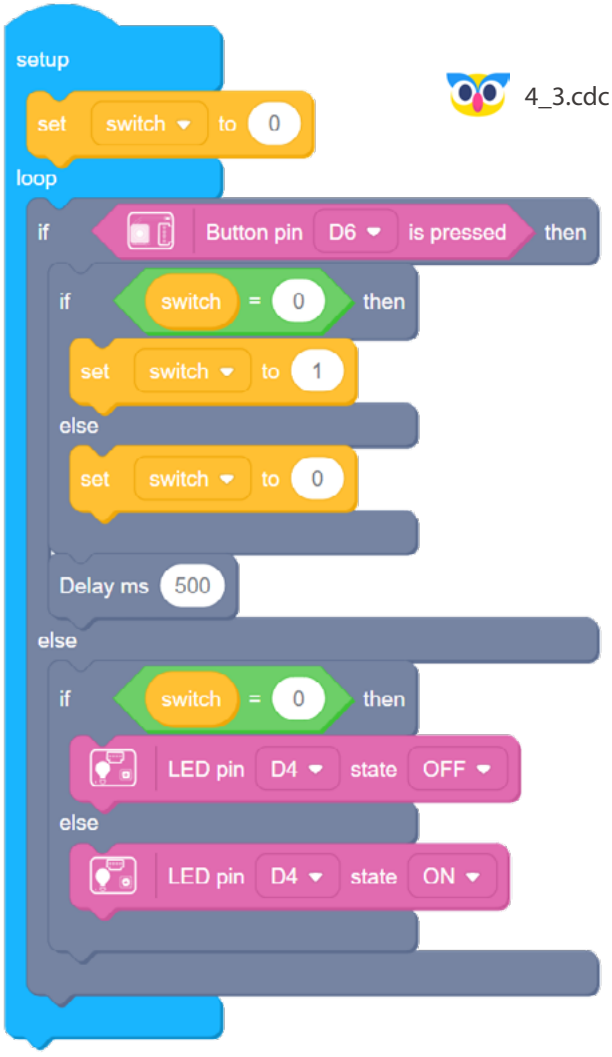
When we press the button, the LED lights up. But when we release the button, it stays on. Let's change the IF code block to IF-ELSE code block to correct that.



4_2.cdc

Step 3: Use the button to switch the LED on and off

Ah, much better. Still, as far as using this for house illumination it wouldn't be very convenient - we would have to keep this button pressed all the time, so not to allow our apartment to descend into darkness. Is there another helpful programming concept we can use to allow our program to "remember" the light state and toggle it with a button press? Of course there is! We can use a variable.

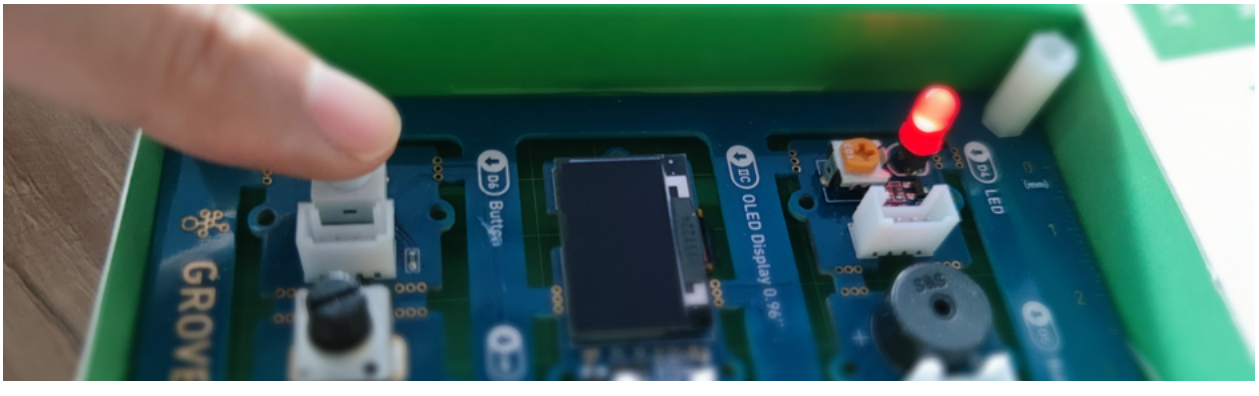


4_3.cdc

We have two if statements here - the first one changes the variable switch to it's opposite on button press. So if switch equals 0, then when button is pressed we set it to 1 and vice-verse. The second IF condition looks more familiar - it controls the LED, but this time the condition for switching LED on and off is not the state of the button, but the value of the switch variable.

⭐ Outside the Box

- What happens if you remove delay block from task 3? Try it out and see what happens.
- Use button with a while loop to make LED blink when button is pressed.
- Use the variable to gradually raise the brightness of LED with each button press.

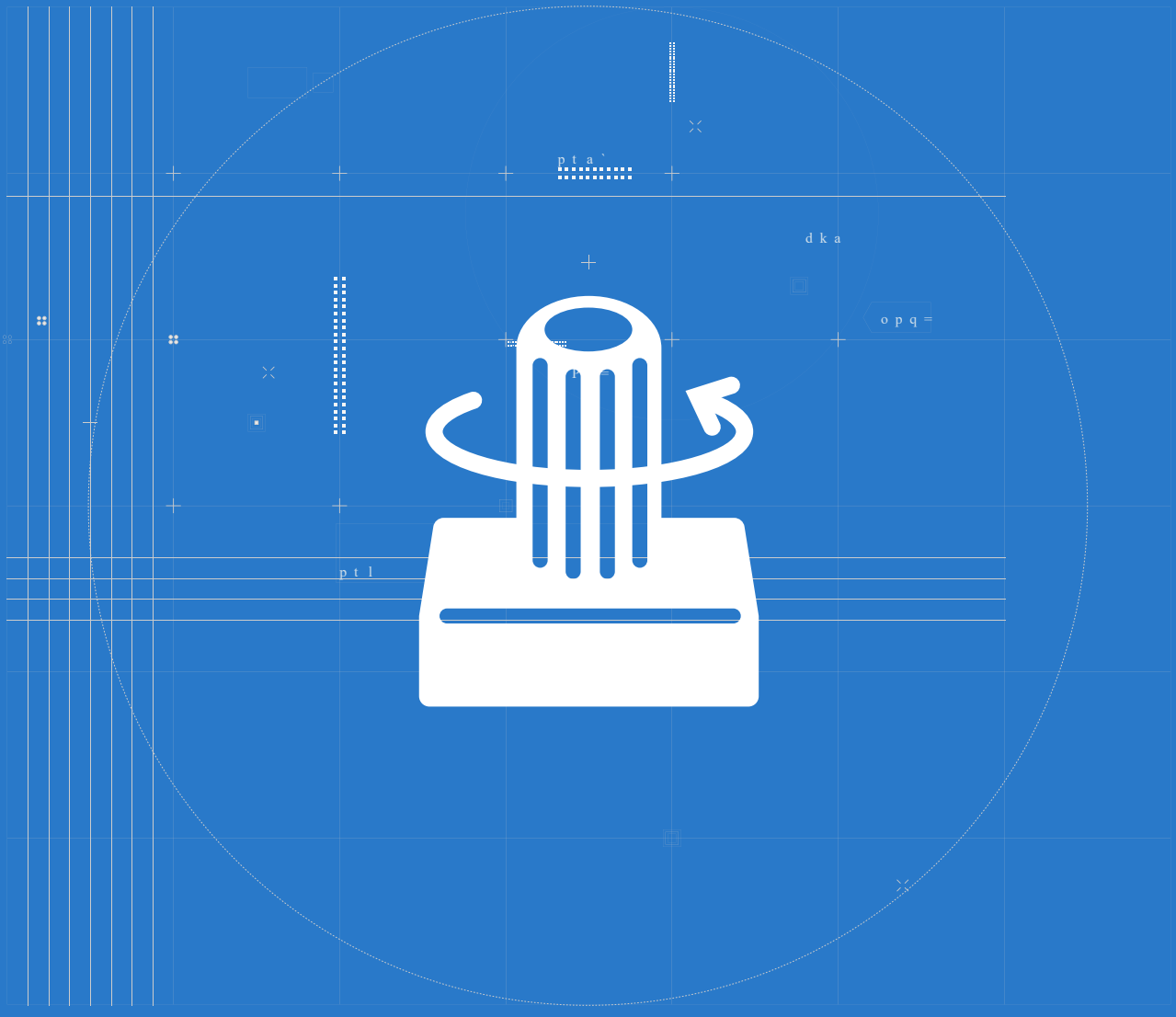



GROVE

Lesson 5

The Potentiometer Keeps on Turning

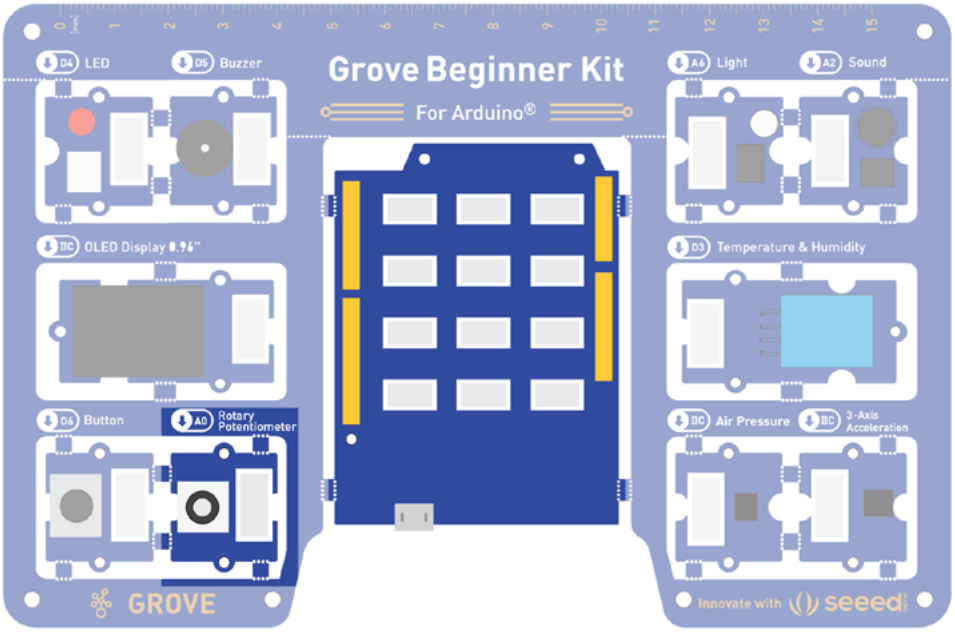
The world would be a very dull place if we only had two colors. Likewise, if we only were able to check for absence(digital 0) or presence(digital 1) of current, the things we could make would have many limitations. There are many values in physical world that cannot be measured by just an on/off state of a switch, for example temperature, humidity, angle or sound loudness. In this lesson we will take a step further, into an analog realm!



The Big Picture

What is potentiometer and how does it work?

Have a look at the potentiometer in Grove Beginner Kit.



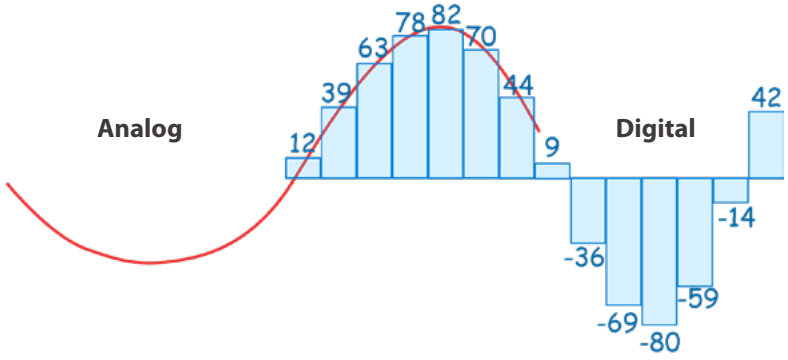
Just like in case with a switch (or a button), you very likely already encountered it in everyday life, but by different names. For example, a handle of a thermostat is a potentiometer, or these handles in sound control equipment:

Potentiometer can change the resistance for the electric current flowing inside of it or change turn a large voltage into a smaller one.

To use scientific terminology, potentiometer can be used as rheostat (variable resistor) or as a voltage divider. If you remember from the previous lessons, changing voltage in the circuit is exactly what we did programmatically to dim the LED. Only in this lesson instead of outputting different voltages with our Grove Beginner Kit, we are going to measure and display the input voltage and see its number representation changes as we turn the handle.

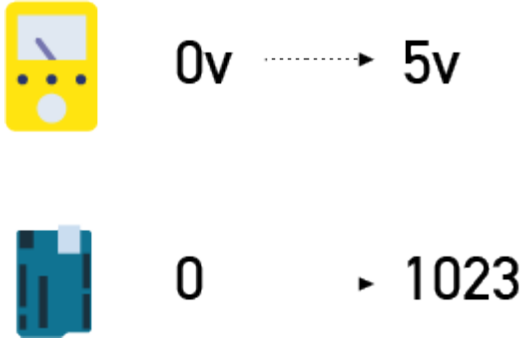
Analog to Digital Converter(ADC)

Speaking about numbers, how does our microcontroller can process the values different from 0 and 1? When a microcontroller is powered from five volts, it understands zero volts (0V) as a binary 0 and a five volts (5V) as a binary 1. What about number 3.45453546 volts?



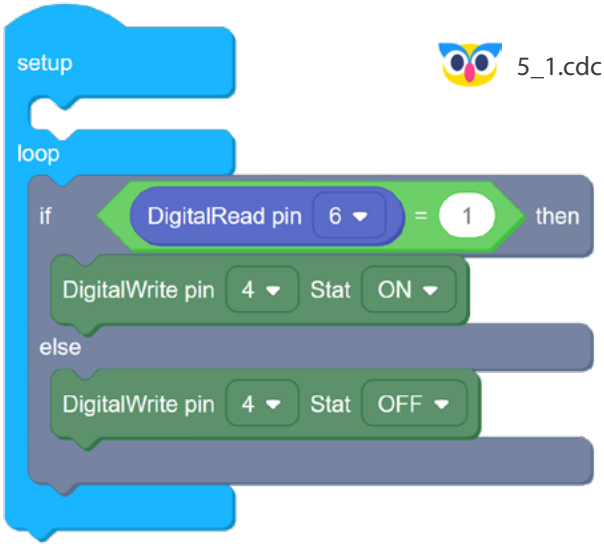
Here we encounter another engineering concept called ADC or analog to digital converter. In electronics, an analog-to-digital converter is a system that converts an analog signal, such as a sound picked up by a microphone or light entering a digital camera, into a digital signal. ADC converter is responsible for measuring the voltage and converting it analog representation(the above mentioned 3.45453546 volts) to digital representation.

Not every pin on a microcontroller has the ability to do analog to digital conversions. On the Seeeduino Lotus board, these pins have an 'A' in front of their label (A0 through A5) to indicate these pins can read analog voltages. And our potentiometer is already connected to pin A0. Means we all ready to start coding!



Task 1: Use DigitalRead and DigitalWrite blocks

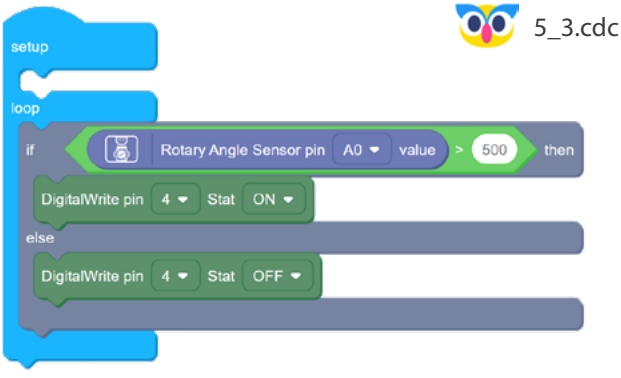
For this lesson, let's start with a little bit of warm-up to better understand the distinction between digital and analog signals. Using blocks in Input and Output categories write the following program.



Once you upload the program, Press the button on Grove Beginner Kit. You will see that the result is absolutely the same with the second program from lesson 4. Only this time we are using digitalRead and digitalWrite function blocks directly. This is why we have to add EQUALS 1 - digital signal can be either 1 or 0, where 1 signifies the presence of current flow and 0 - its absence. Same goes for digitalWrite, only here as the name suggests (write) instead of checking for current, we control it.

Task 2: Use AnalogRead and Rotary angle sensor blocks to get the reading from potentiometer

Now let's see what difference there will be if we use analogInput and potentiometer. For next task you can use both analogInput block or dedicated block from Grove Analog tab called Rotary Angle Sensor - they both generate same code and the result will be absolutely the same.

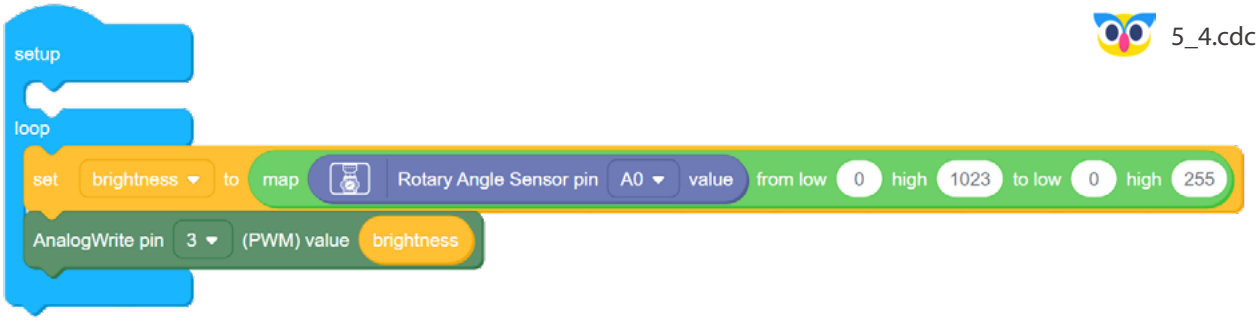


We will still use the dedicated blocks for modules, just for the sake of aesthetics and convenience. The input voltage range of ADC converter in our board is from 0 to 5V - when it receives 0V, it will output value 0. When it receives 5V, it outputs value 1023. So, 500 should be about 2.5V or half of full rotation of the handle. Try out and see the result!

Task 3: Adjust the brightness of an LED with Potentiometer and map function

Now it would be nice to use the Rotary angle sensor (another name for potentiometer) to gradually adjust the brightness of LED. The only problem is - as we saw before, analogOutput function accepts values ranging from 0 to 255 and analogRead function returns values from 0 to 1023. This might be confusing from the first look - the reason for that is in Seeeduino Lotus, Grove Beginner Kit control board, two different and completely separate systems are used for measuring voltage(analogRead, ADC) and outputting regulated voltage(analogWrite, PWM). So we will need to utilize map function, which will help us to re-map values in range 0-1023 to another range 0-255.

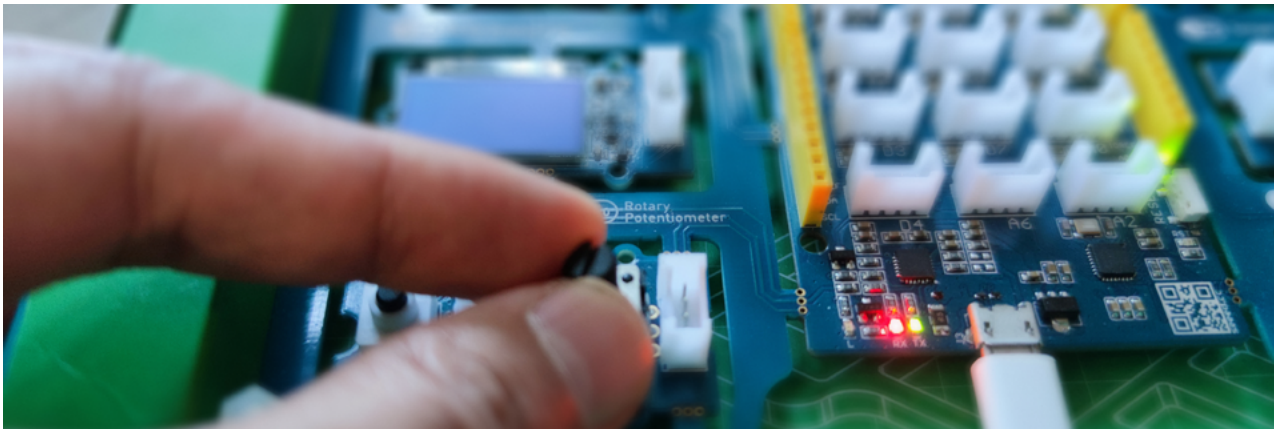
5_4.cdc

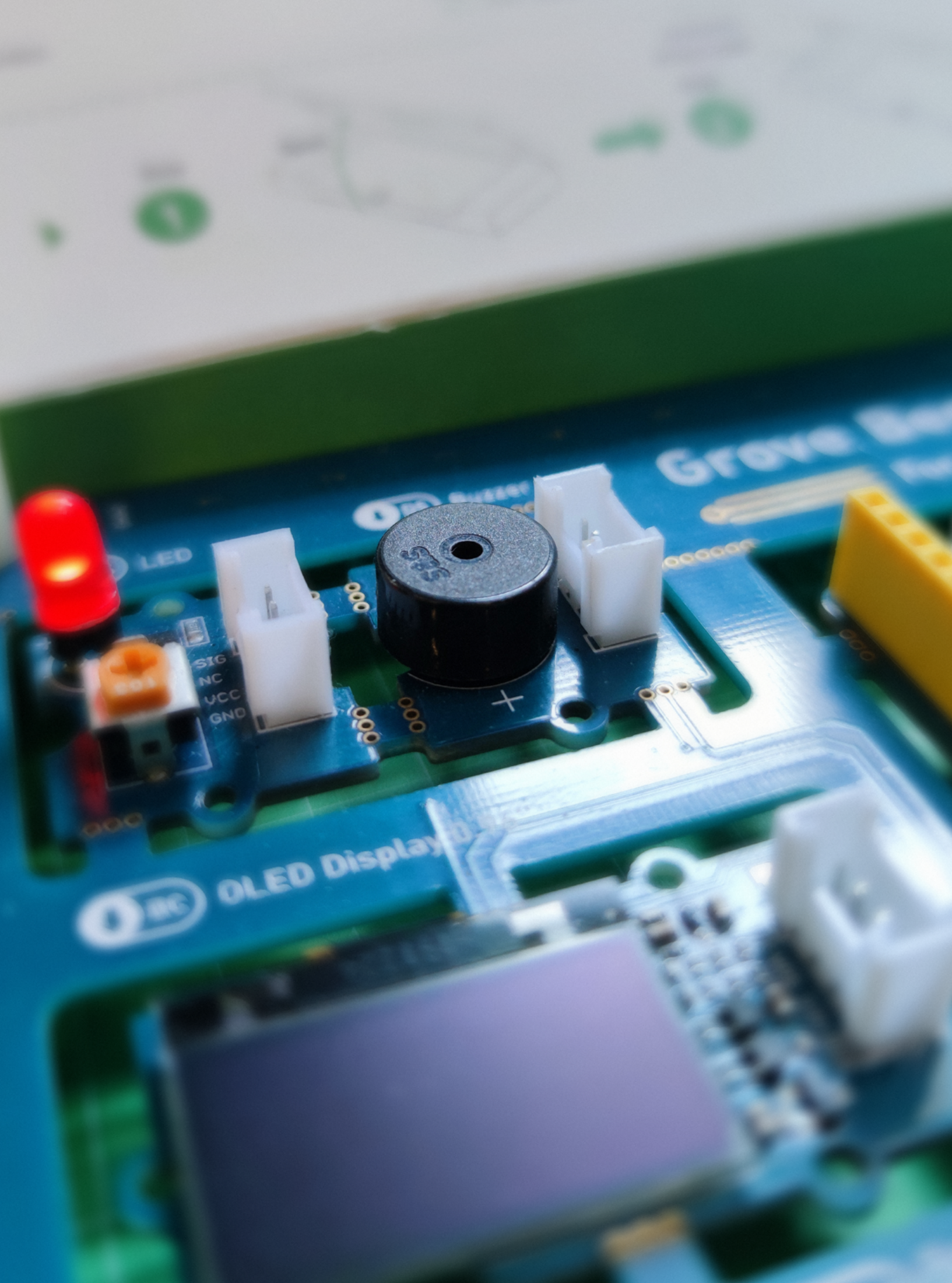


Again, remember it doesn't make a difference if you use analogWrite block or dedicated LED pin set to [number] block - both of them have the same effect.

Outside the Box

- What is the minimal voltage difference ADC converter on our board can measure? Calculate yourself, knowing that it can measure voltages between 0V and 5V and outputs values ranging from 0 to 1023.
- Re-use the code from the last lesson to create a program that would allow switching the LED light on/off and controlling its brightness with potentiometer.



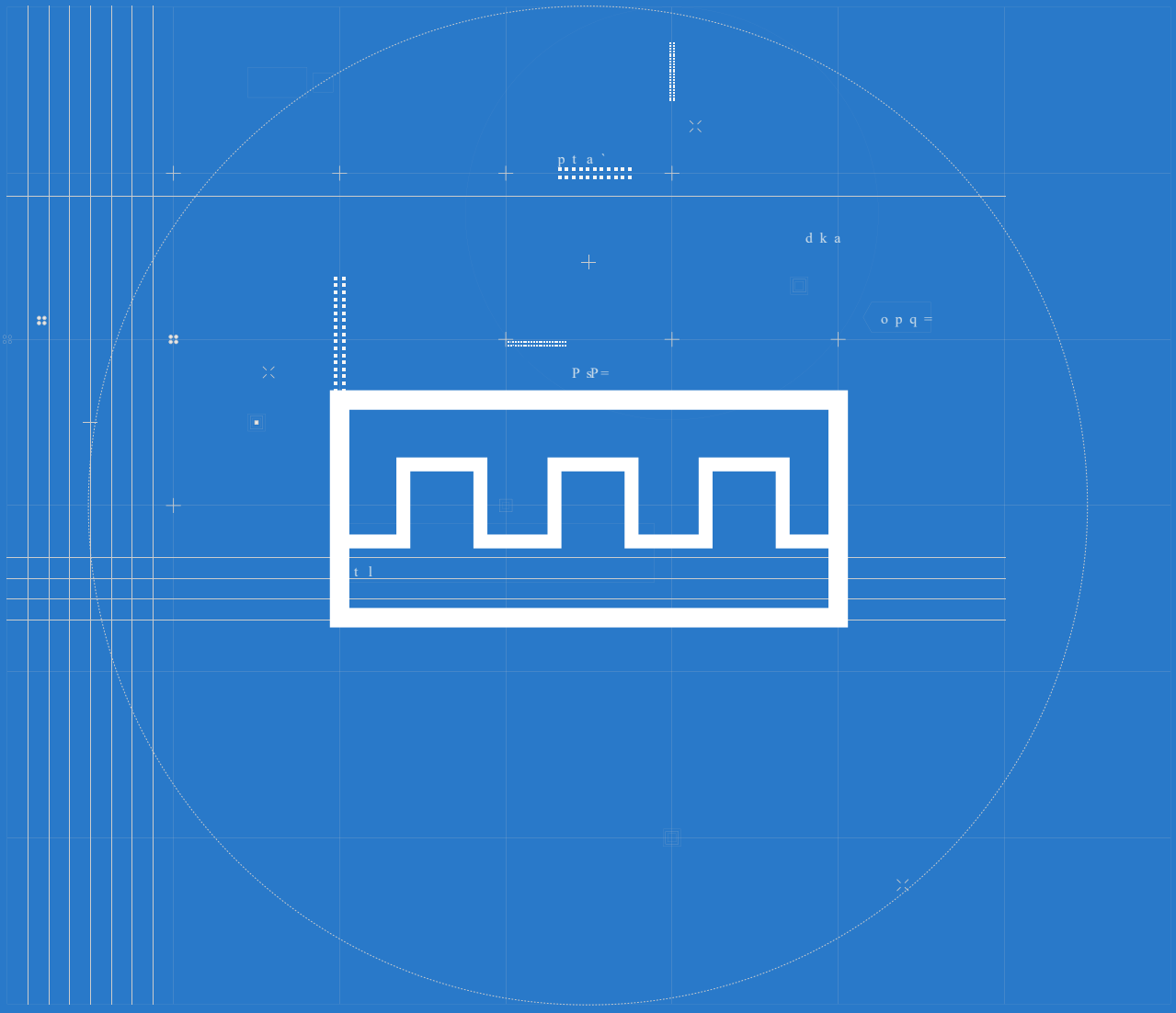

LED
SIG
NC
VCC
GND
OLED Display

Lesson 6

Morse Code

.....

or SOS in human speak - the universal international code for disaster. In popular usage it is associated with phrases such as "Save Our Souls" and "Save Our Ship", but in fact it is just a distinctive Morse code sequence that is not an abbreviation for anything. What is Morse code exactly? How can we communicate using Morse code? In this lesson we learn the basics of producing sounds with Grove Beginner Kit and make our own secret messages!



The Big Picture

Morse Code

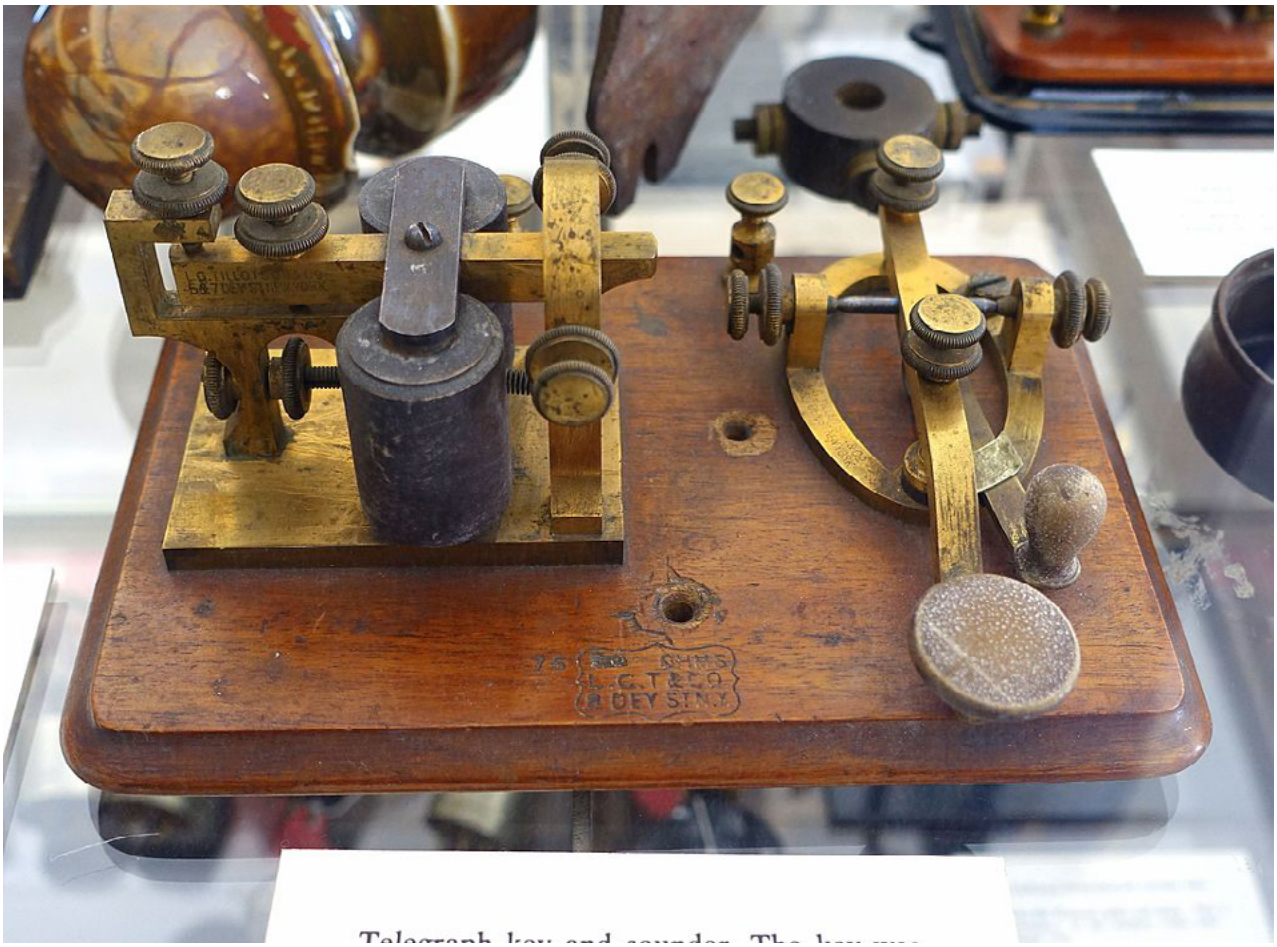
Morse code is a method used in telecommunication to encode text characters as standardized sequences of two different signal durations, called dots and dashes or dits and dahs. Morse code is named after Samuel Morse, an inventor of the telegraph.
The International Morse Code encodes the 26 English letters A through Z, some non-English letters, the Arabic numerals and a small set of punctuation and procedural signals (prosigns). There is no distinction between upper and lower case letters. Each Morse code symbol is formed by a sequence of dots and dashes. The dot duration is the basic unit of time measurement in Morse code transmission. The duration of a dash is three times the duration of a dot. Each dot or dash within a character is followed by period of signal absence, called a space, equal to the dot duration.

Samuel Morse

SOS, the standard emergency signal, is a Morse code prosign.

With this kind of coding, people can send information in many ways, such as by tapping. Or as shown in the figure below, use the on / off of the signal lamp and send it by light.

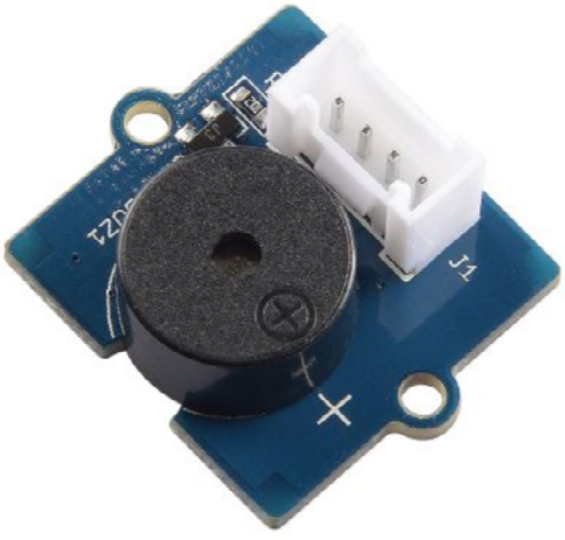
Of course, the most used is in the underdeveloped period of communication, people use Morse code through radio for long-distance information transmission. The old equipment below is the early transmission equipment.

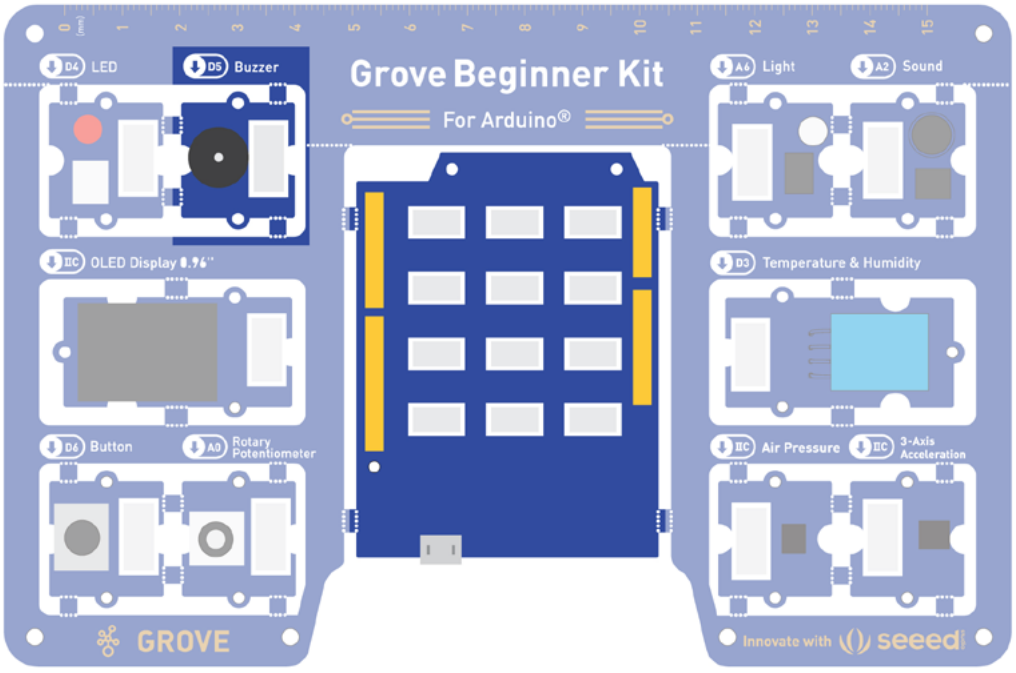


If you are interested in Morse code, you can visit the following website, which can translate the letters and numbers you input into Morse code, and provide download of sound files.
https://morsedecoder.com

How does buzzer work

The working principle of buzzer, included in Grove Beginner Kit is similar to the working principle of the loudspeaker. It creates the sound wave by rapidly vibrating a surface - in case with loudspeaker the surface is a fabric, plastic, paper, or lightweight metal cone (sometimes called a diaphragm).

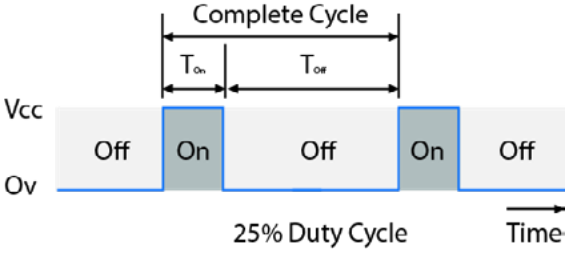




In case with buzzer the surface is made with piezo material. When an alternating voltage is applied to the piezoceramic element, the element extends and shrinks diametrically. We can control the tone of the buzzer by using PWM - or pulse width modulation.

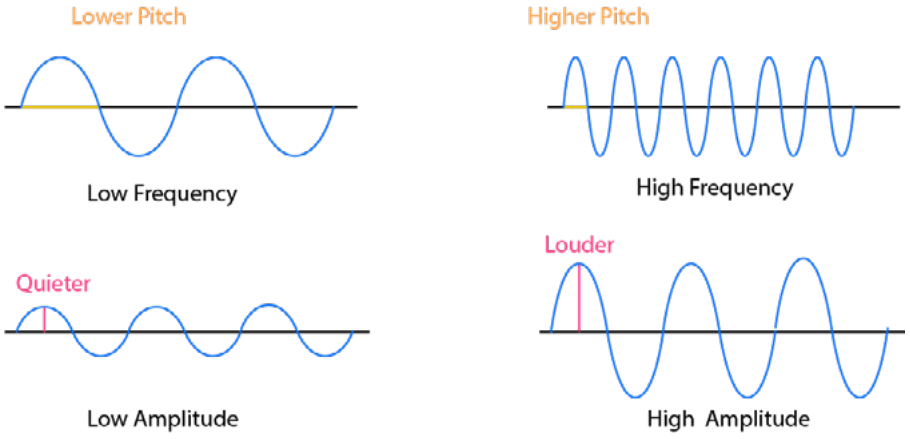
Detailed explanation of PWM

We mentioned PWM before - it as clever technique that allows us to mimic the analog signal with digital signal - i.e. having different voltage supplied to the devices, such as DC motors(to change their speed), LEDs(to change their brightness) or buzzer(to emit different tones). A PWM signal consists of two main components that define its behavior: a duty cycle and a frequency.



Duty cycle is measured in percentage. The percentage duty cycle specifically describes the percentage of time a digital signal is on over an interval or period of time. The frequency determines how fast the PWM completes a cycle (i.e. 1000 Hz would be 1000 cycles per second), and therefore how fast it switches between high and low states. By cycling a digital signal off and on at a fast enough rate, and with a certain duty cycle, the output will appear to behave like a constant voltage analog signal when providing power to devices.
So, in case with buzzer and speaker, changing duty cycle allows us to control the volume

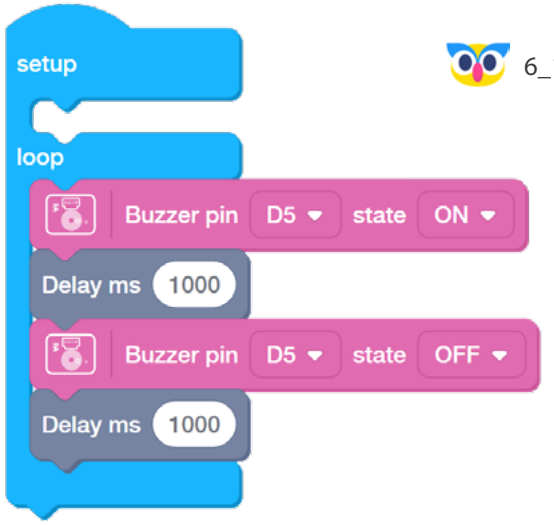
and changing frequency of allows us to emit different notes. Notes are sounds of different frequencies - if you remember from physics lessons sound waves also have frequency.



For example note A4 has frequency of 440 Hz, B4 - 493 Hz and so on. The higher pitched sound is, the higher the frequency. So, in the end it all fit together like pieces of puzzle - PWM duty cycle corresponds to amplitude(or strength) of sound wave and PWM frequency corresponds to sound wave frequency.

Task 1: Emit sound with buzzer

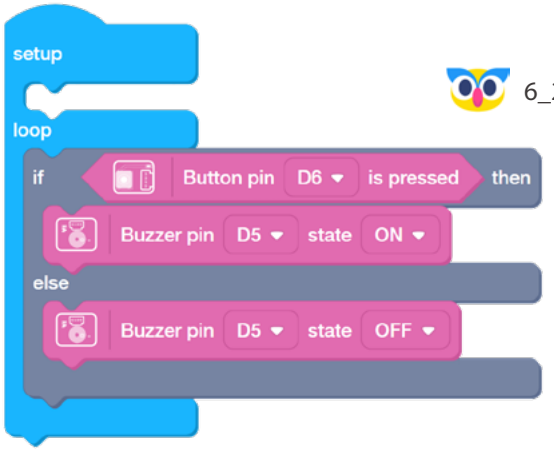
First we're going to try using buzzer as is - with the default tone. Create a program with the following blocks that would emit sound with buzzer for 1 second followed by one second on silence.



6_1.cdc

Task 2: Use button and buzzer to make Morse code machine

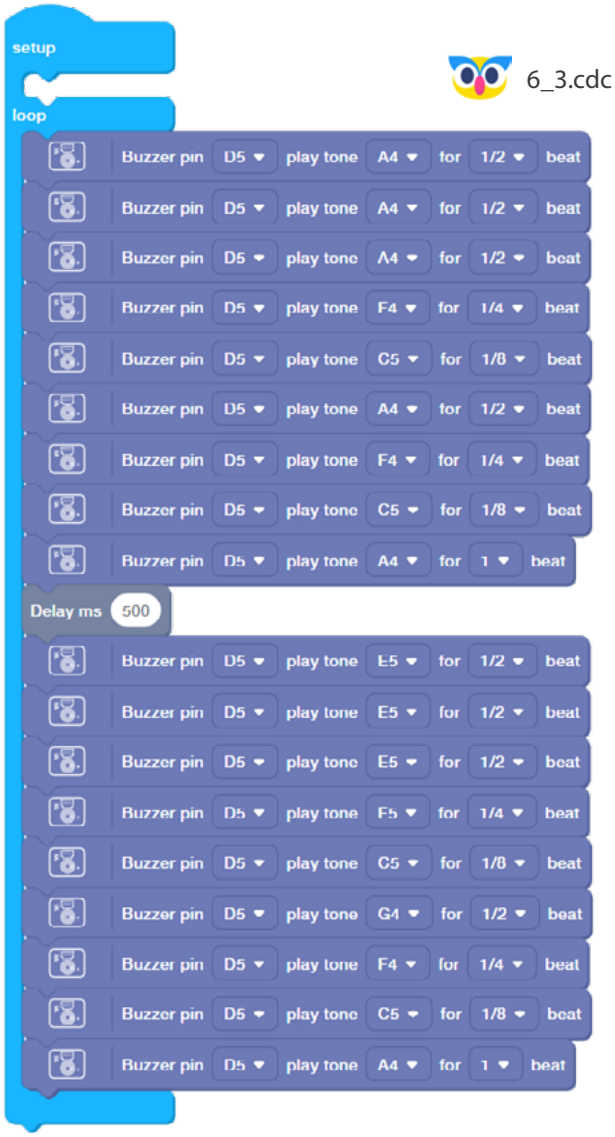
Next we will implement a simple Morse Code generator by using a Button in addition to Buzzer. Exchange the messages with your partner - use the look-up tables if needed.



6_2.cdc

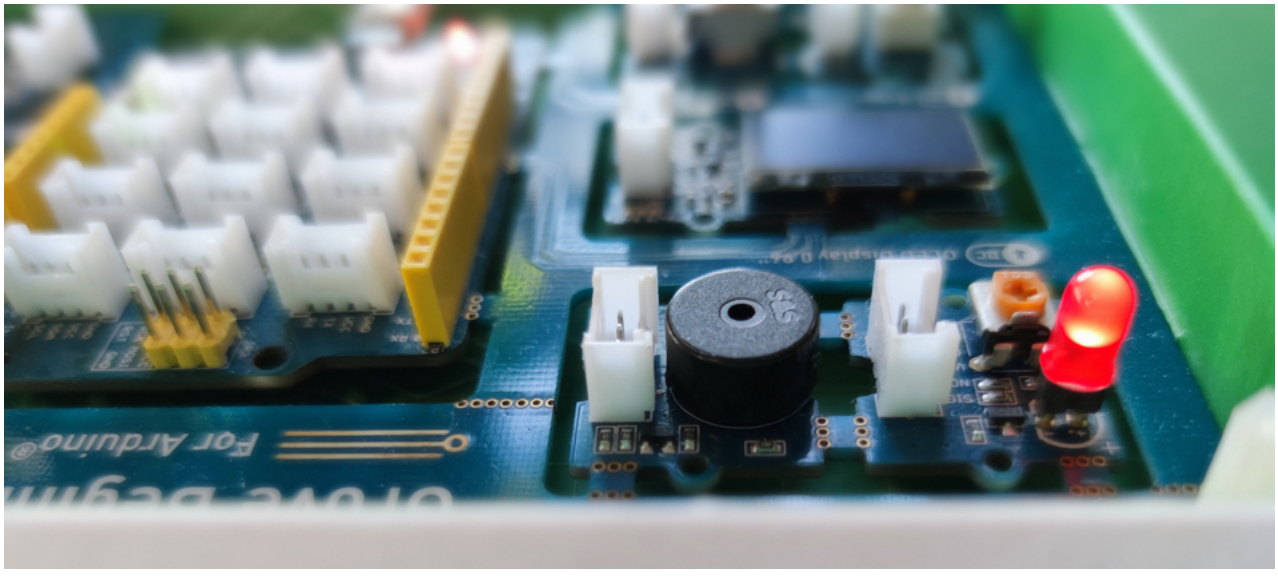
Task 3: Make a song

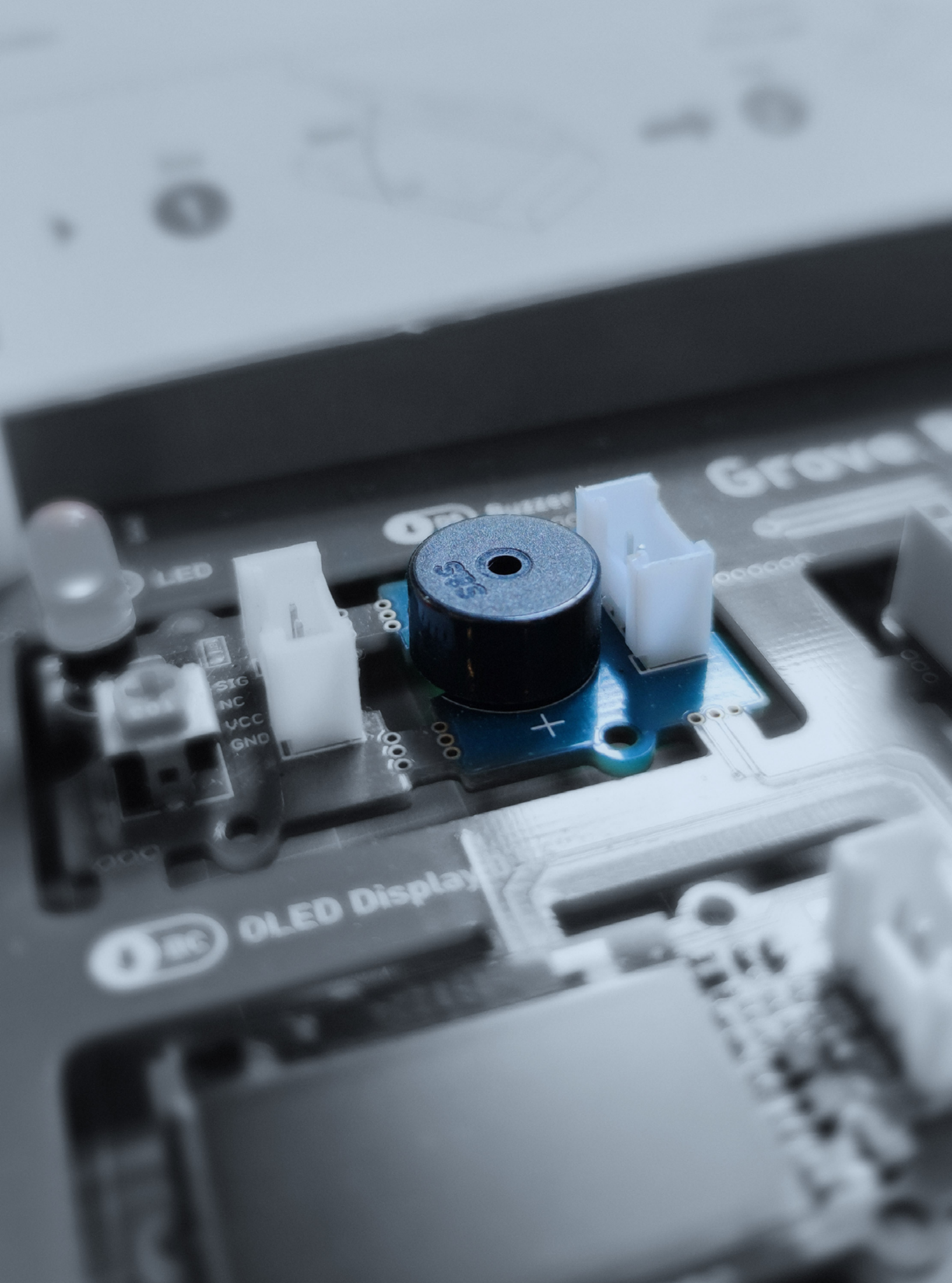
Finally we're going to try emitting sounds of different frequencies with the help of this block. Make the buzzer play a simple tune following this notation:
It will take a few minutes to arrange the blocks, but we promise it will be worth it.

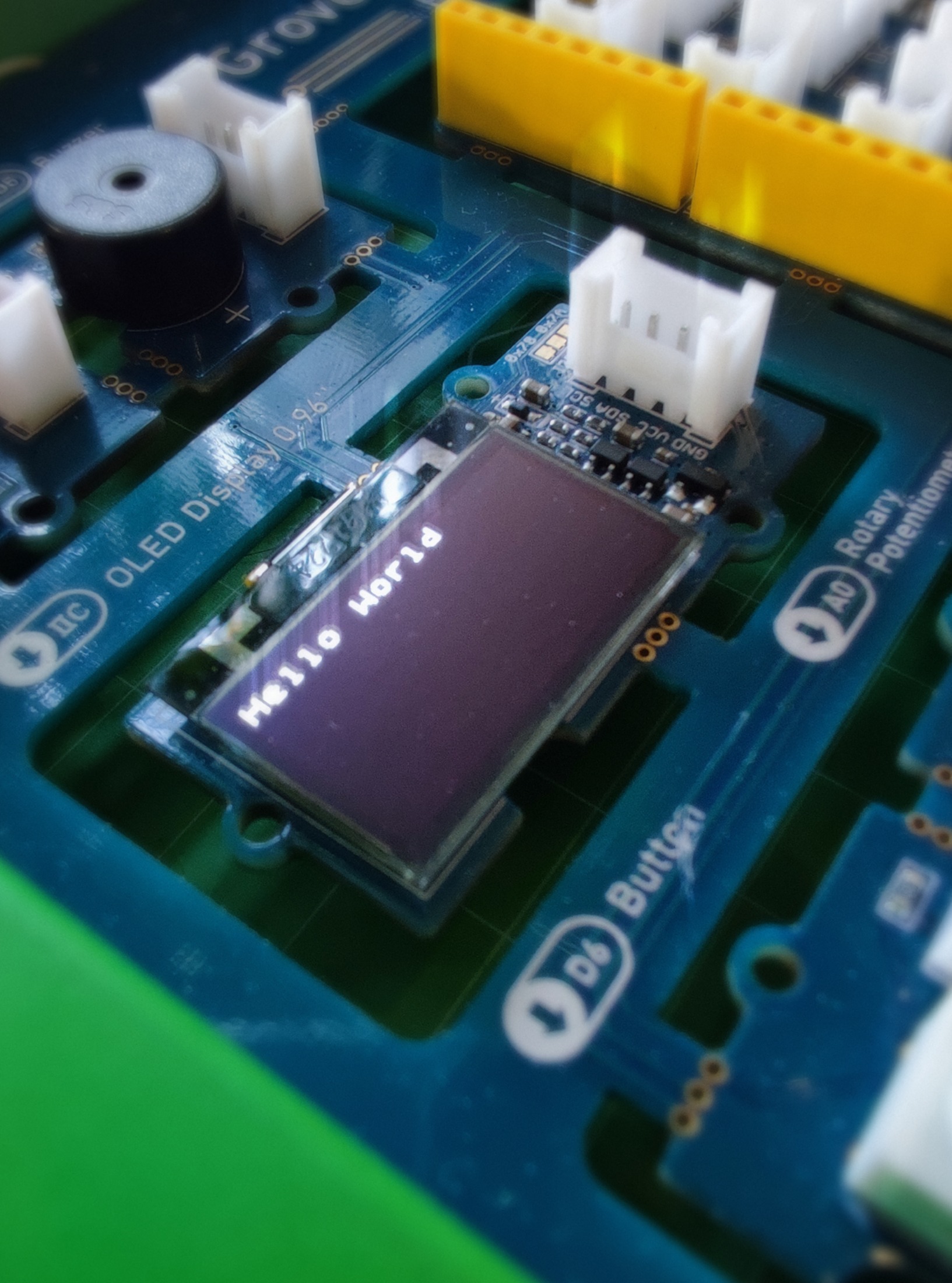


Outside the Box

- What if you try connecting buzzer to a different pin with Grove Cable? Will it still work?
- You can use the code from the last lesson about analog potentiometer to change the tune of the buzzer by turning a handle.
- Another things to try is to make your Morse code machine emit both sound and light by adding LED light.



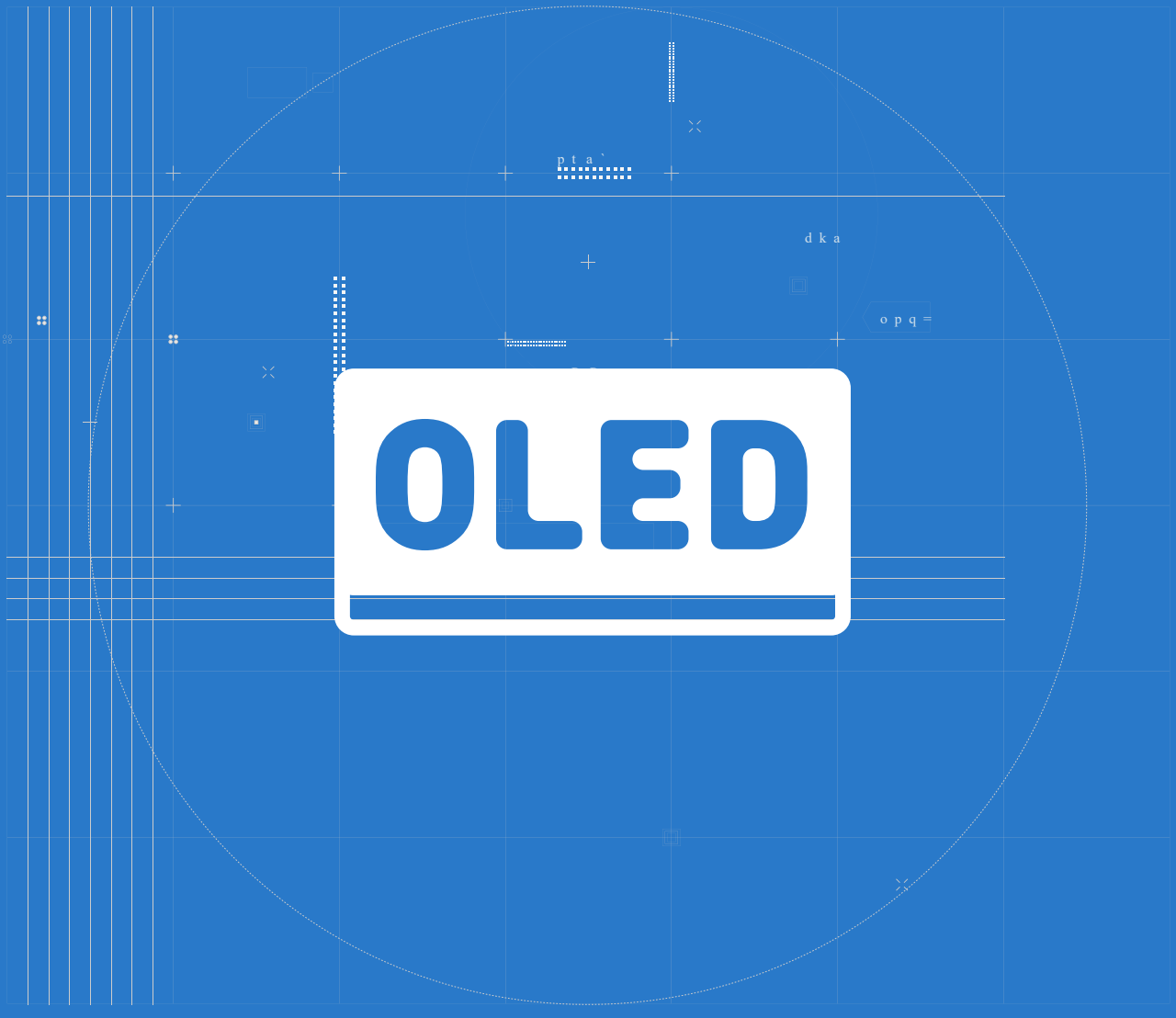

LED
SIG
NC
VCC
GND
OLED Display


Hello World
OLED Display 0.96"
GND VCC SDA SCL
Rotary Potentiom
Button
D5
A0
D2

Lesson 7

Motion Picture

So far we have only used relatively simple output modules to interact with the user - LED module and buzzer. These two are enough if your only objective is to notify the user about something, for example to give a warning when water runs out or the door is closing. What if your application scenario requires richer interaction? Take weather monitor for example - communicating with Morse code is fun to try, but it would be quite painstaking to use it for temperature/humidity information output (- . -- .--. . .-. .- - ..- .-. . / .. ... / ..--- ----- / -... . --. .-. . . ... / -.-. . .-.. ... .- ... you can try translating it to English out of curiosity).It is time to move to graphical interface.

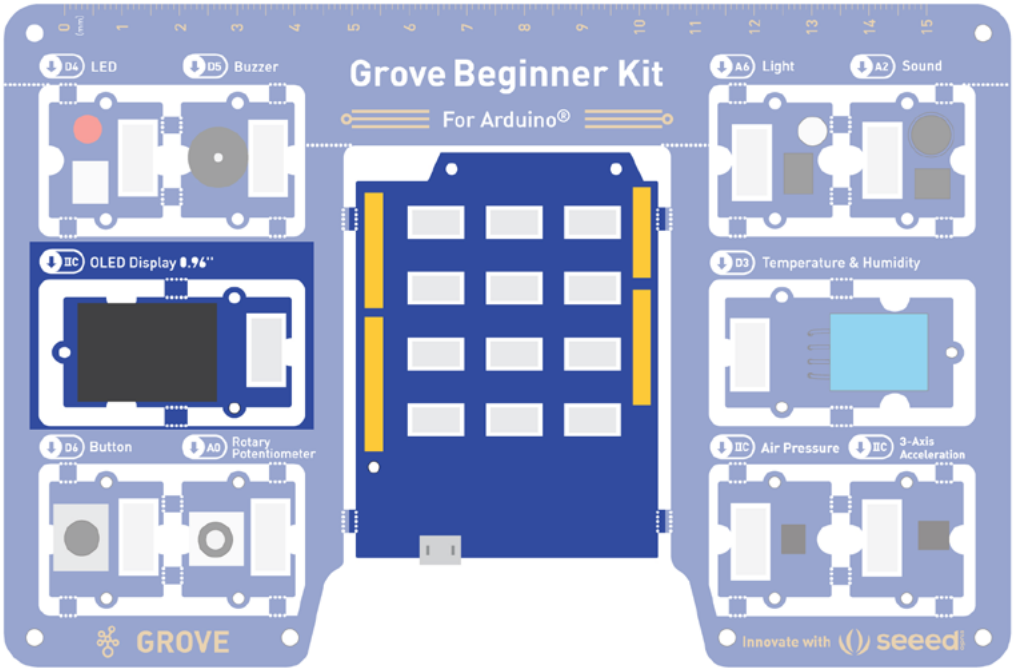


The Big Picture

How does an OLED screen work?

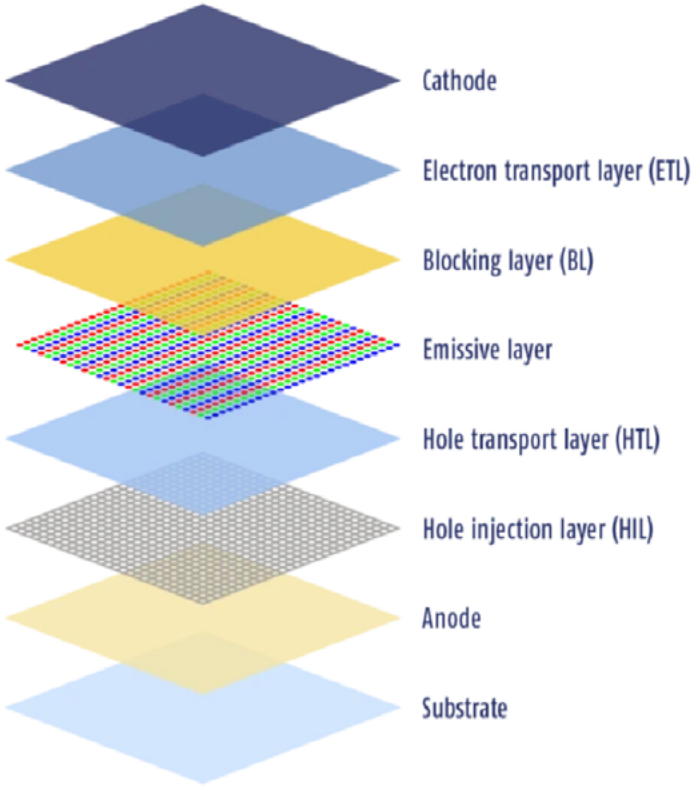
We already are familiar with digital and analog signals. Before the distinction was quite clear - if we needed to make a simple binary action, such as switching on or off or read a binary state we were dealing with digital signals. If the task involved reading a continuous value from sensors, then analog signal would be involved.

Things get complicated with the next module we're going to use - OLED screen. First of all, what is its working principle? How does it convert the electrical energy into patterns we see on the screen?

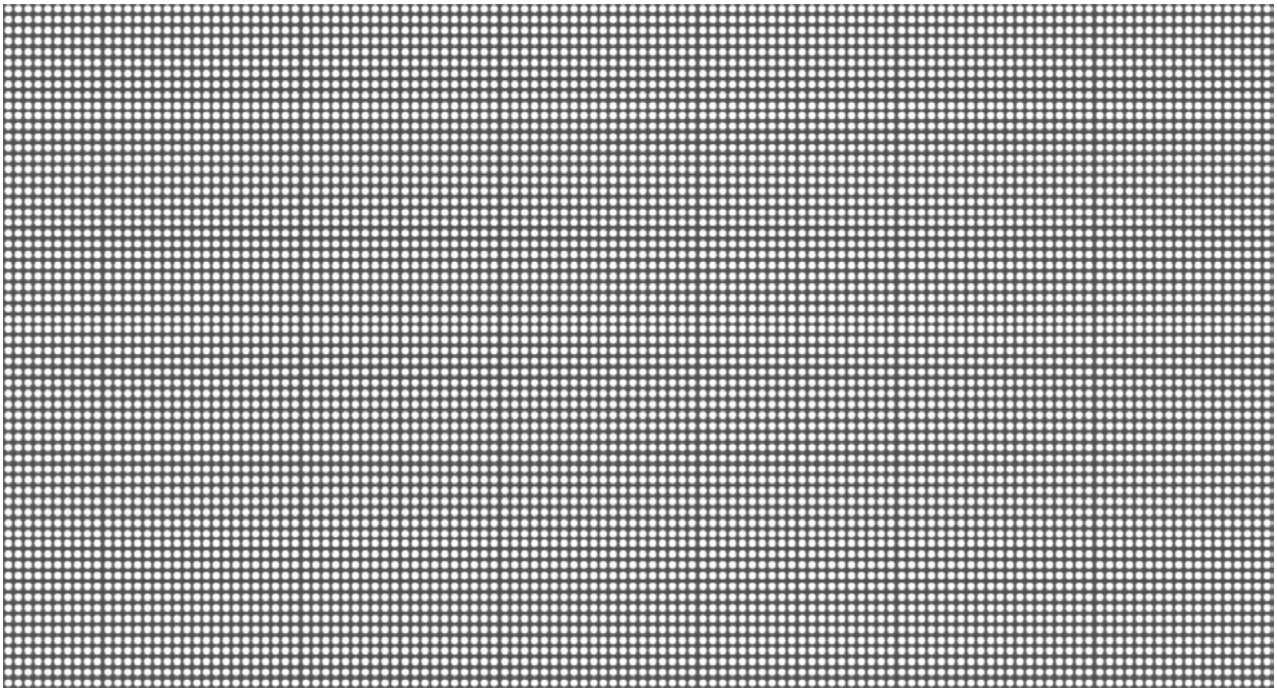


OLEDs work in a similar way to conventional diodes and LEDs - these are made with two slabs of semiconductor material, one slightly rich in electrons (n-type) and one slightly poor in electrons(p-type).

When we allow the current flow in circuit electrons cross border between these two materials and release surplus energy, giving off a quick flash of light. All those flashes produce the dull, continuous glow. A simple OLED is made up of six different layers.



To make an OLED light up, we simply attach a voltage (potential difference) across the anode(positive terminal) and cathode(negative terminal). OLED screen is made of hundreds or thousands of individual OLEDs (128 x 64 = 8192) and by turning each of them individually we can draw pictures and output text in the way we see fit.



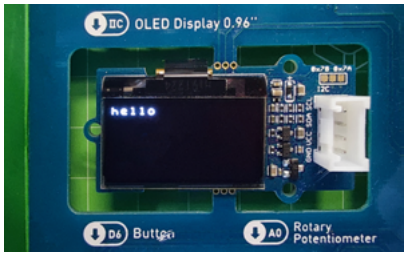
Obviously manual switching the individual OLEDs would be a nightmare. So, to help with that, there is an integrated circuit on OLED screen module that takes care of low-level, individual OLED control, together with software on our microcontroller. All of that allows us to simply type or draw pictures in Codecraft interface and then have them displayed on the screen without too much hassle.

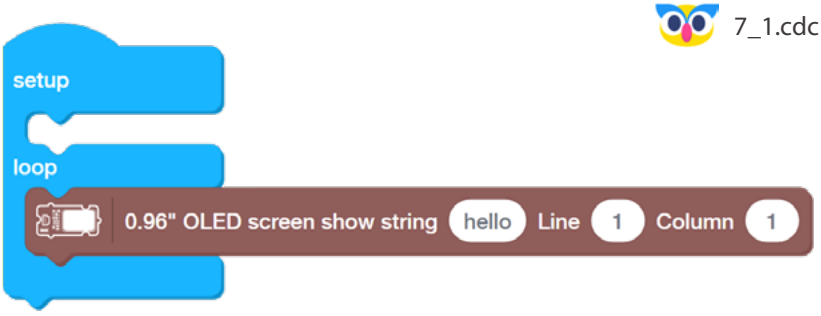
Task : Display text and images on OLED

Step 1: Display a word

Let's start by displaying a phrase on OLED screen with this block.

7_1.cdc

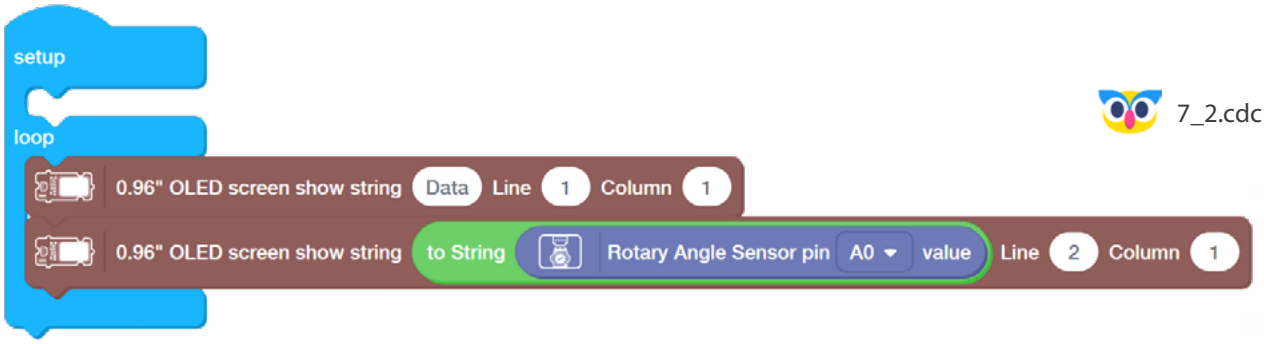




Step 2: Display data from Rotary Angle sensor

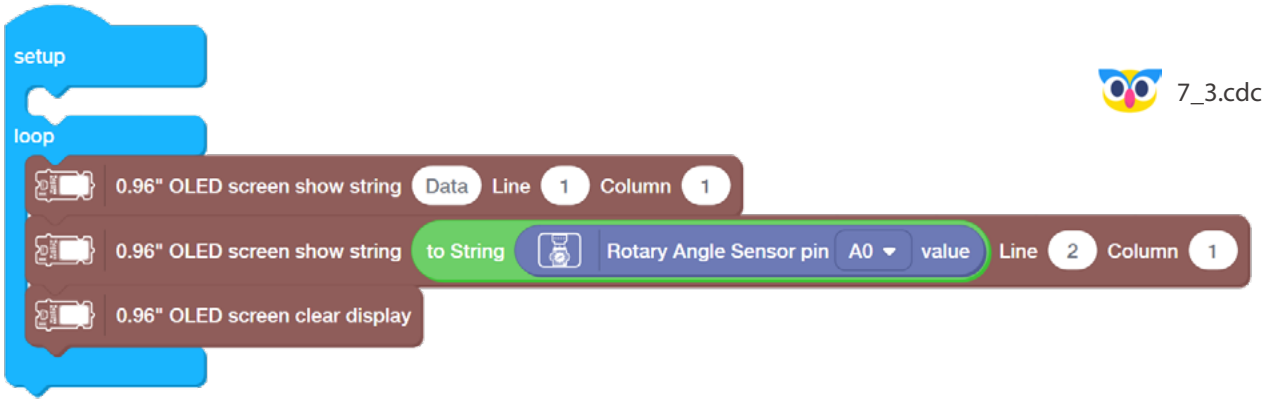
Now, where we can display simple hello, we can also display data from sensors and any other text data we want. Let's try displaying the numerical data from potentiometer(or Rotary Angle sensor). To display numerical data we first need to convert it to text representation, by using to String block from Operators tab.

7_2.cdc



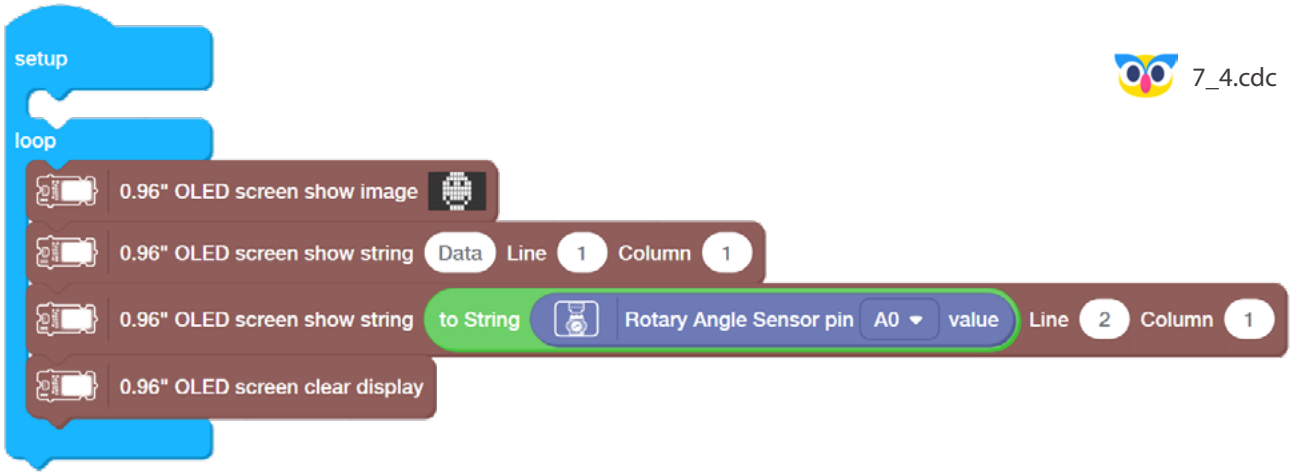
At first look it seems that everything is fine. But then we notice, that after a few turns the numbers start to look a bit strange. We can discover that it is because numbers are not being erased, but instead are being partially replaced by new numbers. Computers have no common sense, so our program did exactly what we asked it to do - it wrote numbers on the screen. We ourselves didn't say anything about erasing these numbers to make a way for new ones. Let's add the clear screen block and see if it can improve things.

7_3.cdc



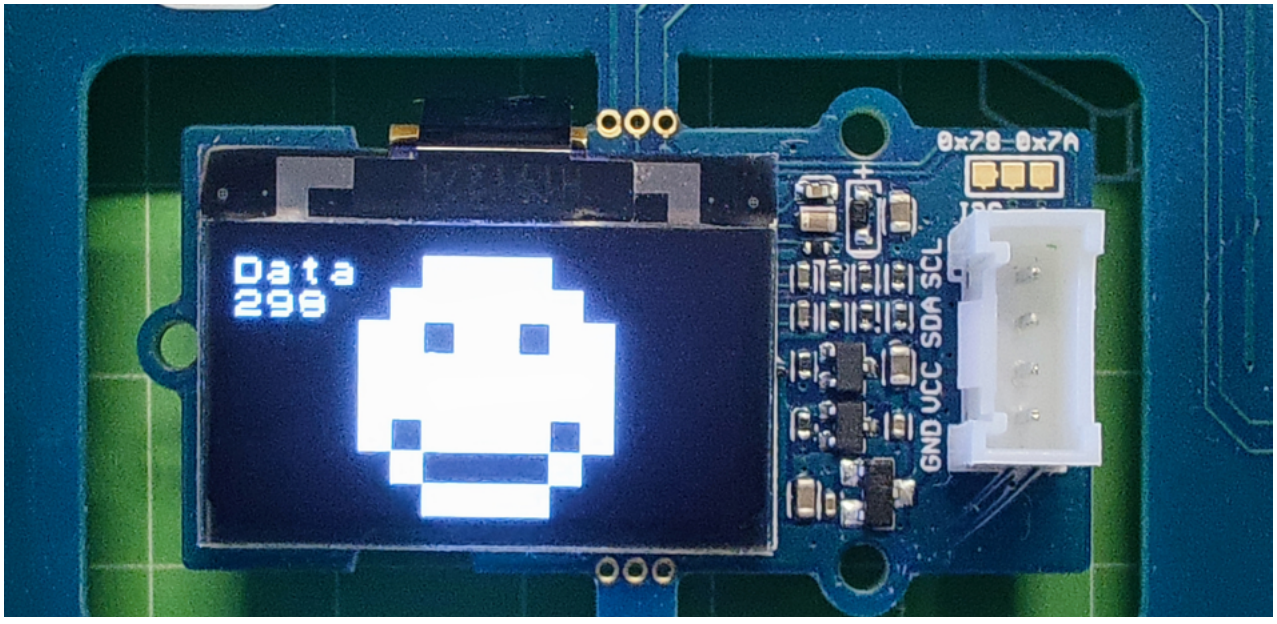
Step 3: Display data from Rotary Angle sensor and a drawing

Much better now. And finally, let's do some drawing.



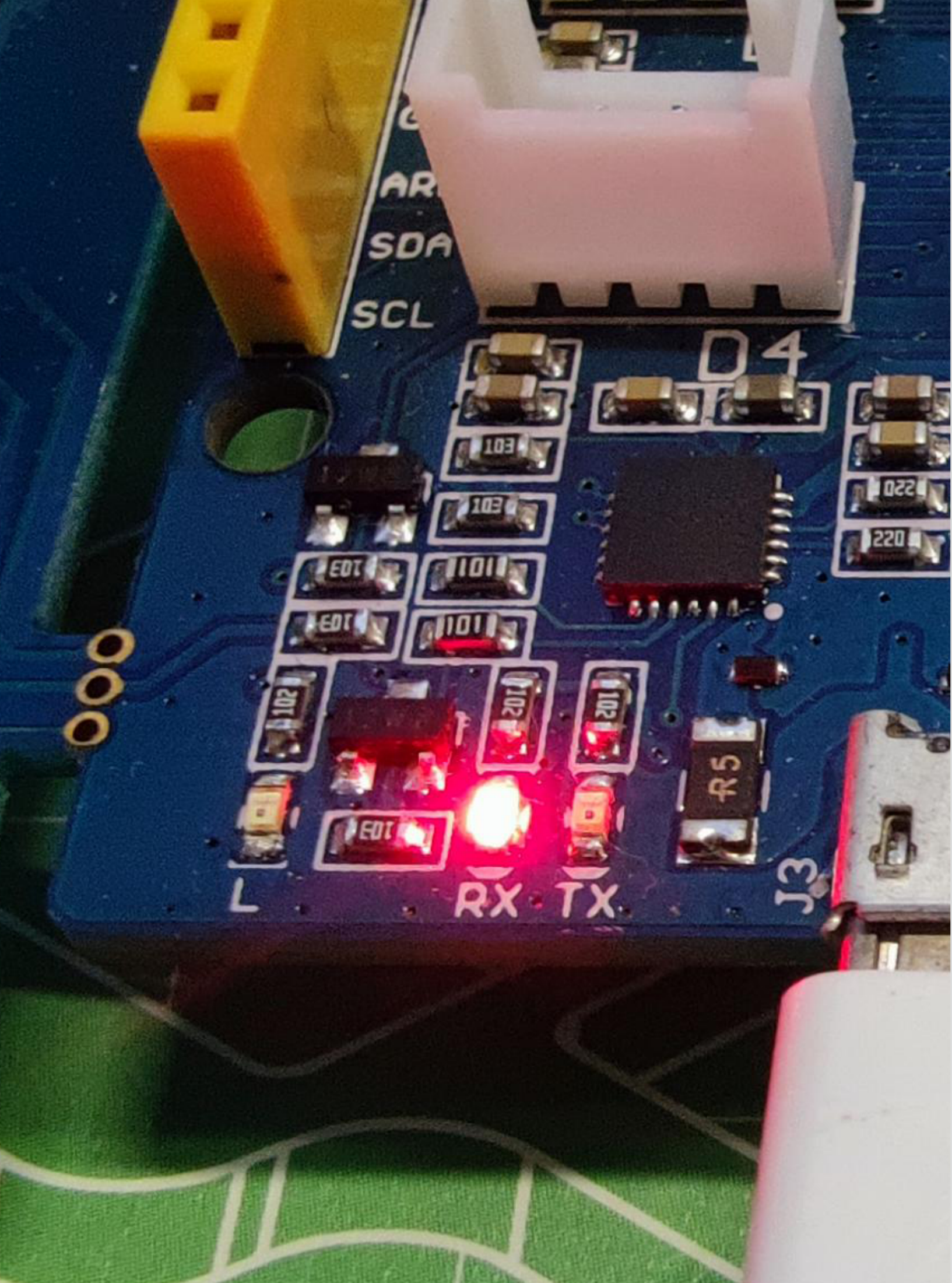
7_4.cdc

We will see that the image of duck disappears with the text data only to be drawn again. While it is not ideal effect, it is one of the limitations of graphical programming environments, such as Codecraft. If we were to write this program in C language we would have much more control over display drawing and erasing functions. Well, all in due time.



⭐ Outside the Box

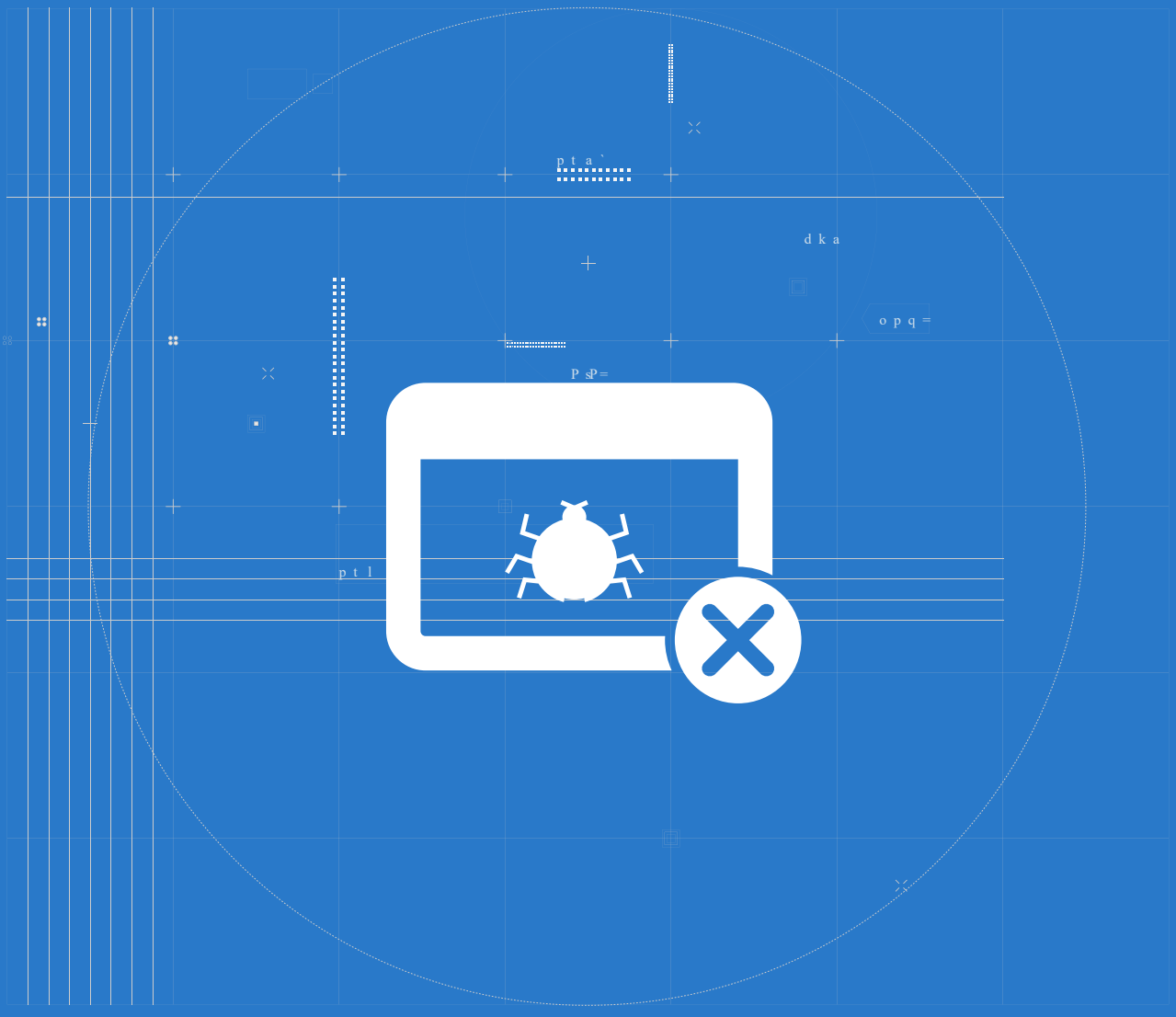
- Implement a program that draws two different images, if button is pressed/released.
- What happens if we don't use to String block? Check and see the result.
- Use while block to write a program that would move text down with each button press until it reaches the bottom.


SDA
SCL
D4
103
103
EDT
101
EDT
101
102
102
102
EDT
R5
220
220
L
RX
TX
J3

Lesson 8

Direct Access

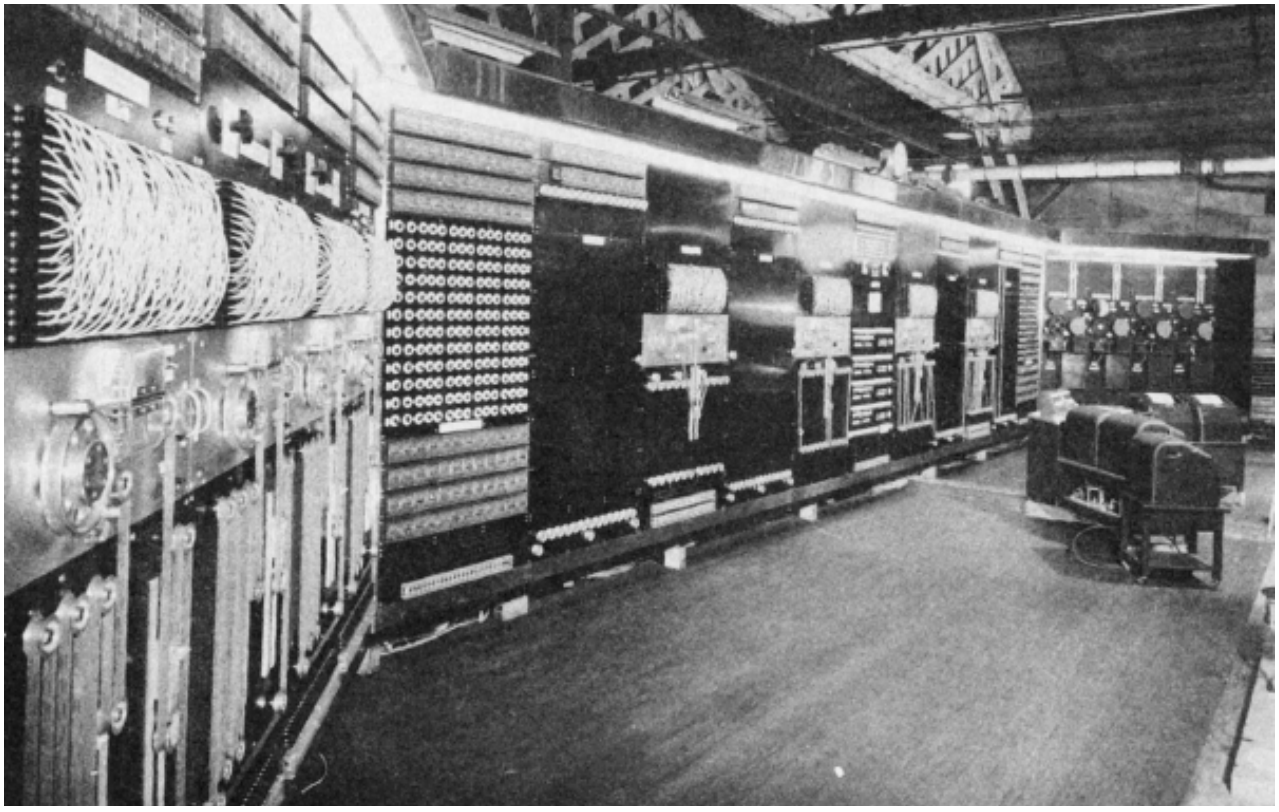
This course focuses on how to debug the program, learn how to use the serial port monitor and serial port chart to obtain the key information of hardware equipment.



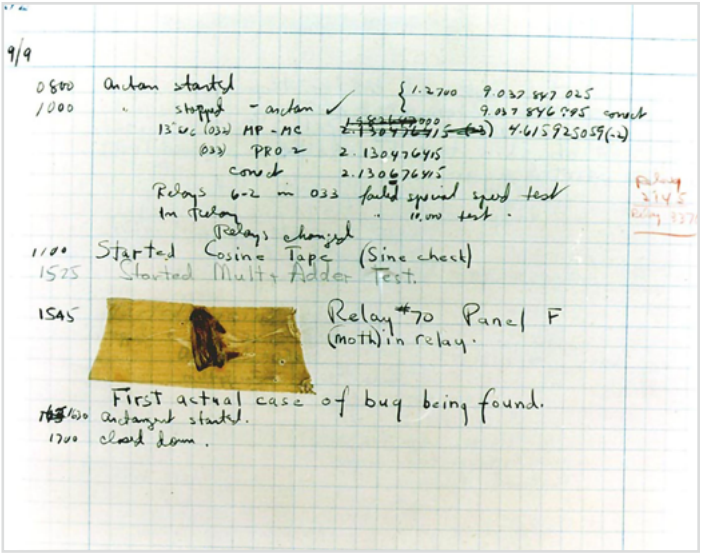
The Big Picture

Program bugs

A long time ago computers were large machines occupying whole rooms of buildings. It was a hot mess of wires, switches and other electronics inside.



Literally quite hot - that meant that in the winter when the weather got colder, the electronics inside these computers provided a good shelter for the insects. Not surprisingly, sometimes an insect would stumble upon a wire and break the whole intricate circuit - this is why we now call the errors in the computer software and sometimes hardware "bugs", because at the dawn of computers, these were REAL bugs, that would cause the malfunctions.



And the process of finding errors or bugs is called debugging. How do we debug in the modern times?

If you see a bug, who you gonna call?

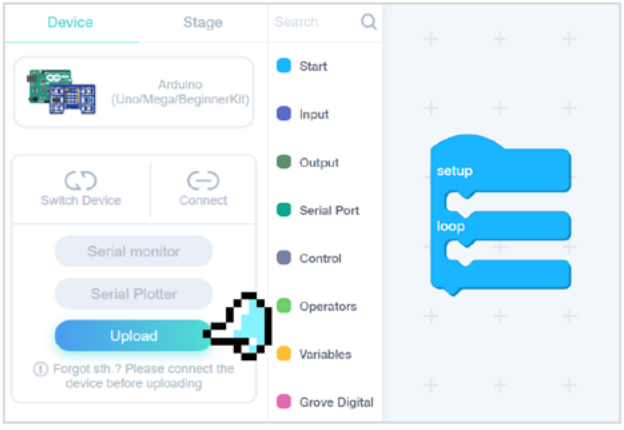
A lot of the times bugs in the program lead to two distinct outcomes:

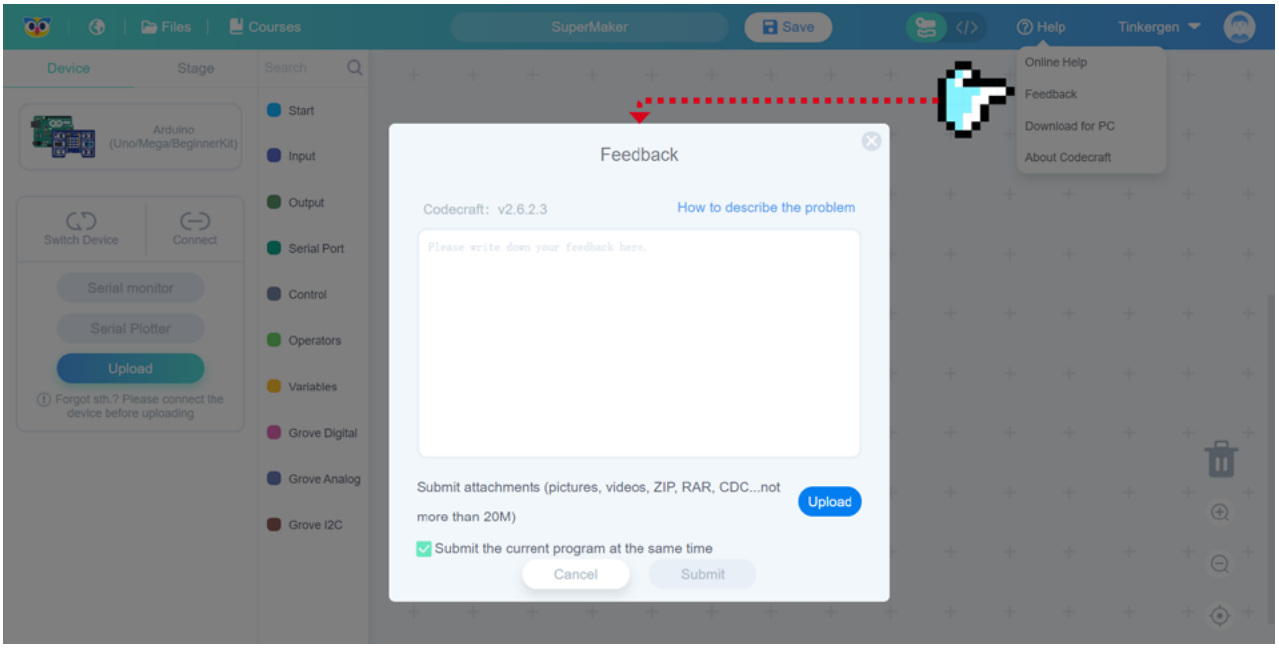
- program doesn't work at all, showing an error code
- program does run successfully, but the end result of execution is not what we want

We have taken care of the first type of bugs for you - our graphical programming software Codecraft will properly structure the code according to the graphical blocks you put in the program, so the likelihood of your program having syntax or usage errors is really low.

In case that type of bugs does happen - as evidenced by Upload failed screen showing after you press the upload button, first thing to do is to make sure that hardware is not an issue - swap Grove Beginner Kit with a fellow student and try uploading the program again. Check the USB cable as well. Make sure you have chosen the correct port when uploading program - your computer can have multiple devices connected.

If you checked all the items above and you still cannot upload your code, but can upload an empty program (with just setup-loop block) - then contact us at techsupport@chaihuo.org - remember to thoroughly describe the problem, attach your program code, describe the environment(OS, Codecraft version - online/offline, version number, which can be found by pressing Help - About Codecraft) or use the online feedback function directly from Codecraft. The more details you provide, the easier for us to "reproduce" the bug - see it on our own computers and proceed to fix it.



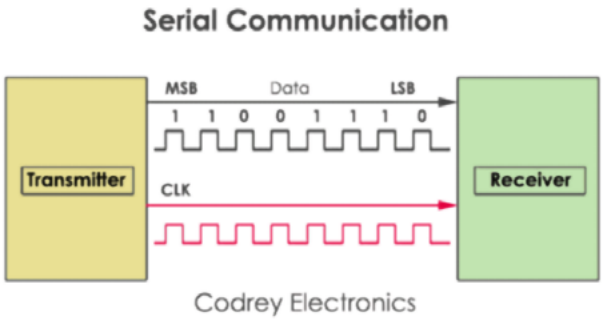


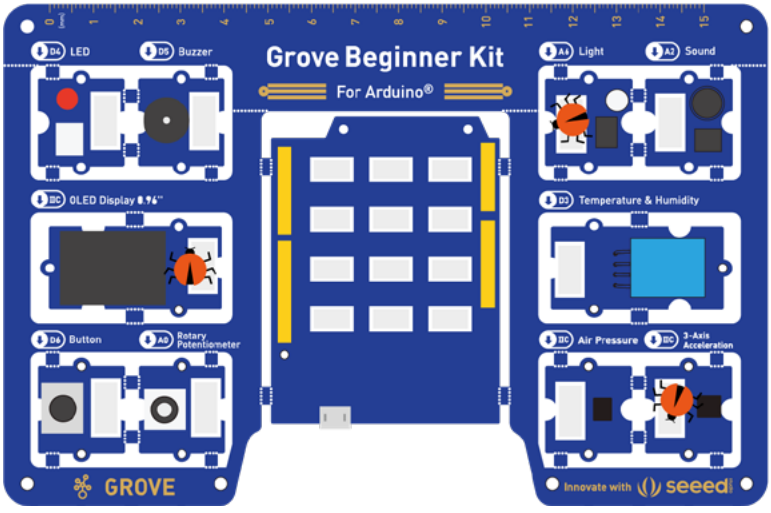
Why do we want to use Serial Monitor?

The second type of bugs you might be able to resolve on your own. Here the main topics of today's lesson come in play - serial connection and serial monitor.

Serial communication is the most widely used approach to transfer information between data processing equipment and peripherals. Every device might it be your personal computer or mobile runs on serial protocol. In serial communication, data is in the form of binary pulses. In other words, we can say Binary One represents a logic HIGH or 5 Volts, and zero represents a logic LOW or 0 Volts.

Serial communication can take many forms depending on the type of transmission mode and data transfer. The protocol is the secure and reliable form of communication having a set of rules addressed by the source host (sender) and destination host (receiver). Serial monitor(and serial plotter) allows us to display the values received from Grove Beginner Kit through serial connection directly on our computer. Why is it important and how it is related to debugging?





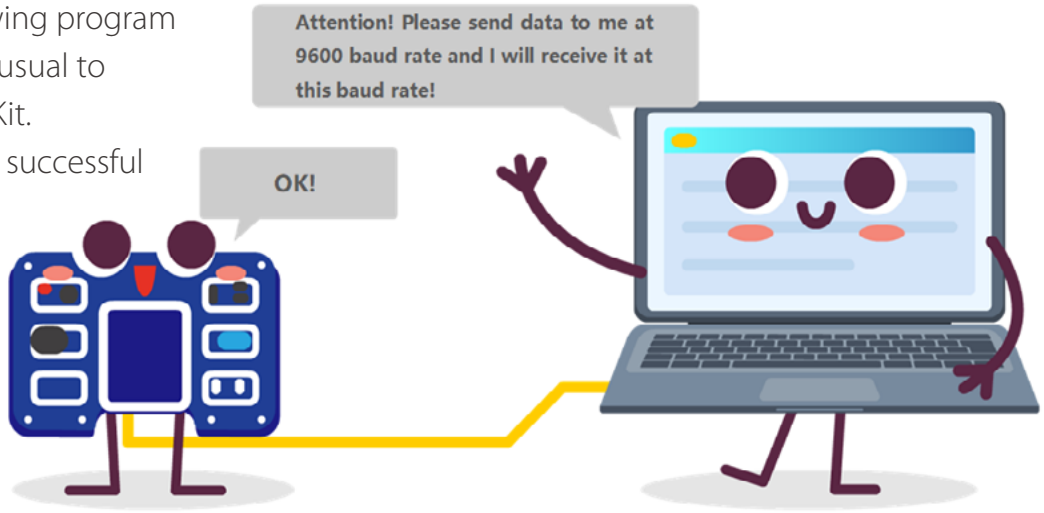
Modern electronics is fairly complex - so it might be difficult to pinpoint exact reason for bug when many components are involved. Imagine the following situation - you made a program for blinking and LED with button press. You have a bug in software, that prevents LED from blinking - but you don't know where bug is.

It can be a problem with LED module, button module or wires(if you used wires to connect). When program is more complex, there are even more systems that can malfunction. Serial monitor allows us to see the value from button - this way making sure that at least button module and the wires are working properly. Let's see how can we use these tools to see the direct output from sensors on the computer screen.

Task 1: Output a word on Serial Monitor

The blocks for controlling serial input/output can be found under the tab Serial Port. Baud rate is the communication speed rate - the faster it is the faster the data is being transferred, but higher baud rates increase the chance of errors during transferring/receiving data. 9600 bits per second is more than enough for most of debugging applications - it is slow, but very stable.

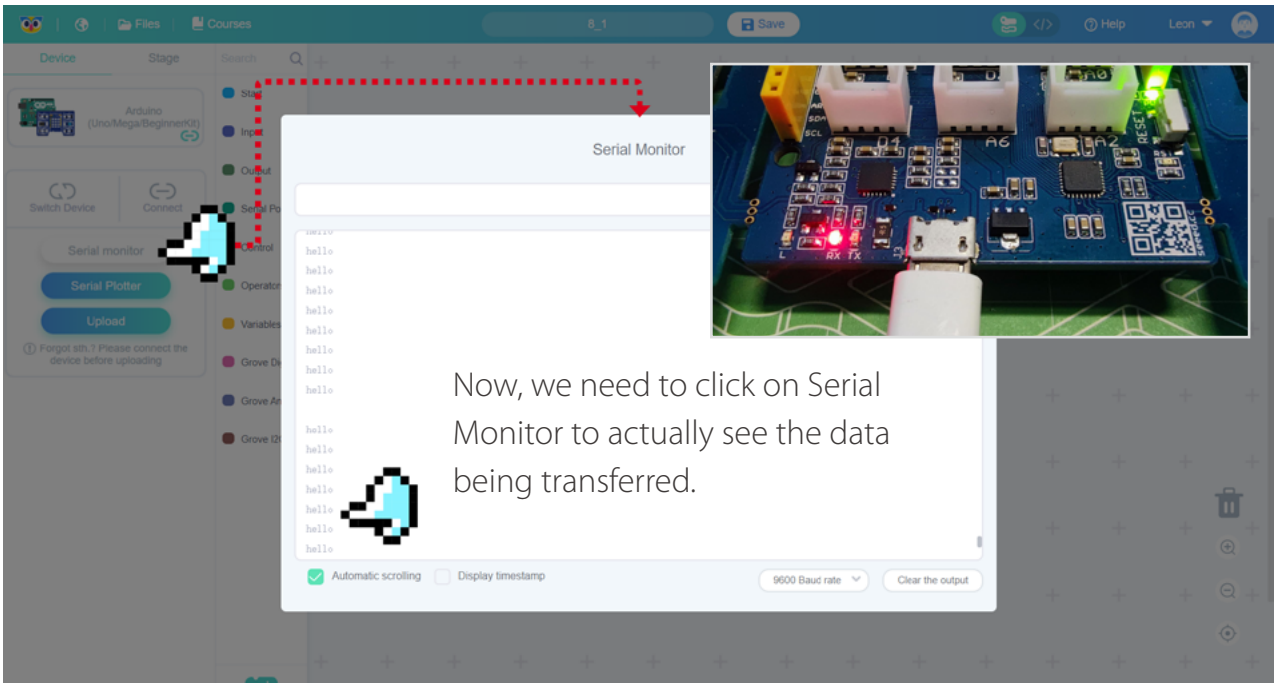
Create the following program and upload it as usual to Grove Beginner Kit.



After upload was successful we need to connect our computer to Grove Beginner Kit in order to start data transmission.

Click on Connect button and choose your Grove Beginner Serial port in pop-up window, then Click Connect. You'll notice that red light started blinking very fast on the mainboard.

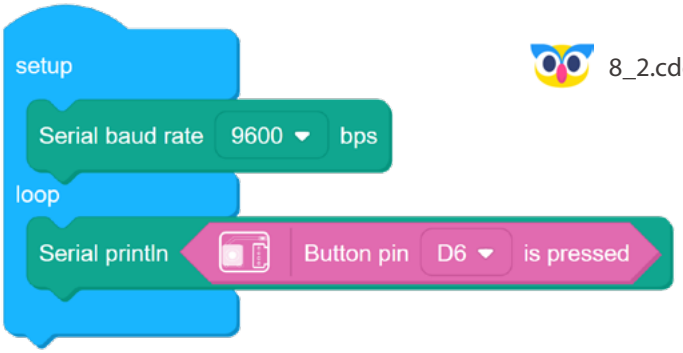
The red light is data Transmission indicator.



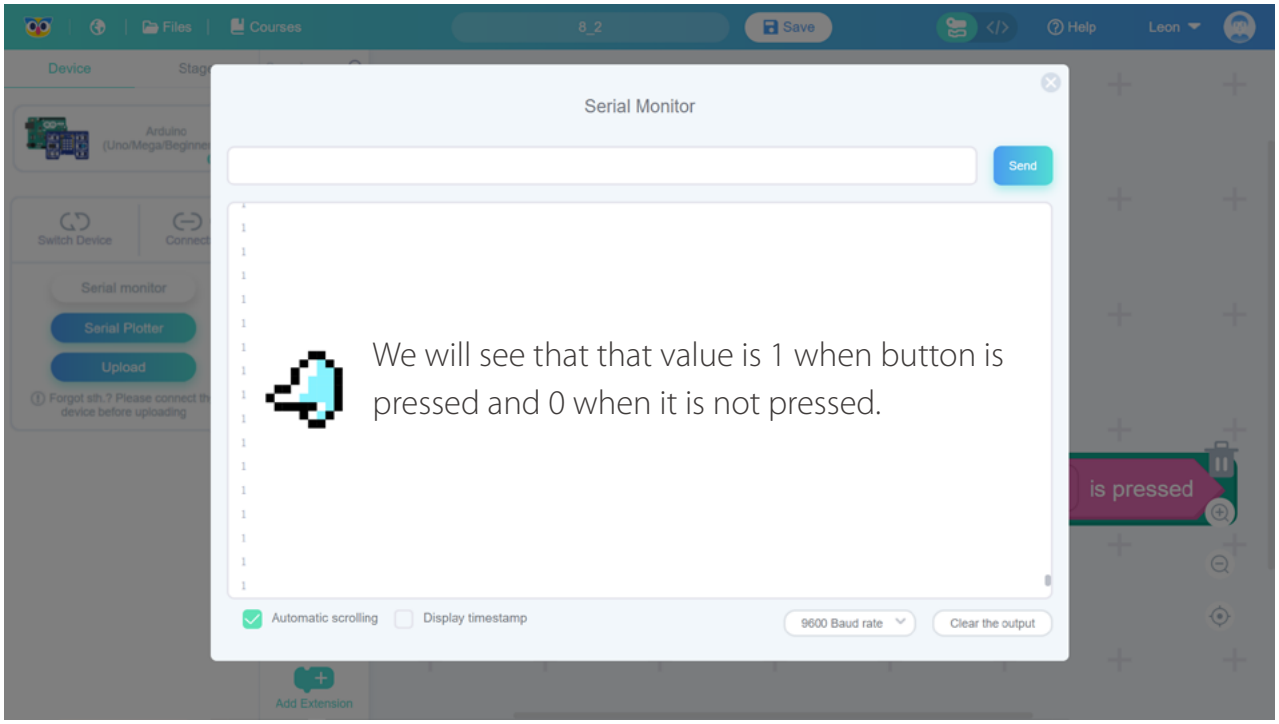
Now, we need to click on Serial Monitor to actually see the data being transferred.

Task 2: Display value for a sensor

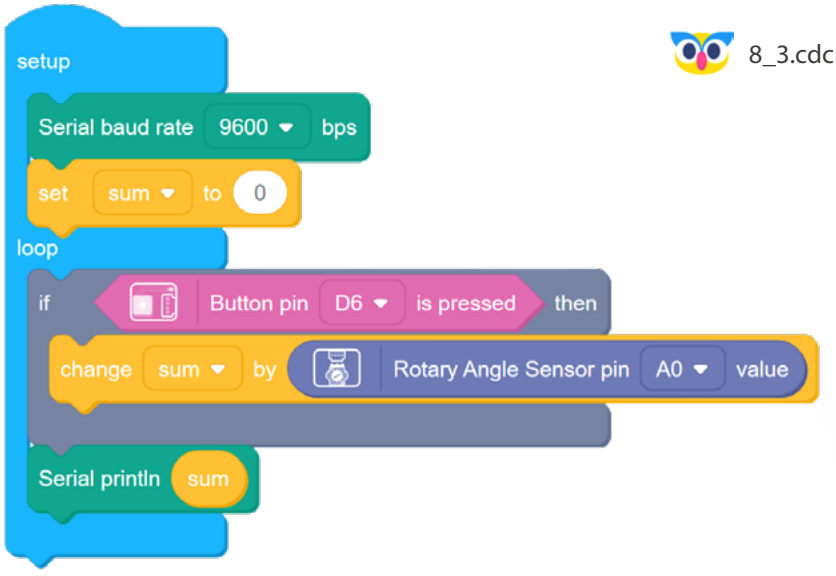
Now let's try displaying a value from button.



8_2.cdc



Let's try making more complicated code with a variable.

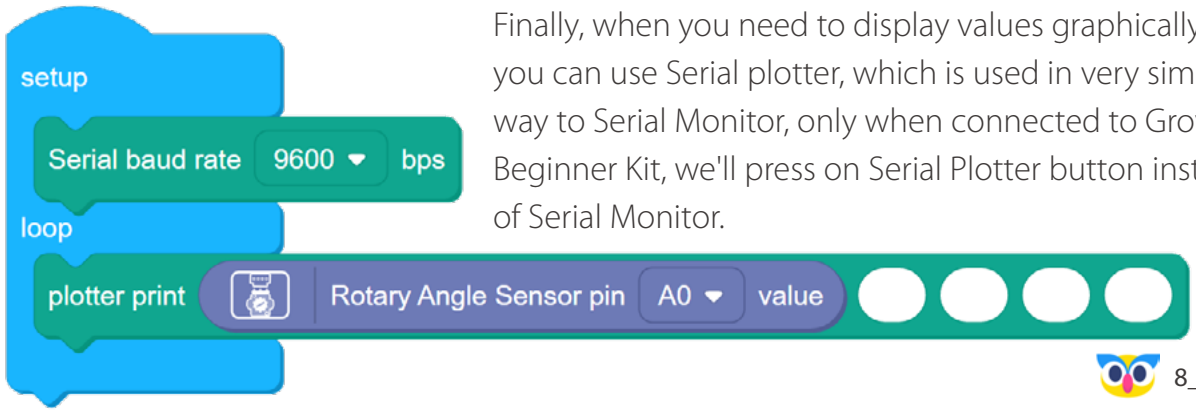


8_3.cdc

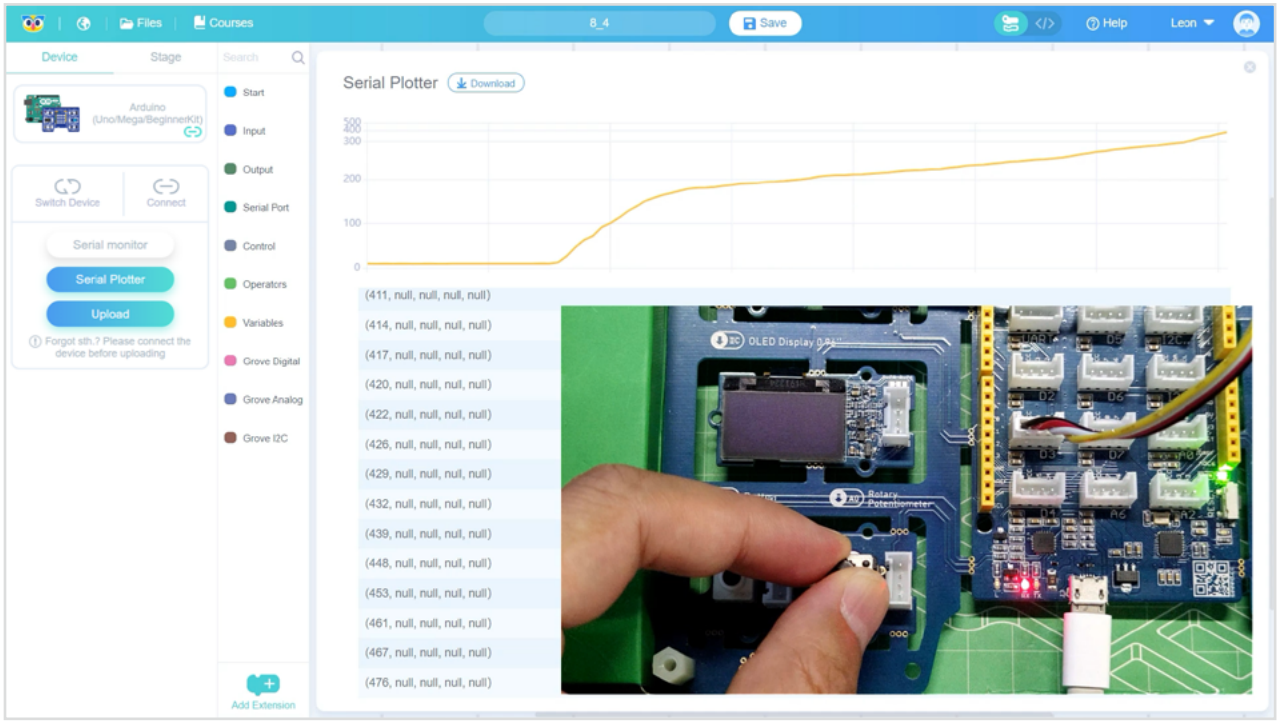
Task 3: Use Serial Plotter to visualize the data

In this program we are able to see the value of the variable which gets increased by value of Rotary Angle sensor, when button is pressed. It is useful since we can see the exact value of the variable. Actually, there seems to be a problem with the code - variable sum changes more than once during the button press…

Finally, when you need to display values graphically, you can use Serial plotter, which is used in very similar way to Serial Monitor, only when connected to Grove Beginner Kit, we'll press on Serial Plotter button instead of Serial Monitor.

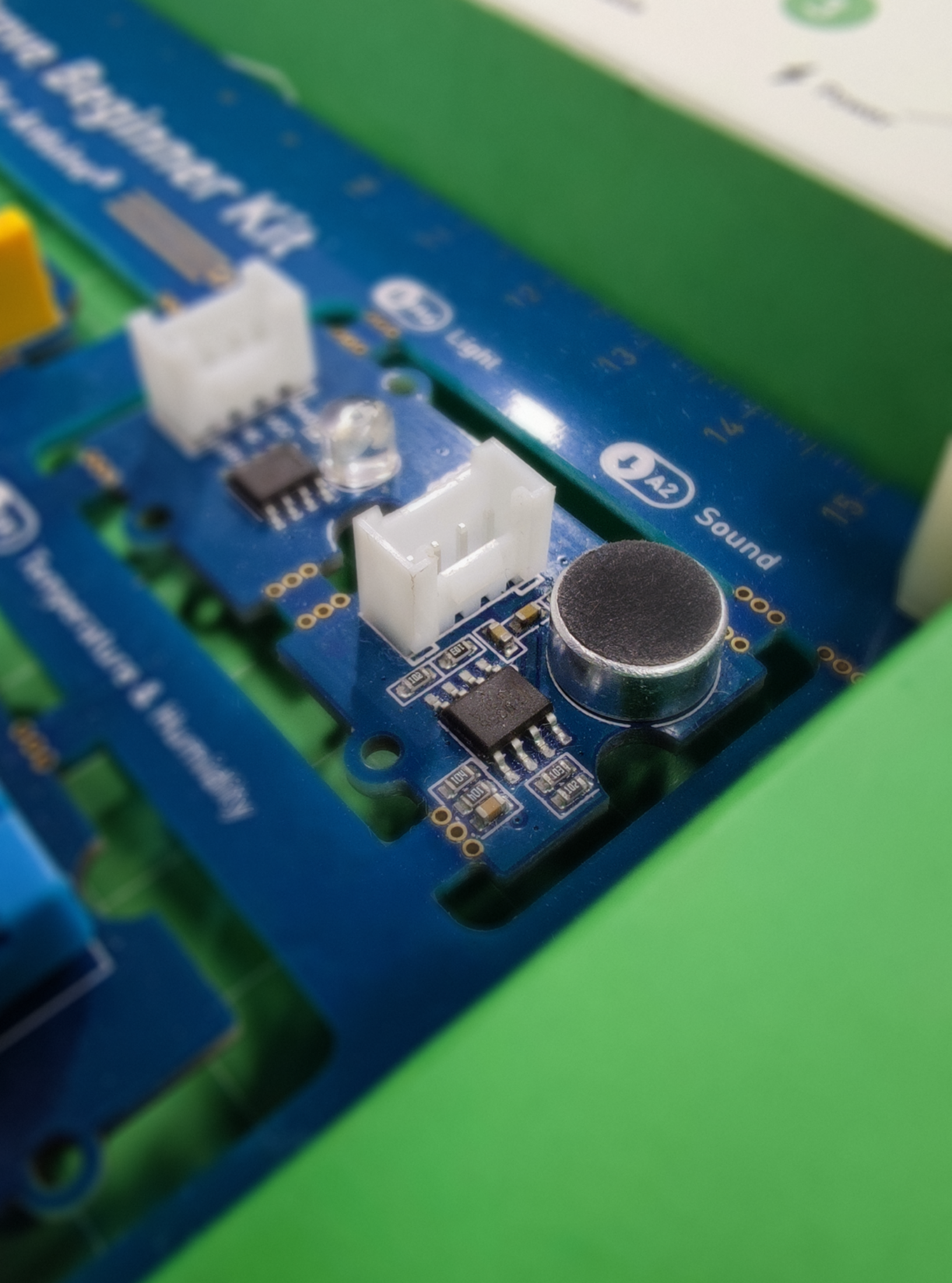


8_4.cdc



Outside the Box

- Use the delay block to fix Program 3 Lesson 8, so that every button press would add Rotary Angle sensor value to variable sum only once.
- Try changing baud rate in code block to other values - remember that you also need to change baud rate on Serial Monitor.
- Use while block and variable to print numbers from 1 to 10 to Serial Monitor every time button is pressed.


Light
A2
Sound

Lesson 9

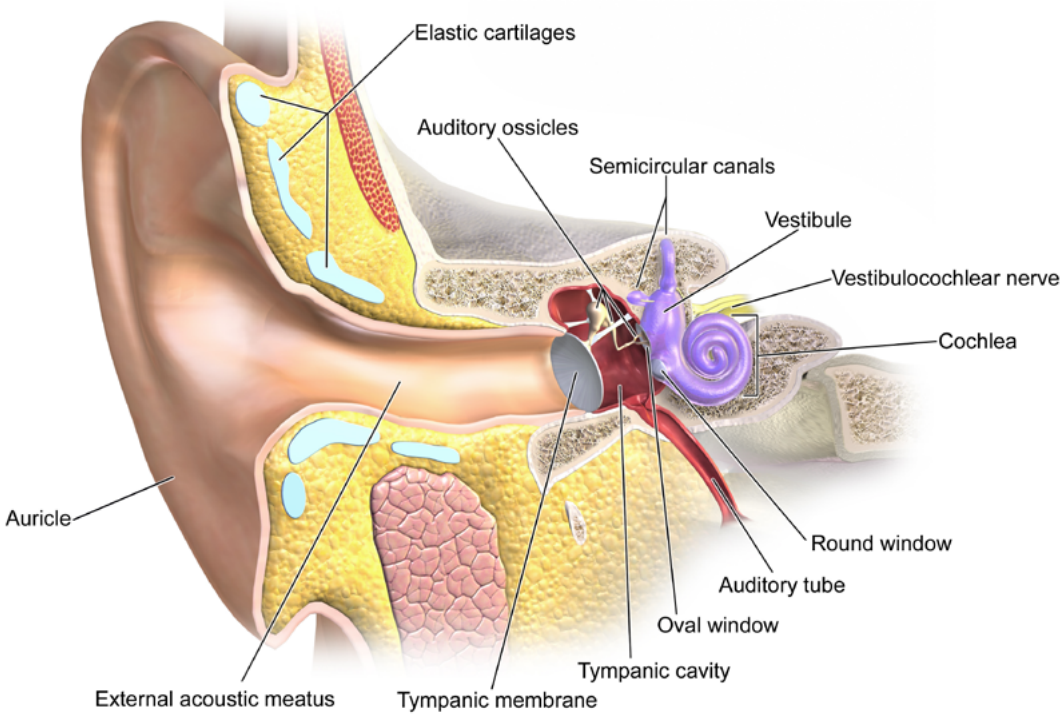
See the Sound

Imagine a world without sound. It would be a lot more boring and dangerous at the same time, since you would have less information about the environment. Human speech, animals screeches and whisper of tree leaves, warning signals and announcements - picked up by our ears all of these compliment and expand the view of the world be obtain from other sensory organs. Until before late 19th century there was no good way to record or transmit sound from one place to another, and even if you could, it was not very practical. We have come a long way since then, and now we have electronic systems which allow us to record, store and transmit sound from anywhere on earth. How do these work? Let's find out!

The Big Picture

What is sound and how do we hear?

Sound is the energy things produce when they vibrate (move back and forth quickly). These vibrations travel through a medium(usually air or water) and make medium molecules move. As the molecules move, they carries energy out from the sound source in all directions. When the vibrating molecules reach our ears, the eardrum vibrates, too. This is what happens from physics standpoint when sound is made and heard - one the vibration reaches our eardrum, it converts these vibrations into signals that are sent to the auditory cortex of our brains and then can be understood by us.

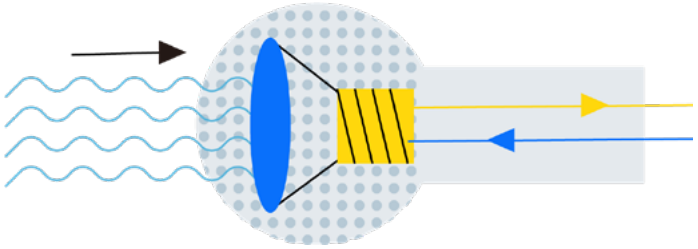


The Anatomy of the Ear

This is the biological aspect of sound. What we described above is how animals including humans perceive sounds - what about machines?

How does Sound sensor work?

Pretty much the same way. In a sound sensor, there is a diaphragm (similar to the one you can find in a speaker, but much smaller), that moves back and forth when the sound waves hit it.

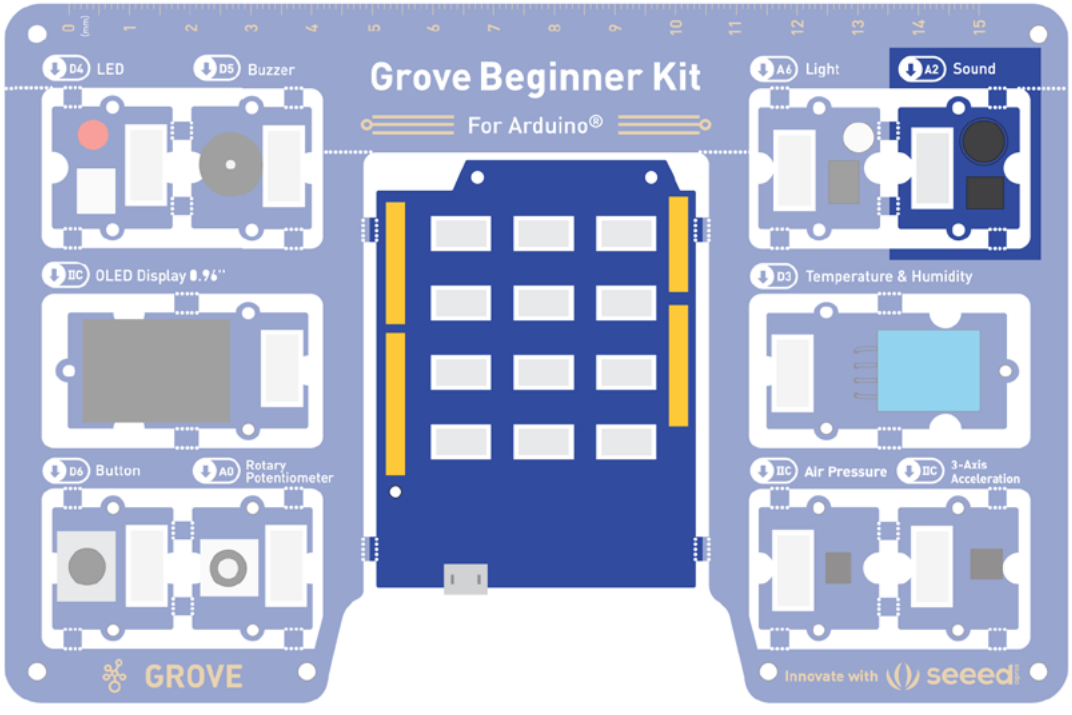


The coil attached to the diaphragm moves with it. As the coil moves back and forth through the magnetic field from a permanent magnet located inside the coil, an electric current flows through it. The current then can be amplified and directly forwarded to speakers(this is how megaphone/electric guitars work for example), or converted into a digital representation. The current from moving coil is an analog signal - louder sounds produce larger voltage and vice-verse. So, by using Analog to Digital Converter we can transform that analog signal from Sound sensor into digital, that the microcontroller in Grove Beginner Kit can "understand", pretty much the same way we did when using Potentiometer.

And by plotting the resulting numbers on a Codecraft interface we can see the sound wave.

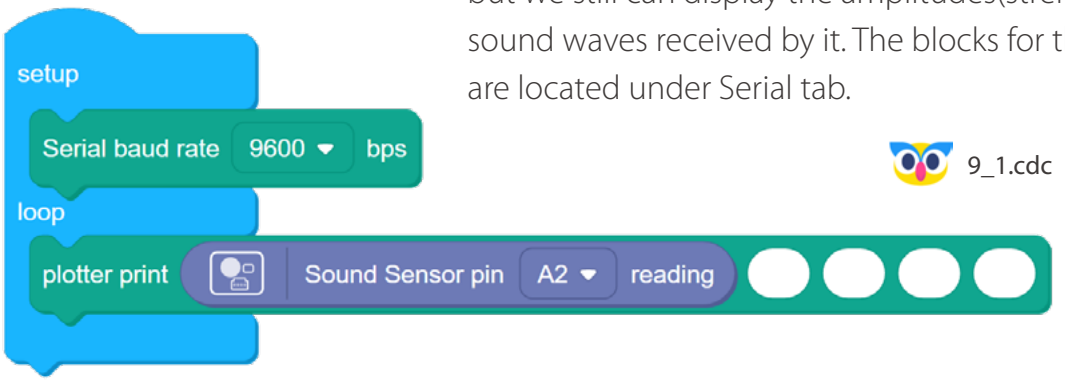
Sound sensor module in Grove Beginner Kit

In our Grove Beginner Kit, there is a sound sensor module.

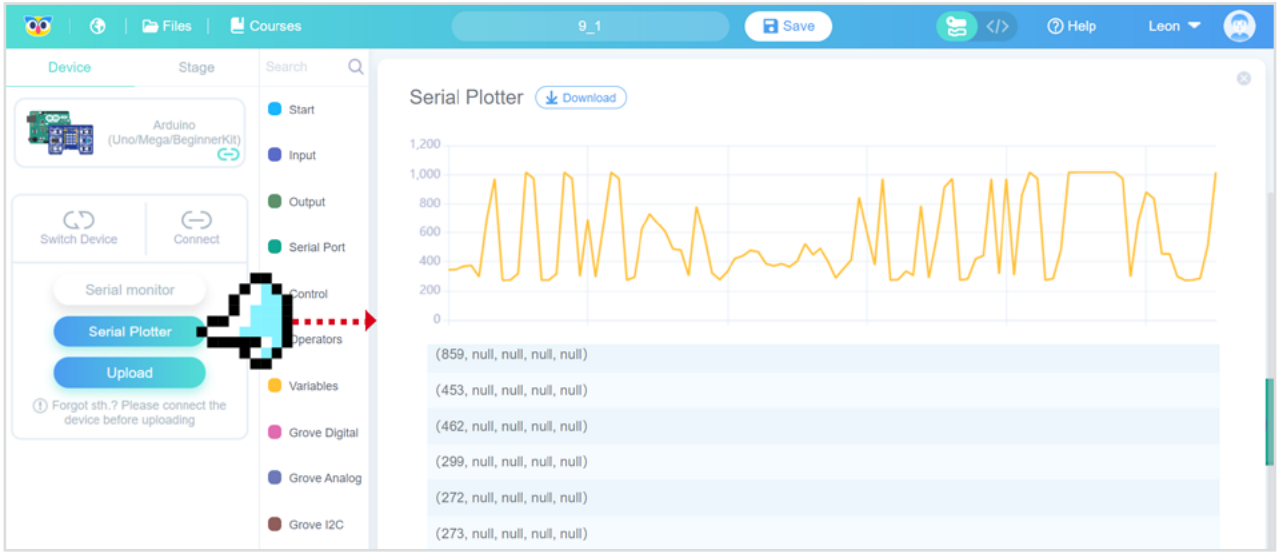


Task 1: Plot the Sound amplitudes with Serial Plotter

First we'll try displaying sound signal using Codecraft built-in Serial Plotter. The sound sensor we have in Grove Beginner Kit is relatively crude comparing to highly-sensitive microphones, but we still can display the amplitudes(strengths) of sound waves received by it. The blocks for this program are located under Serial tab.

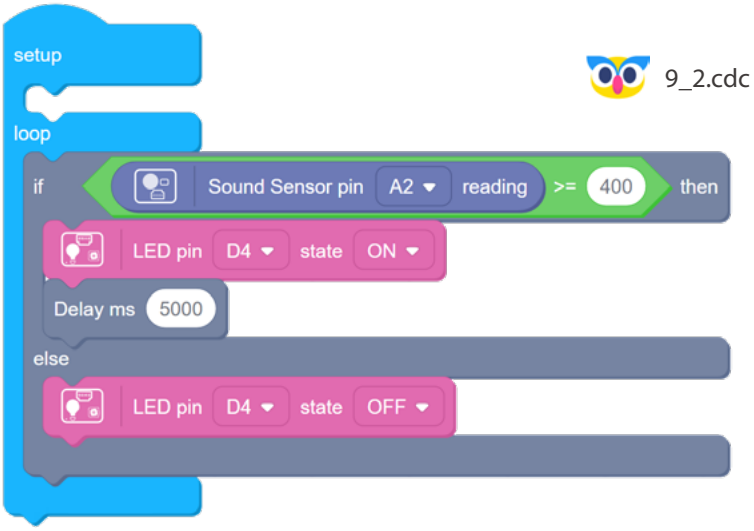


9_1.cdc



Task 2: Trigger an LED with loud sound

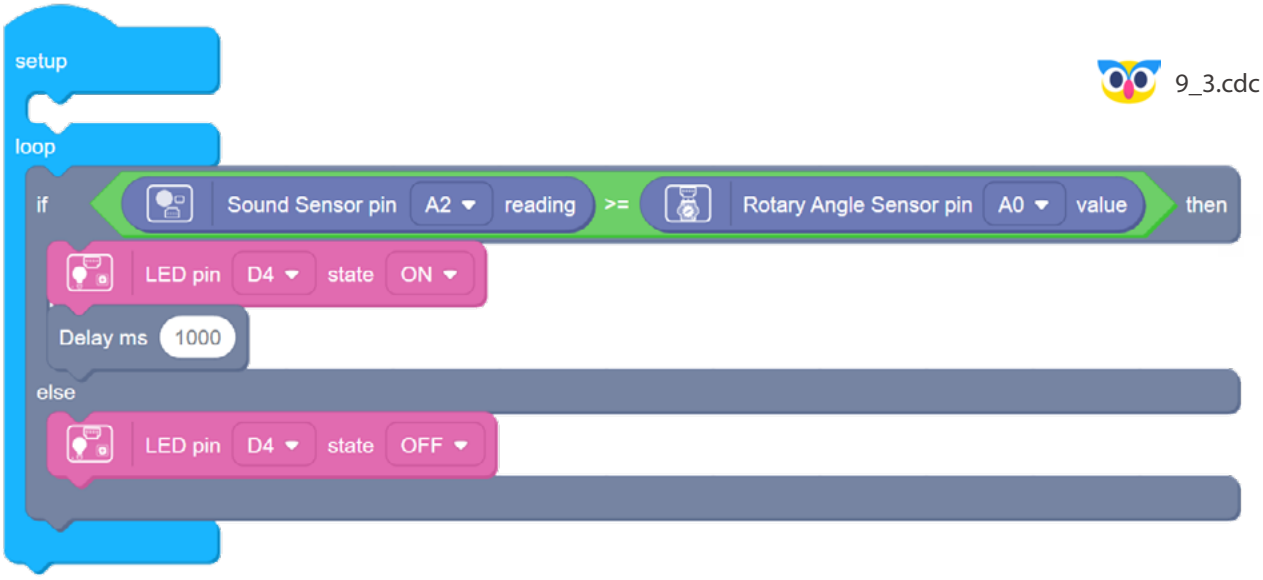
When we have a digital value of a sound signal, we can use it to trigger an LED or any other module at our disposal. For example, let's write a program that will switch an LED when loudness is higher than threshold.



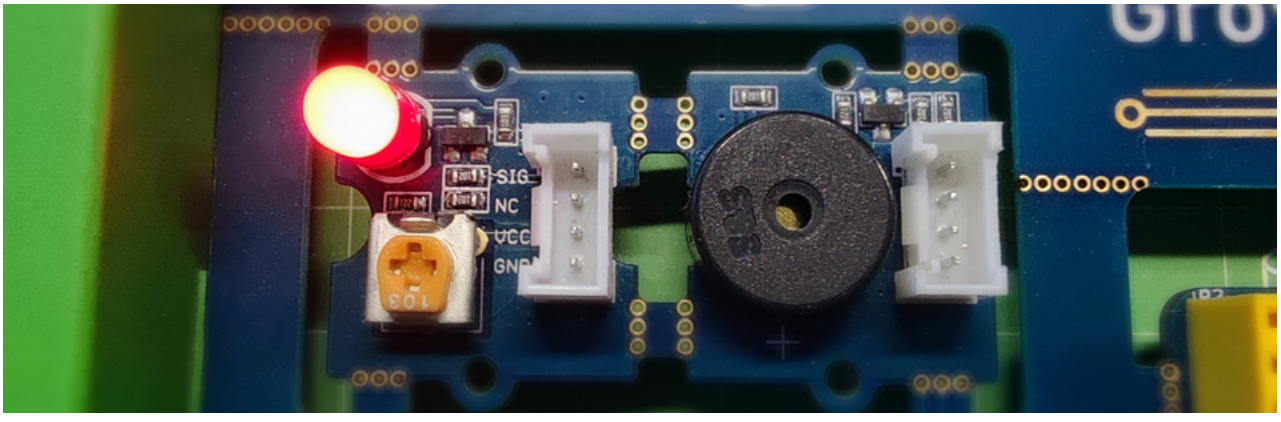
9_2.cdc

Task 3: Adjust the trigger threshold with Rotary angle sensor

It is quite inconvenience that to change that loudness threshold we need to change the value in code and re-upload the program. We could compare Sound sensor output to the output from another sensor! For example, the output of Rotary angle sensor(aka potentiometer) - and since both of these are analog signals,that have range 0-1023, we don't need to re-map them.
Now you can adjust the sensitivity of your sound detector with a simple turn of a handle!



9_3.cdc



Outside the Box

- Re-map the values from sound sensor to control the brightness of an LED module.
- Re-map the values from sound sensor from range 0-1023 to range 0-100 and display them on Serial Plotter.
- Use the button module to write program that can toggle sound detection with press of the button.

A2
Sound
A6
Light
D3
Temperature & Hu

Lesson 10

Speed of Light

Just as with sound, there was time in history(a long time) actually when the only way we could visually perceive the world around us was using our eyes. A few hundred years ago, if you wanted to capture a historical moment, you would need a person to paint it. That all changed with the invention of photography in 1822 and its subsequent development. From early on photos were created by chemical means - by exposing light-sensitive material such as photographic film to light going through a diaphragm. Later, in the 1981 first digital camera was unveiled - it used an electronic image sensor to record the image as a set of electronic data rather than as chemical changes on film. In our Grove Beginner Kit we have a very simple version of it - a Light sensor. Let's see how it works and how to use it.

The Big Picture

What is light?

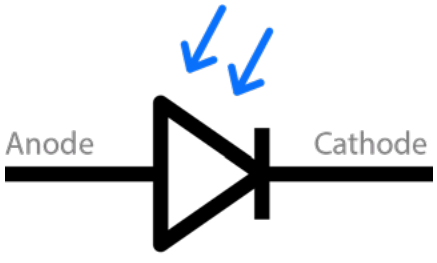
Comparing to sound, light differs a lot. It is much faster(in air 340 m/s vs 299 800 000 m/s), and unlike sound, which is vibration of other particles, light is the particle itself. The particle of light is called photon, it is s massless, has no electric charge, and also makes up for radio waves. There is a lot to process about light - for example the above mentioned fact that radio waves, visible light and X-rays are in fact the same phenomena!
They all are forms of electromagnetic radiation, as we mentioned in lesson one.

How does light sensor work?

A light sensor is a photoelectric device that converts light energy (photons, the light particles) detected to electrical energy (electrons). There are different types of light sensors, the one used in Grove Beginner Kit is a photodiode.
Photodiodes are mainly made from silicon and germanium materials and comprise of optical filters, built-in lenses and surface areas. Photodiodes work on the working principle called the inner photoelectric effect. Photodiode is a type of semi conducting device with PN junction. Between the p (positive) and n (negative) layers, an intrinsic layer is present. The photo diode accepts light energy as input to generate electric current. If you need to refresh your memory about P and N materials, go back to The Big Picture section of Lesson 7.

The brighter the light present, the stronger the electrical current will be. On circuit schematics, the symbol of the photodiode is similar to that of an LED but the arrows point inwards as opposed to outwards in the LED, which makes a lot of sense, since the working principle is similar.

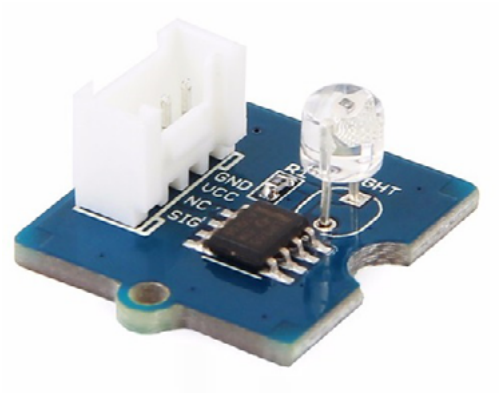
As you probably realized Light sensor is an analog module - it will output a voltage from 0V to 5V, depending on amount of light present. Let's see how to use Light Sensor in Codecraft!

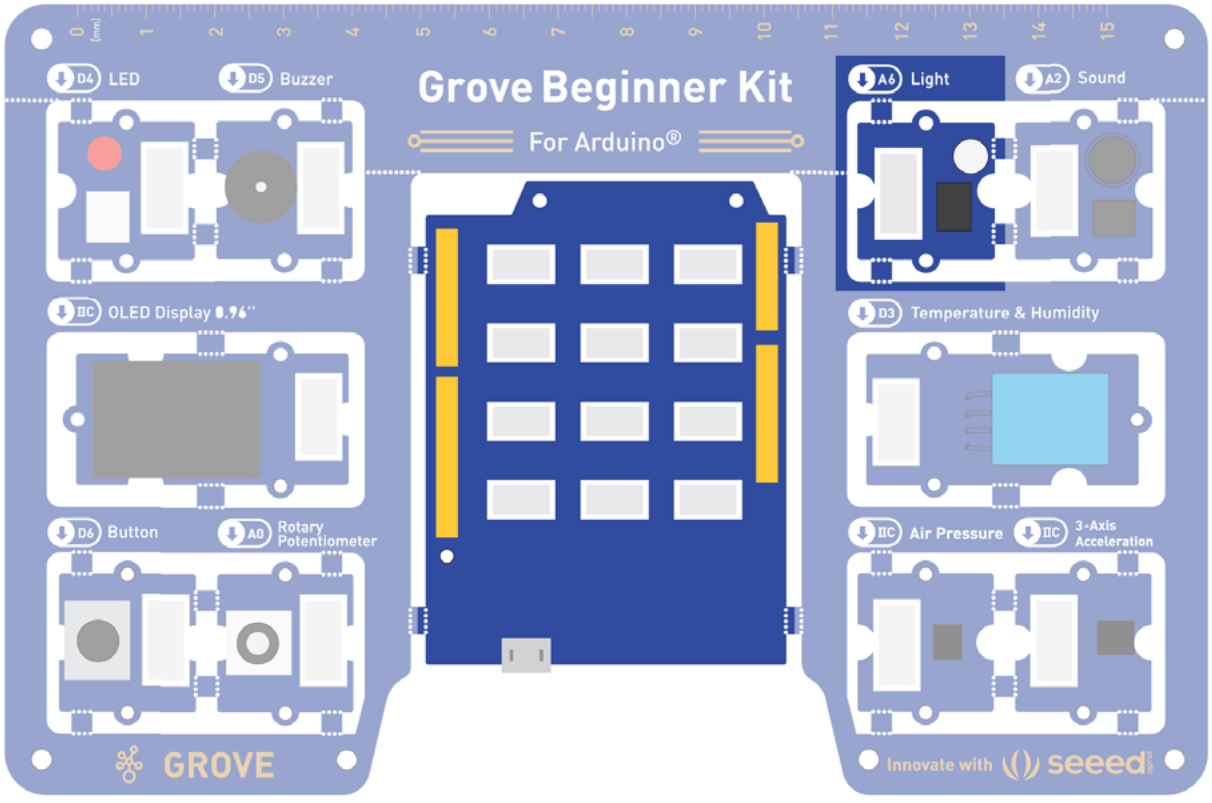


Photodiode symbol

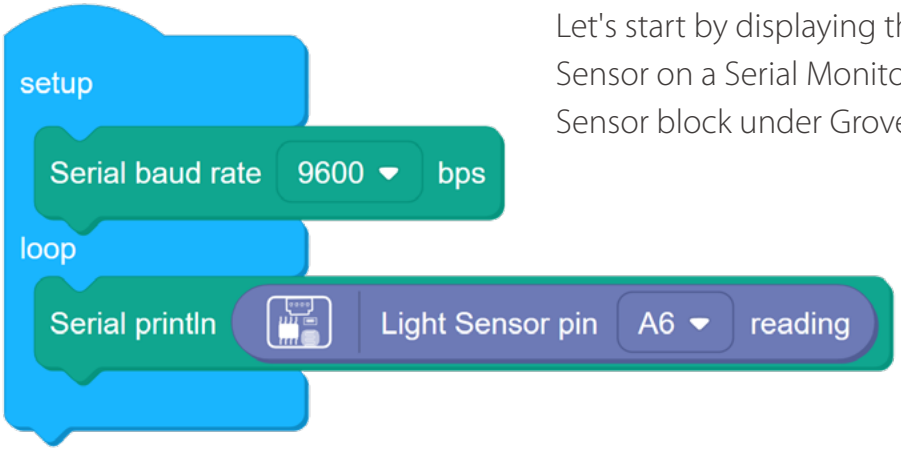
Light sensor module in Grove Beginner Kit

In our Grove Beginner Kit, there is a light sensor module.





Task 1: Visualize Light sensor output with Serial Plotter



Let's start by displaying the values from Light Sensor on a Serial Monitor. You can find Light Sensor block under Grove Analog tab.

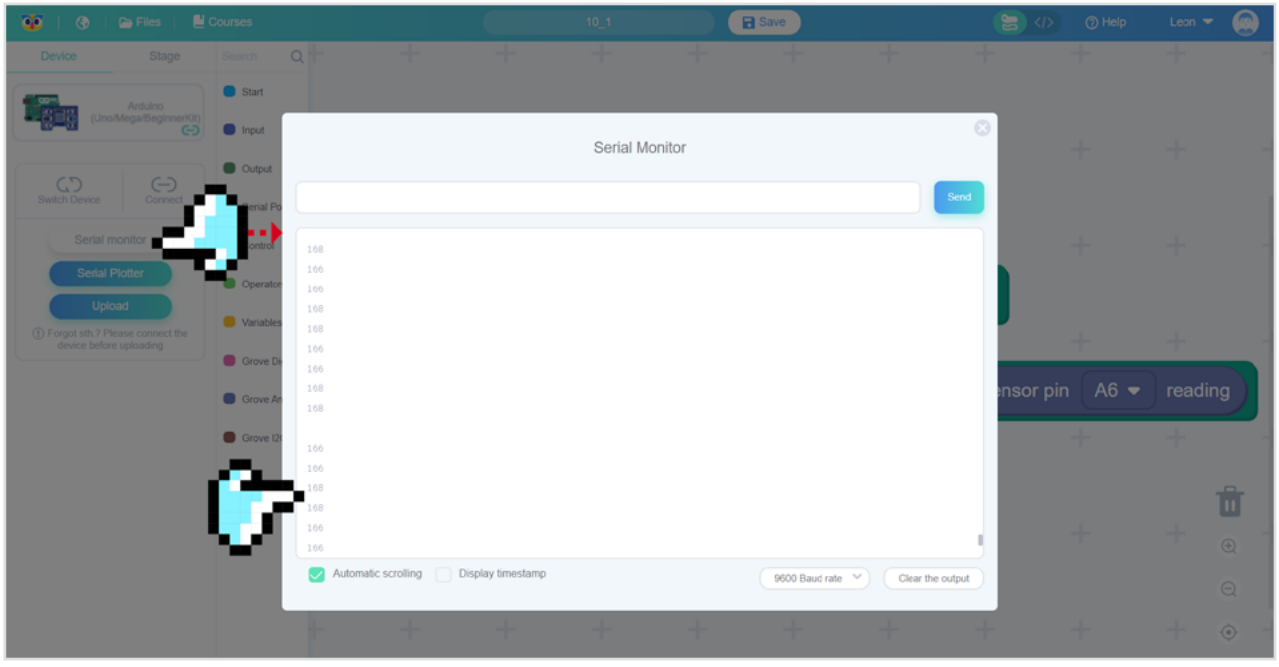
10_1.cdc

upload and connect the device

Upload the completed program to the starter kit, and connect the device in Codecraft again to confirm that the device is connected successfully.

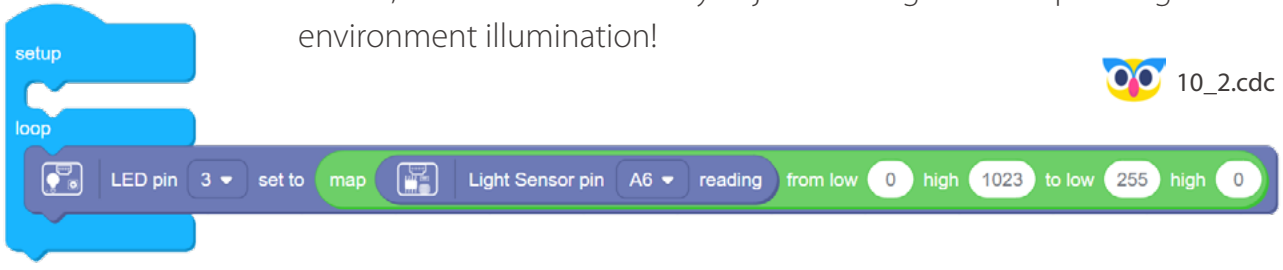
Open the Serial Monitor

Click the "Serial monitor" button to open the serial monitor window. You can see the numbers rolling. Try to block the light sensor with your hand. If you add light, you can see that the numbers are changing.



Task 2: Regulate brightness of an LED depending on environment illumination

Next, let's make an example program similar to what we have in our smartphones and tablets, that can automatically adjust the brightness depending on environment illumination!



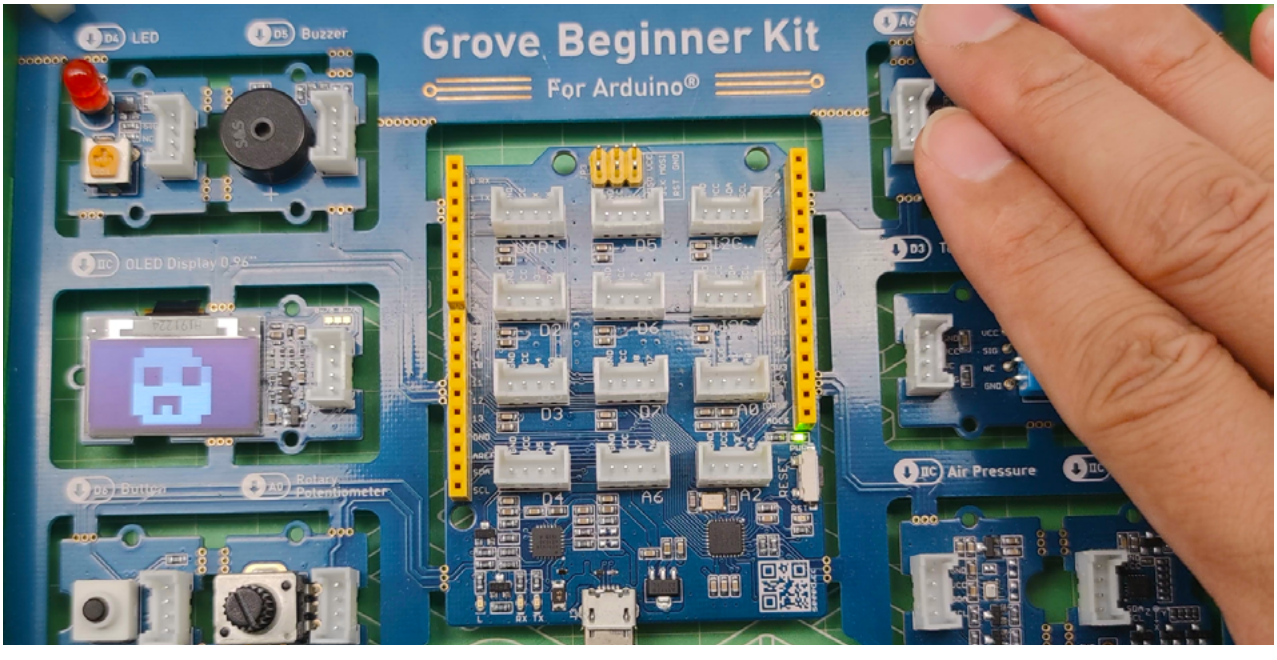
10_2.cdc

Here words low-high mean low and high ends of range and confusingly in this case low end of the range is higher than high end of the range for re-mapping, because we want the LED to get brighter if there is the environment around it is dark and vice-versa. Remember that for LED to change brightness you need to connect it with Grove cable to pin 3 - it is because pin 4, to which LED module is connected by default doesn't support PWM output.

Task 3: Make OLED screen display different graphics depending on environment illumination

Let's make an application using OLED screen to display the different pictures, depending on the brightness level. We will need to use logical AND operator for that - AND operator evaluates as True when both inputs to it evaluate as True.

10_3.cdc

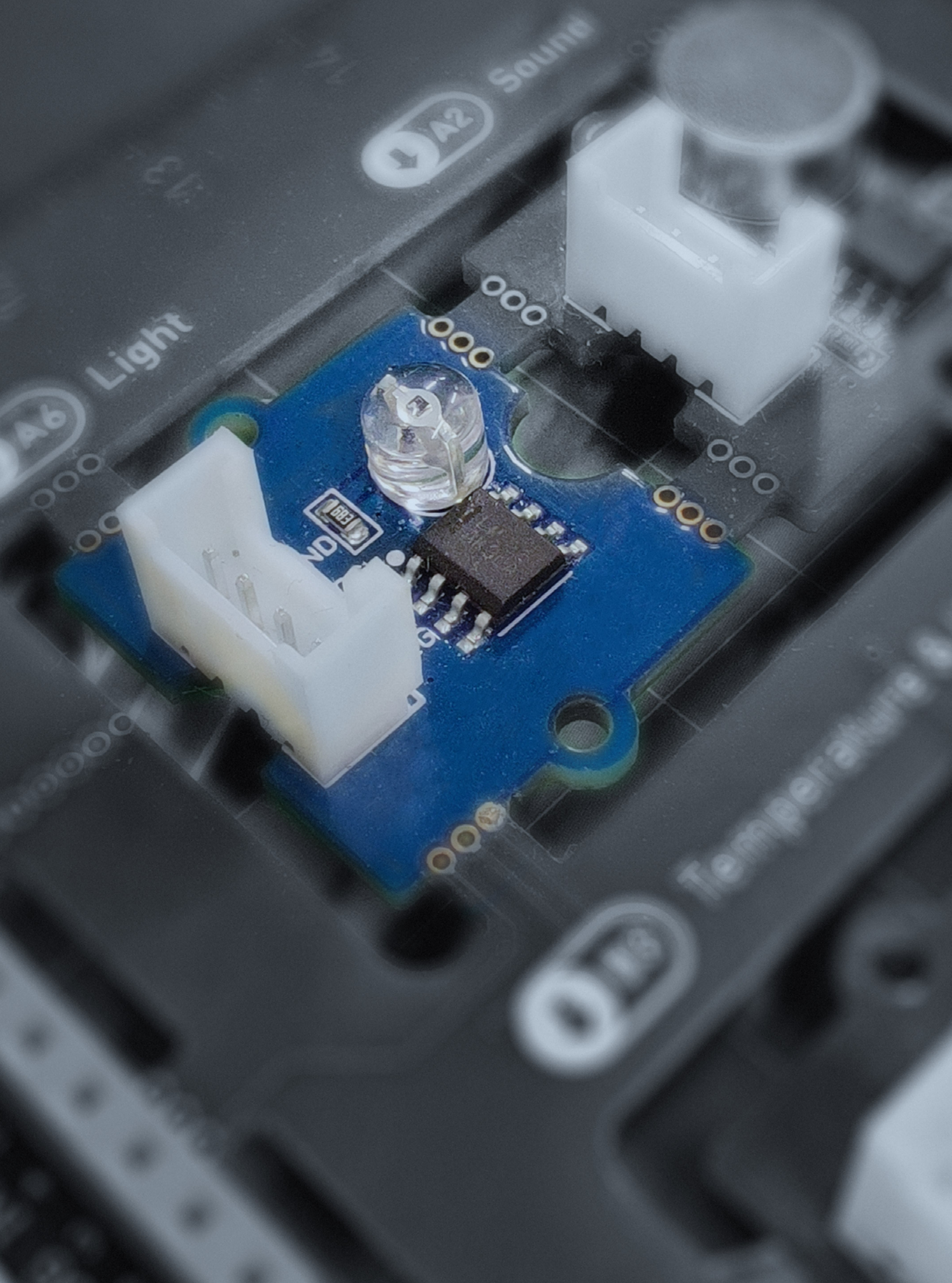


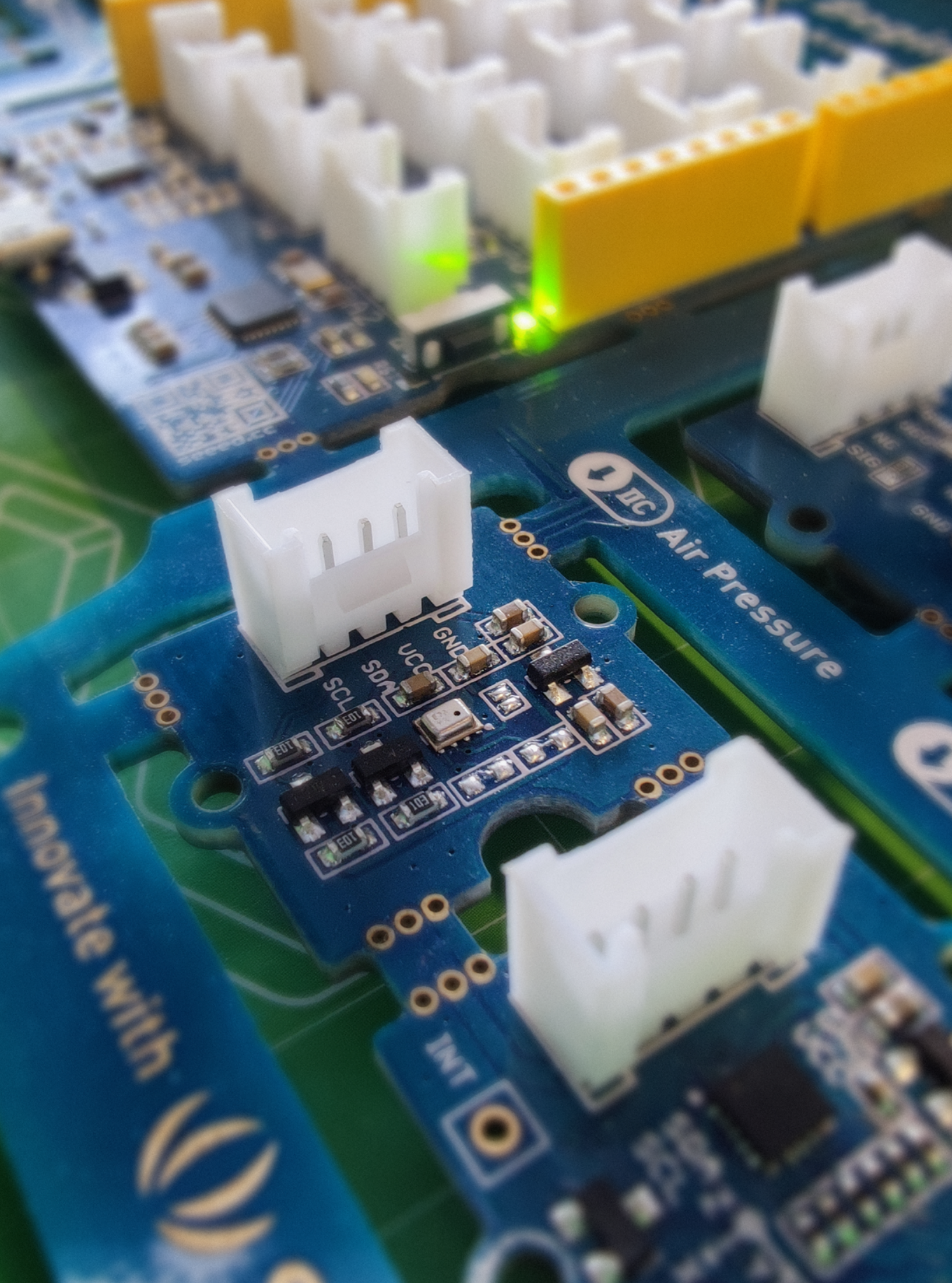
Logical operators are very important in any programming language and they help us take decisions based on certain conditions. Suppose we want to combine the result of two conditions, then logical AND and OR logical operators help us in producing the final result.

Logical AND operator. If both the operands are non-zero, then condition becomes true.	(A && B) is false. 
Logical OR Operator. If any of the two operands is non-zero, then condition becomes true.	(A || B) is true. 
Logical NOT Operator. Use to reverses the logical state of its operand. If a condition is true then Logical NOT operator will make false.	!(A && B) is true. 

⭐ Outside the Box

- Write a code using AND operator to control the buzzer: emit different notes depending on the amount of light detected by Light Sensor.
- Plot Light Sensor reading with Serial Plotter.
- Make a fun reaction testing game - when button is pressed, the board will start counting time (using System Running time block in Input), finish counting time when Light Sensor reading is lower than 200 (meaning the sensor is covered with hand) and then display that time as a number on the screen. Then you can take turns measuring who's reacting is faster.

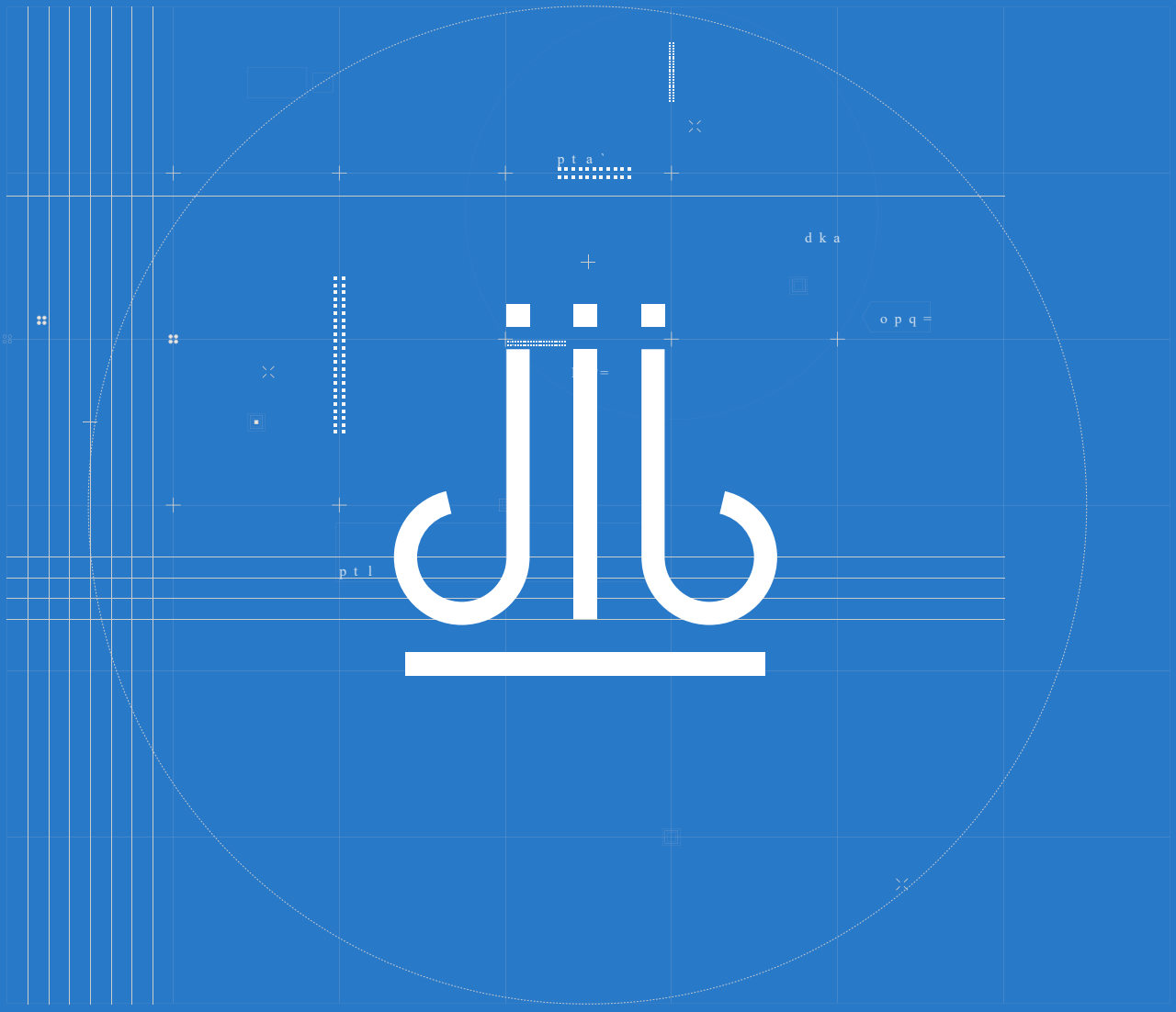

A2
Sound
A6
Light
Temperature


IIC
Air Pressure
GND
VCC
SDA
SCL
INT
Innovate with

Lesson 11

Gaining Altitude

Can you feel the weight on your head and shoulders as you sit in the class? The weight of what, you might ask? There nothing that can be seen there! Exactly. Despite you cannot see it, you know that there is air all round you - under normal circumstances of course. And air is a physical substance, that has mass and therefore can exert pressure on the objects. You don't feel it because the air was surrounding us for a long time - so the bodies of animals (and by extension humans) are adapted to it and don't even notice it's presence. How can we measure the air pressure and what influence does it have on our lives? Let's find out!

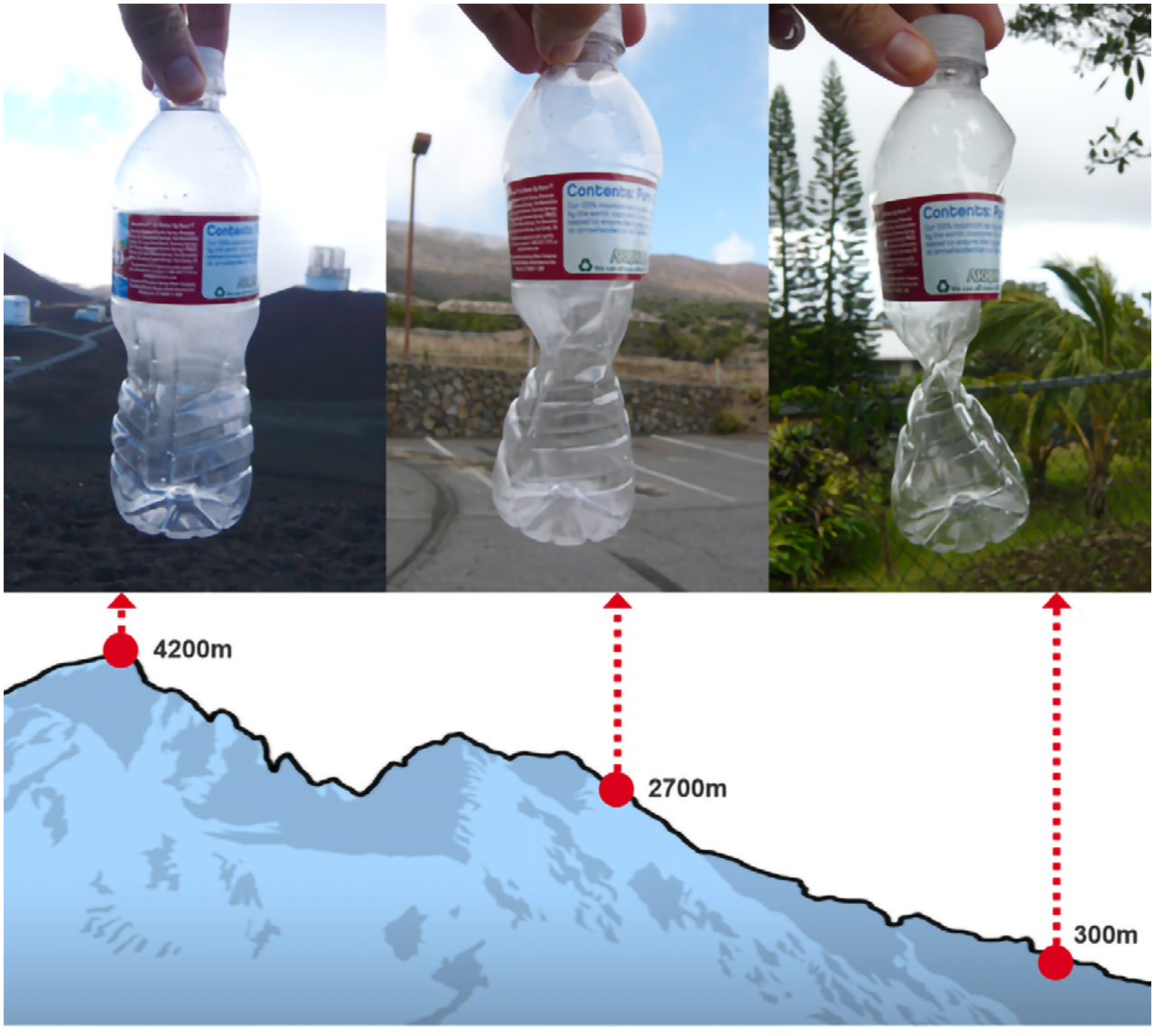


The Big Picture

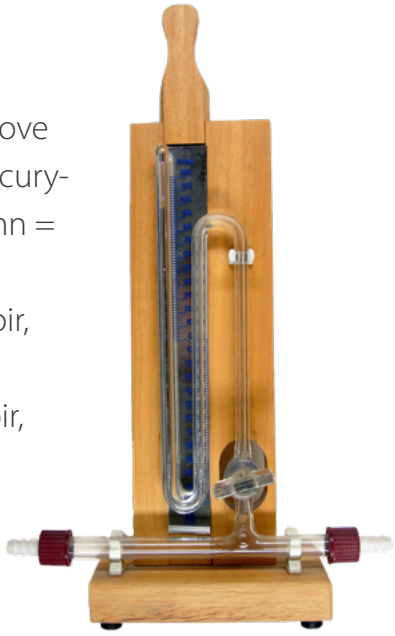
What is atmospheric pressure?

Atmospheric pressure is a force in an area pushed against a surface by the weight of the atmosphere of Earth, a layer of air. The air is not distributed evenly around the globe. It moves, and at different times, the layer of air is thicker in some places than in others. Where the layer of air is thicker, there is more air. Since there is more air, there is a higher pressure in that spot. Where the layer of air is thinner, there is a lower atmospheric pressure. At higher altitude, the atmospheric density and pressure are lower. This is because high places do not have as much air above them, pushing down.

This plastic bottle was sealed at approximately 4200 meters altitude, and was crushed by the increase in atmospheric pressure (at 2700 meters and 300 meters) as it was brought down towards sea level.

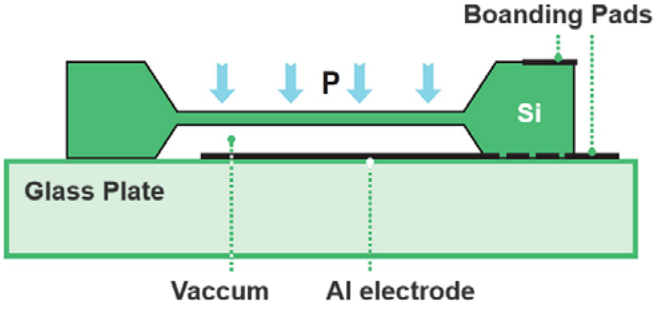


How can we measure air pressure?



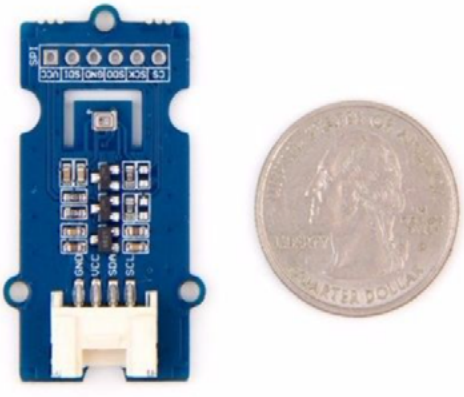
Except for sealed plastic bottles, barometers can be used to measure atmospheric pressure. One of the oldest types of barometers is the above mercury barometer, consisting of a vertical glass tube sitting on a mercury-filled basin at the bottom. The higher the height of the mercury column = the higher the atmospheric pressure is present.
High atmospheric pressure results in more force placed on the reservoir, which forces mercury higher in the column.
Low atmospheric pressure results in lesser force placed on the reservoir, which allows mercury to drop lower down the column.

With the advancement in technology, Modern-day barometric pressure sensors no longer require liquid for sensing, resulting in better accuracy. Such barometric pressure sensor types include aneroid barometers and MEMS barometer. Modern-day barometer uses MEMS(Micro-Electro-Mechanical Systems) technology, it contains a diaphragm that's formed through one capacitive plate that's in contact with the atmosphere.

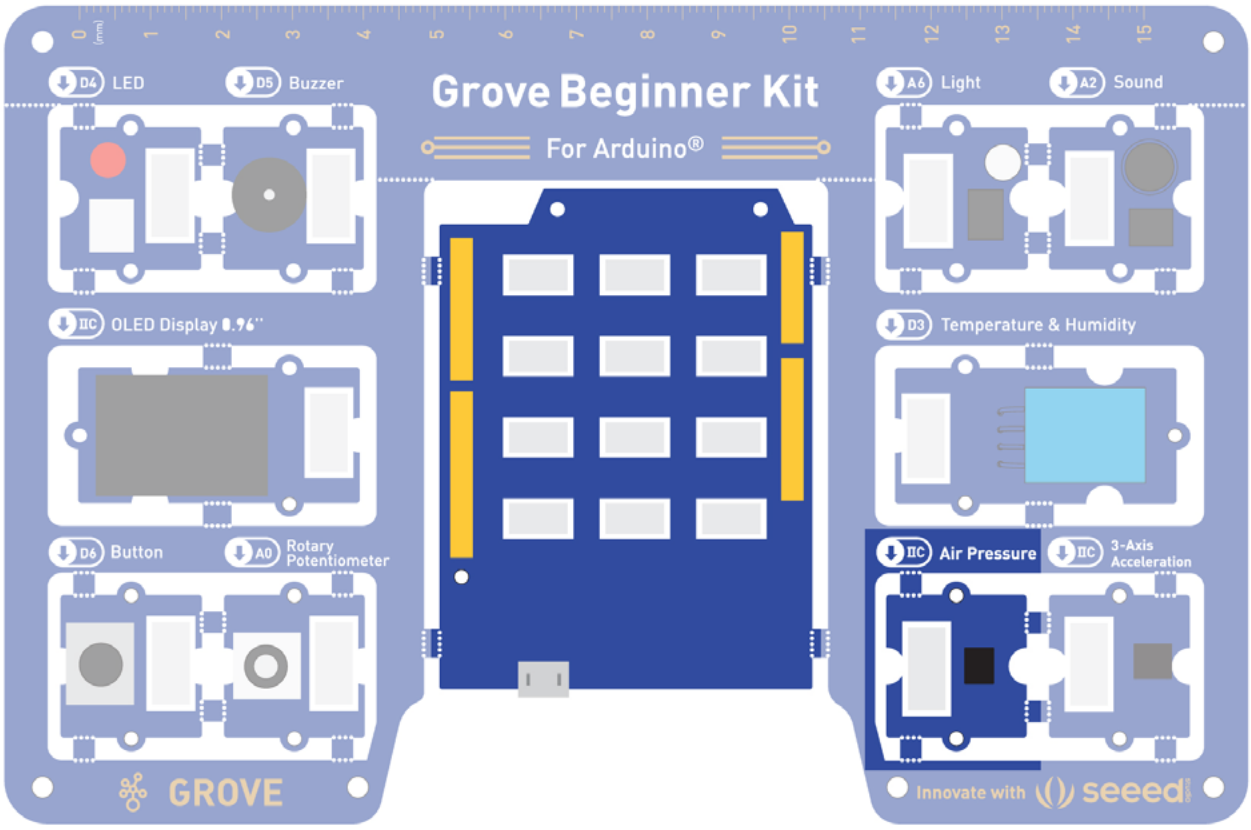


Atmospheric pressure is detected through how much the diaphragm is deformed due to resulting pressure. The higher the pressure, the more the diaphragm moves, which result in a higher barometer reading.
A falling barometric reading means there's a decrease in air pressure, which often indicates there's a higher chance of rain coming. An increase in barometric reading means there's an increase in air pressure, clearing the skies and indicating cool, dry air is coming.

Air Pressure sensor module in Grove Beginner Kit

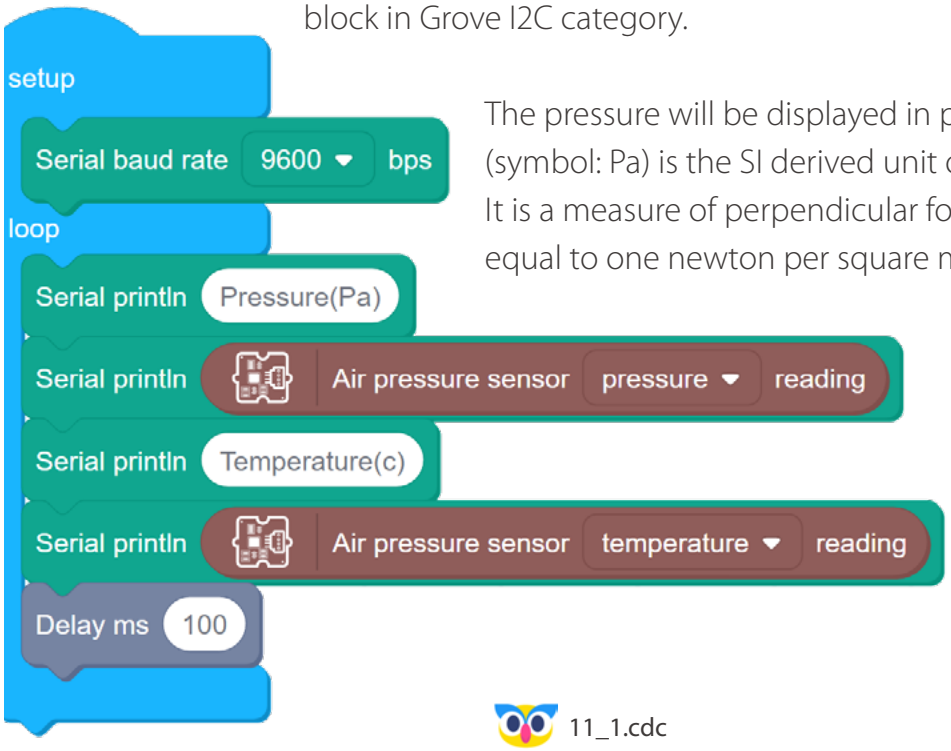


In our Grove Beginner Kit, there is a air pressure sensor module. The module used in Grove Beginner Kit for measuring the pressure is called BMP280 and here is how it looks.



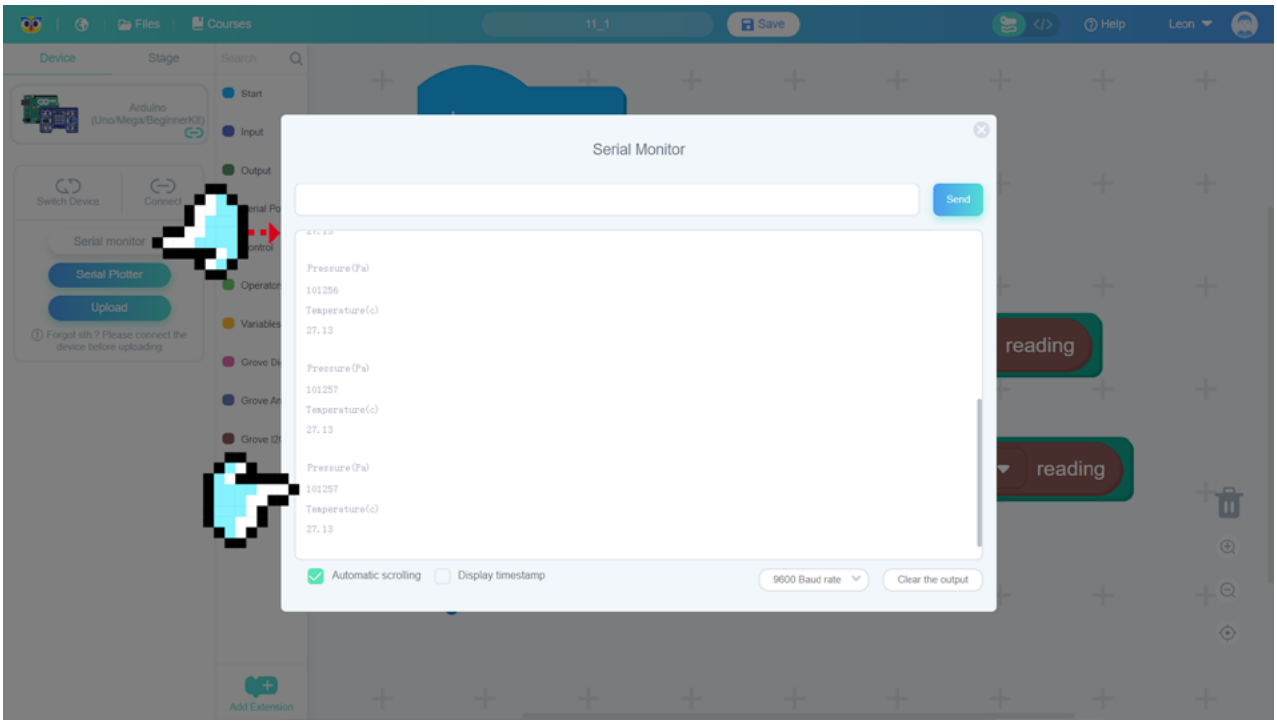
Task 1: Display the air pressure sensor reading on the terminal

First let's start by displaying the readings from barometer sensor on a serial terminal - the BMP280 sensor can measure air pressure and temperature. You can find Air pressure sensor block in Grove I2C category.



11_1.cdc

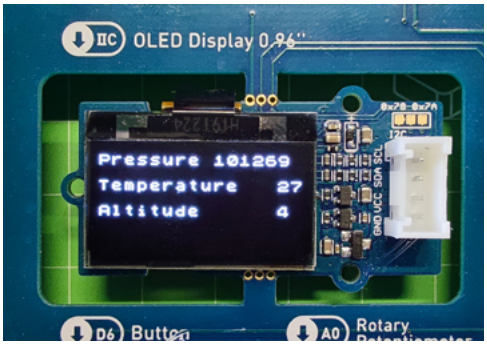
The pressure will be displayed in pascal. The pascal (symbol: Pa) is the SI derived unit of pressure or stress. It is a measure of perpendicular force per unit area i.e. equal to one newton per square meter. The standard atmosphere (symbol: atm) is a unit of pressure defined as 101325 Pa. It is sometimes used as a reference or standard pressure, because it is approximately equal to Earth's atmospheric pressure at sea level.

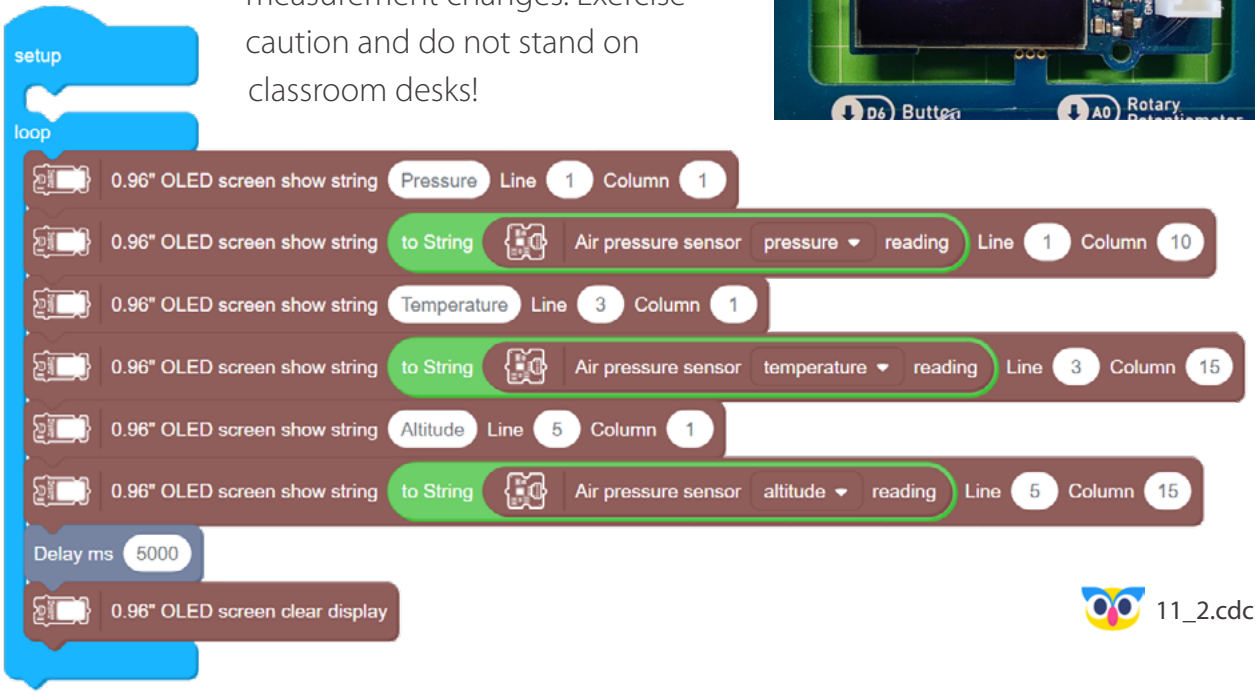


Task 2: Display the data from sensor on OLED screen

Next we can automatically calculate the altitude - elevation above sea level using the data about air pressure and display it together with other parameters on OLED screen!

The Barometer sensor in Grove Beginner Kit is sensitive enough to detect the difference if you raise the board above your computer, up to the full length of the USB cable - try doing it yourself and see how the altitude measurement changes. Exercise caution and do not stand on classroom desks!



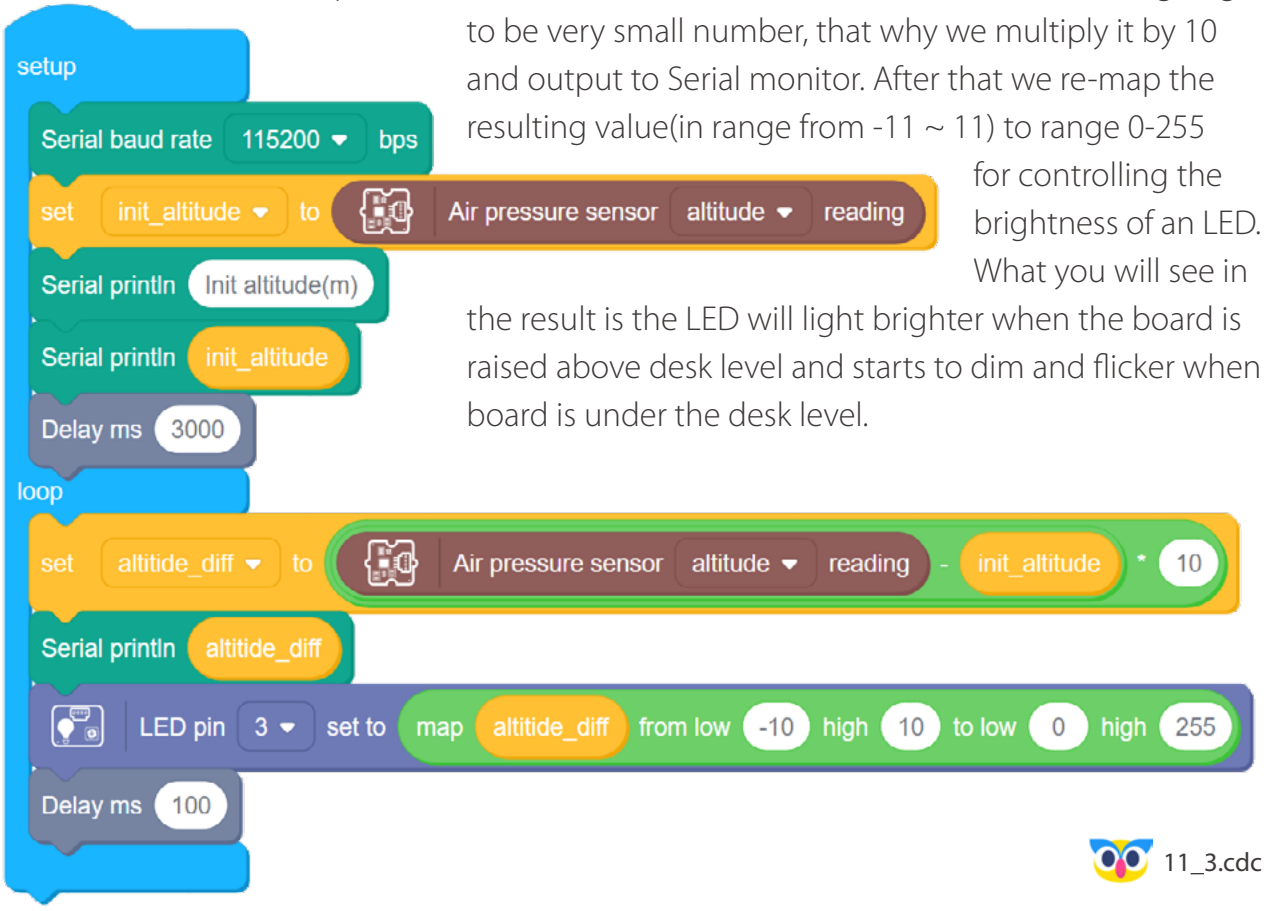


11_2.cdc

Task 3: Use air pressure sensor to change brightness of an LED

Finally we can try making a program that will make LED shine brighter the higher we raise our board!

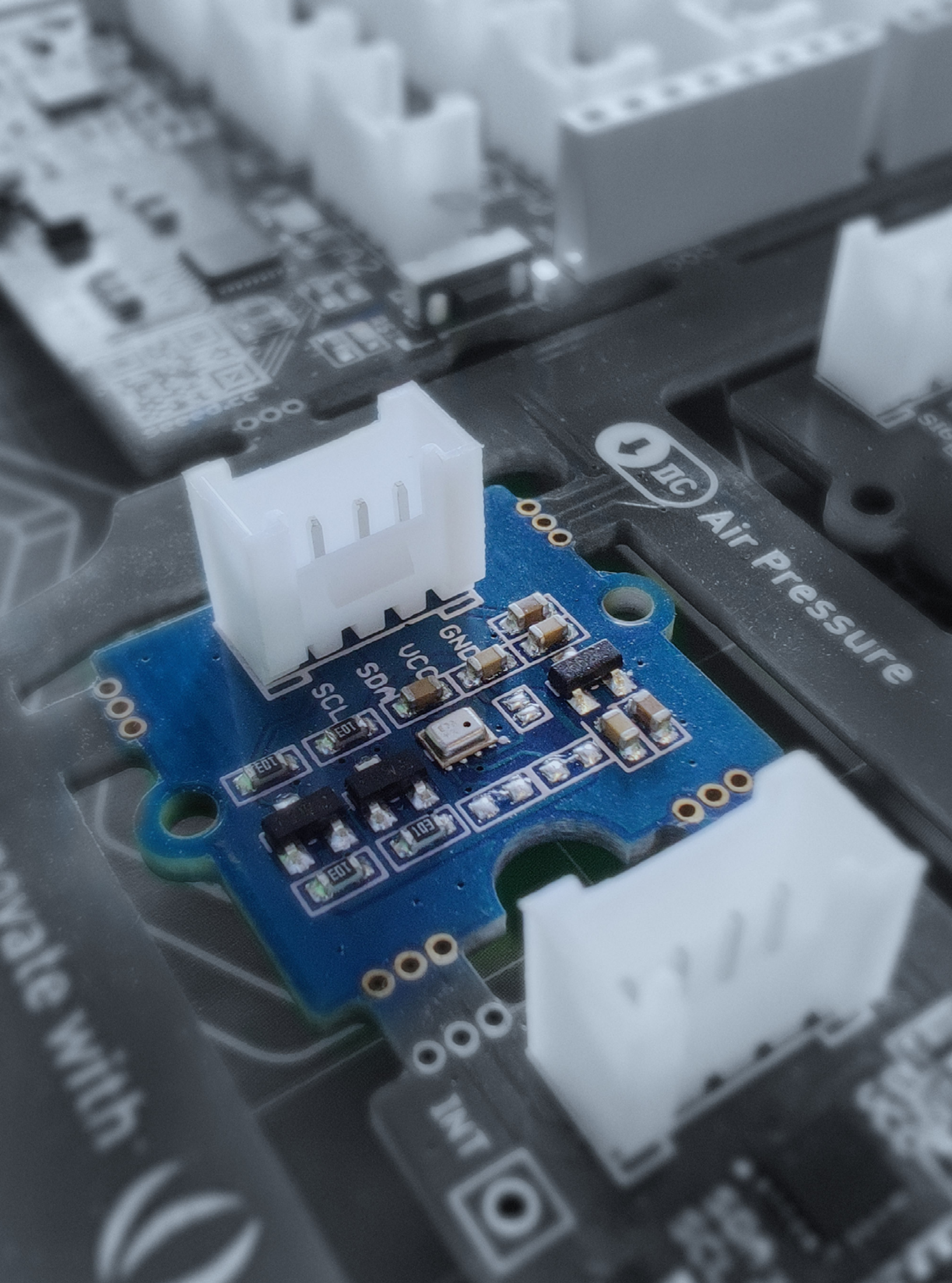
We first record the initial altitude (the board on your desk) to a variable called init_altitude. Then inside the main loop we subtract that initial altitude from current altitude - it is going to be very small number, that why we multiply it by 10 and output to Serial monitor. After that we re-map the resulting value(in range from -11 ~ 11) to range 0-255 for controlling the brightness of an LED. What you will see in the result is the LED will light brighter when the board is raised above desk level and starts to dim and flicker when board is under the desk level.

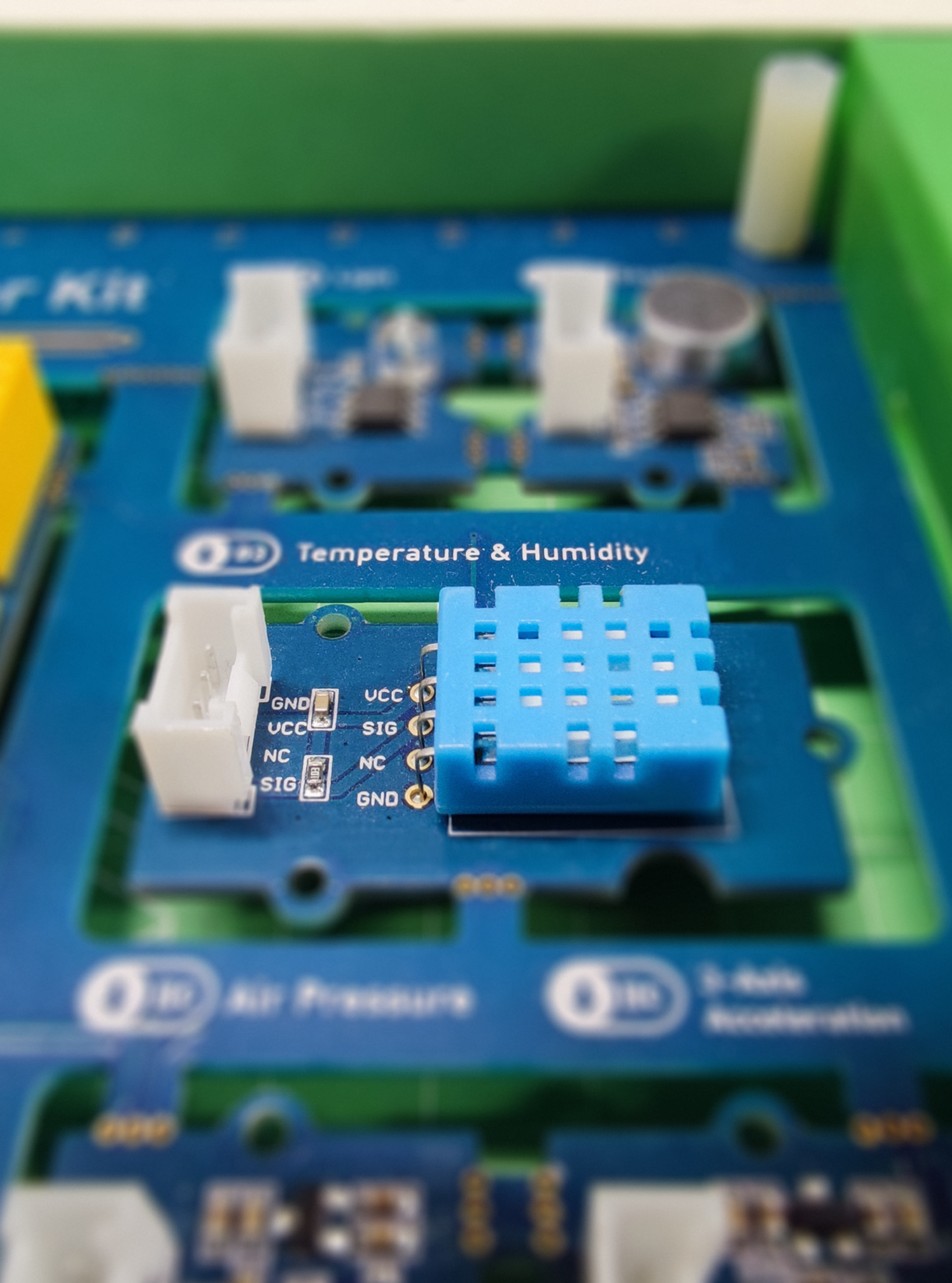


11_3.cdc

Outside the Box

- Re-write the first program of the lesson to display pressure in kilopascals(1000 pascals) and hectopascals(100 pascals).
- Display temperature and pressure on OLED display.
- Use the button to switch between displaying temperature/pressure/altitude on OLED display.


I²C
Air Pressure
GND
VCC
SDA
SCL
EOT
EOT
EOT
103


Temperature & Humidity
GND
VCC
NC
SIG
VCC
SIG
NC
GND

Lesson 12

Rain and Shine

Weather and climate affect almost every facet of our lives. The agriculture and economies depend on good weather for optimal food production. The shipment of good on planes/ ships can be affected by weather. And of course, on an individual level, your weekend plans can be ruined by a heavy rain coming out of nowhere after a sunny Friday. No wonder humans wanted to predict the weather from a long time ago. Originally people were trying to predict the weather by observing the changes in plants, behavior of animals and other indirect sings of weather change. With the advancement of science we now have better tools that can help us to monitor and possibly forecast the weather. Let's have a closer look at one of these tools, Temperature and Humidity Sensor.

The Big Picture

What is a temperature and humidity?

In the previous lesson we already learned how to measure the air pressure - and how it can it help us to predict the weather. Now what is a temperature and humidity?

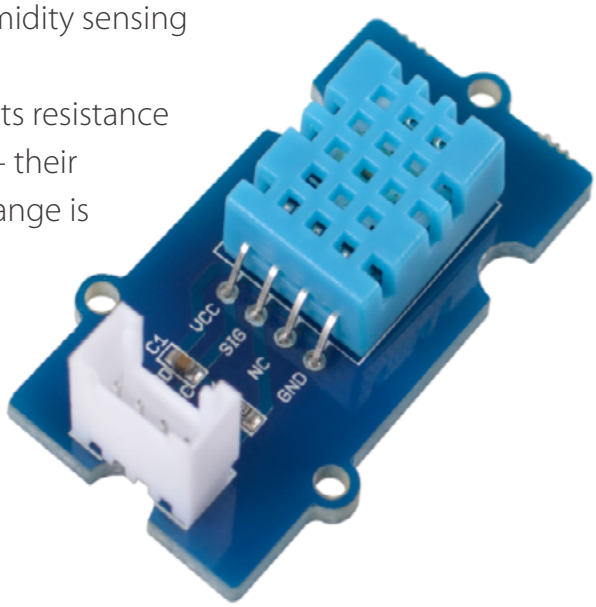
We normally understand temperature as how cold or hot something is - we can feel the temperature with our skin or use thermometer to measure it. Scientifically, temperature is a physical quantity which describes how quickly molecules are moving inside a material. If something has a high temperature, it means the average speed of its molecules is fast. As for humidity, it is the amount of water vapor in the air. If there is a lot of water vapor in the air, the humidity will be high. The higher the humidity, the wetter it feels outside. On the weather reports, humidity is usually explained as relative humidity. Relative humidity is the amount of water vapor actually in the air, expressed as a percentage of the maximum amount of water vapor the air can hold at the same temperature.

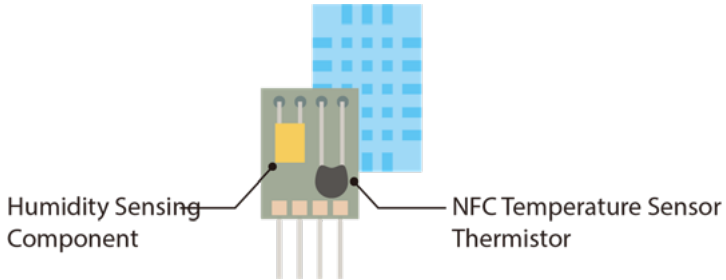
How can we measure temperature and humidity with digital sensor?

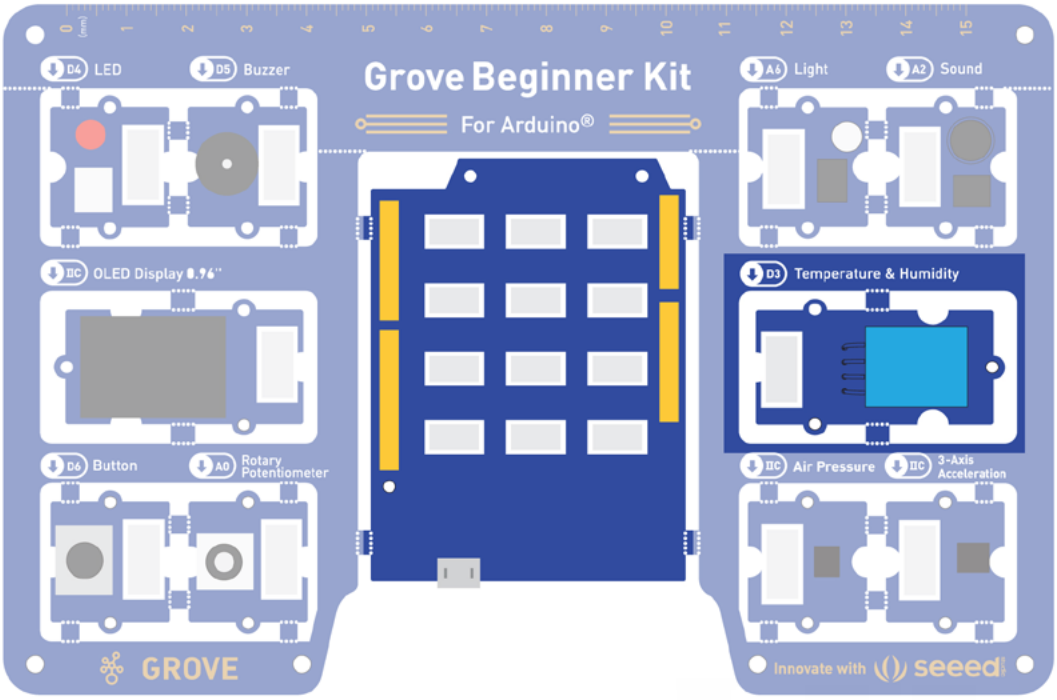
The sensor used for measuring temperature and humidity in Grove Beginner Kit is called DHT11.

Inside our sensor there are actually two components - humidity sensing component and a thermistor.

A thermistor is a thermal resistor – a resistor that changes its resistance with temperature. Technically, all resistors are thermistors – their resistance changes slightly with temperature – but the change is usually very very small and difficult to measure.





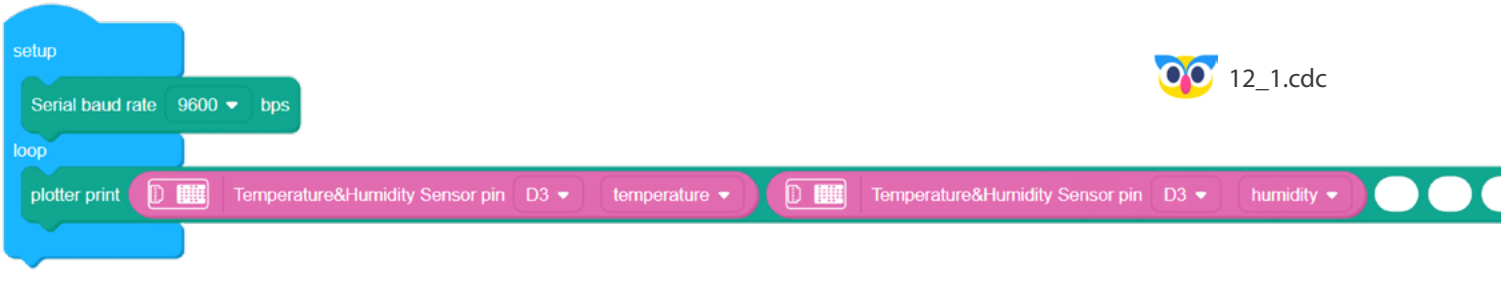


Thermistors are made so that the resistance changes drastically with temperature so that it can be 100 ohms or more of change per degree! As we remember from earlier lessons, changes to resistance in the circuit affect the voltage - and by measuring voltage we can infer the temperature - in this case the air temperature.

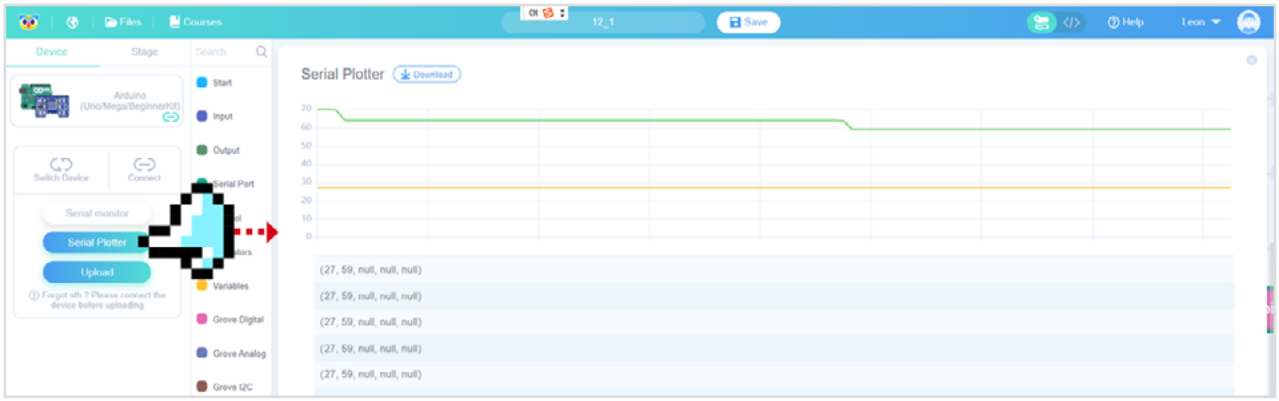
For measuring humidity the humidity sensing component is used, which has two electrodes with moisture holding substrate between them. So as the humidity changes, the conductivity of the substrate changes or the resistance between these electrodes changes. This change in resistance is measured and processed by the IC which makes it ready to be read by a microcontroller.

Task 1: Plot output from sensor with Serial plotter

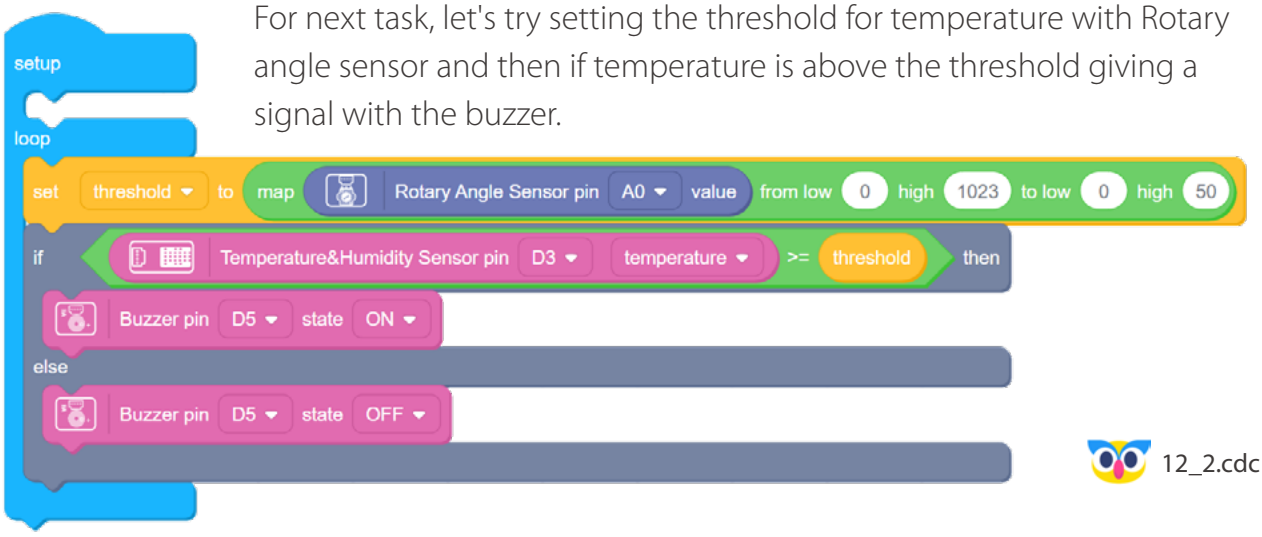
First let's read the values from the sensor and display them on Serial plotter.
When you open Serial Plotter window you'll see the temperature and humidity readings from the sensor - they will not fluctuate as much as with some other sensors, so you can try blowing some hot air on the sensor to see the values change.



12_1.cdc



Task 2: Set threshold for temperature warning sound



For next task, let's try setting the threshold for temperature with Rotary angle sensor and then if temperature is above the threshold giving a signal with the buzzer.

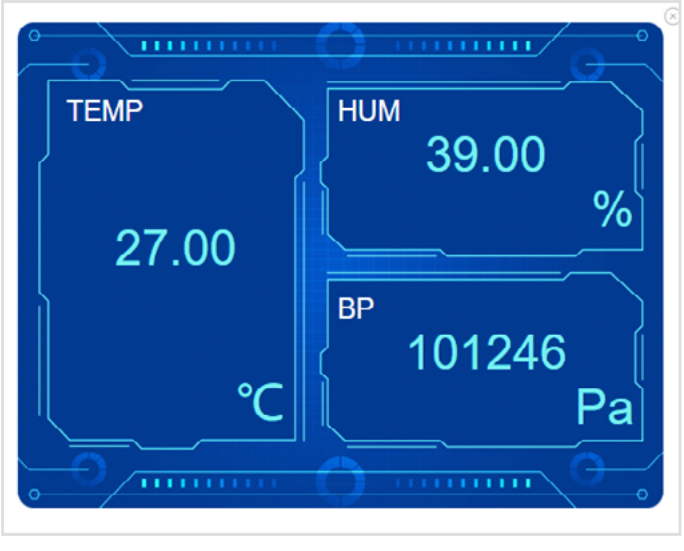
12_2.cdc

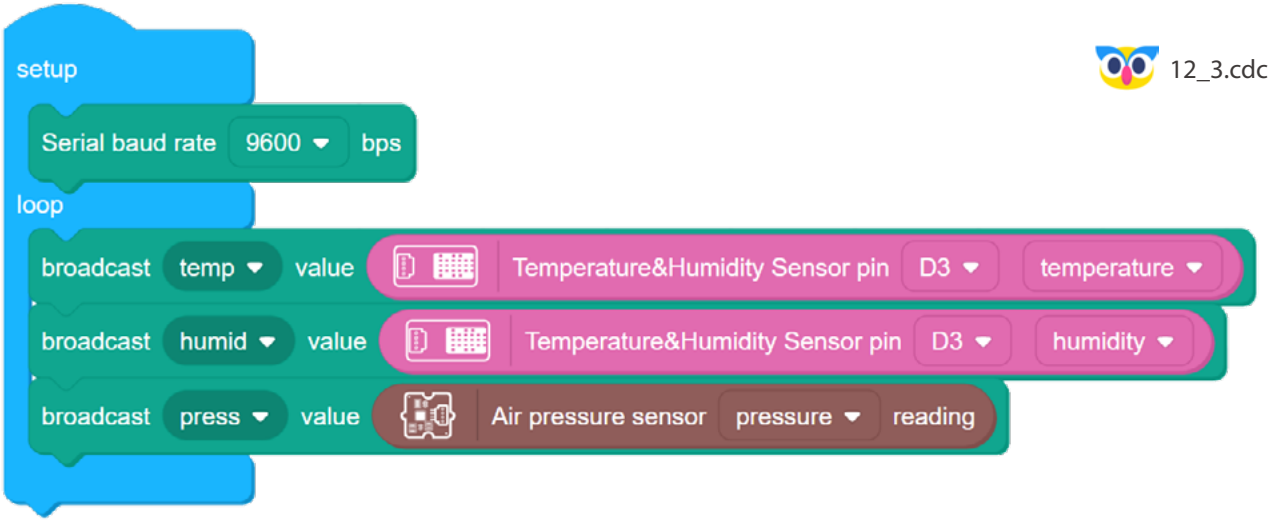
Task 3: Display weather related data on meteostation interface

Finally let's use Codecraft Stage mode to display the sensor data on a sci-fi style web interface for meteostation!

This program will have two parts.

First one for Device mode, where Grove Beginner Kit mainboard is responsible for getting the data from sensors and send them to computer. We use broadcast blocks, which you can find in Serial category, to transfer data over Serial to Codecraft Stage mode. Make the following program and upload to Grove Beginner Kit as usual:

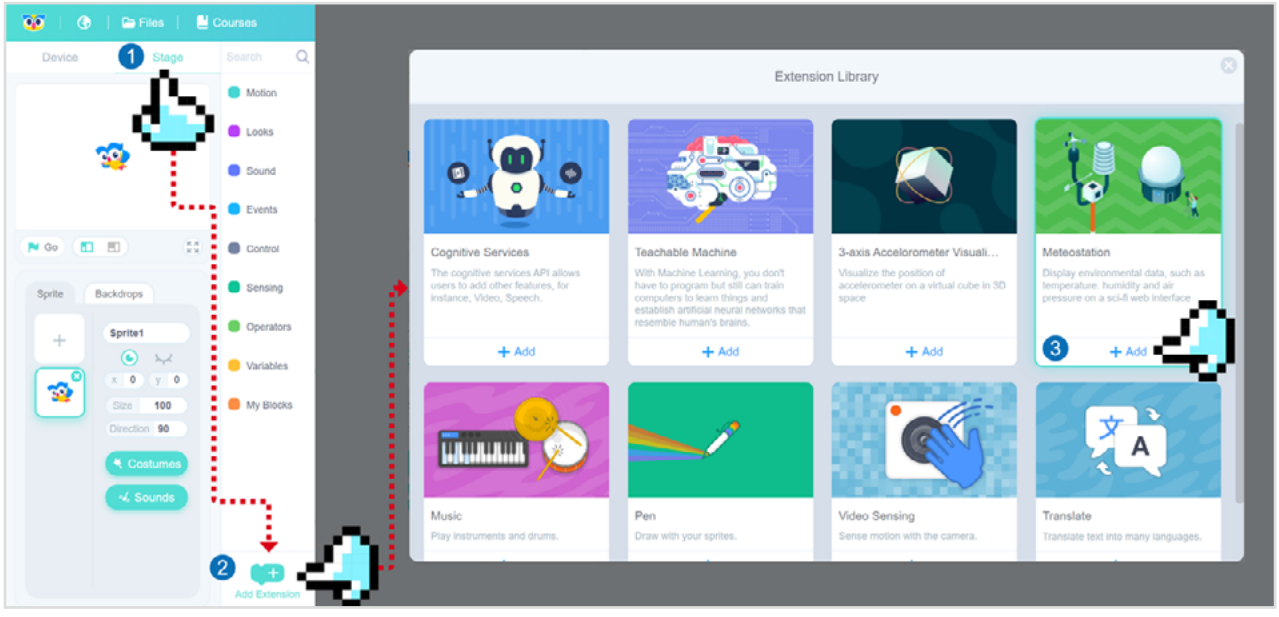


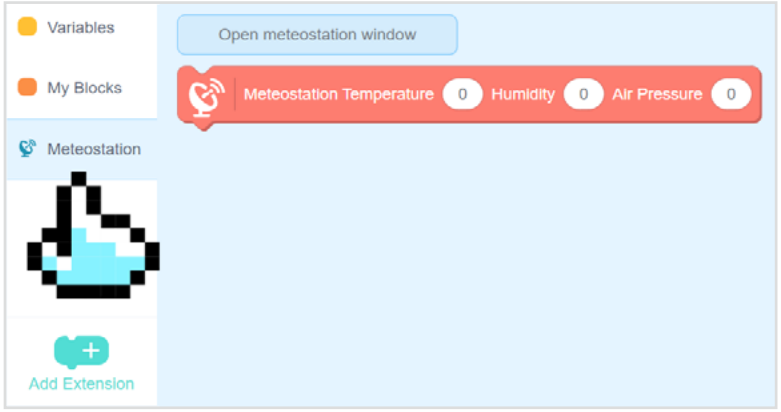


12_3.cdc

Second part is for stage mode, where we use Meteostation extension block to visualize the data received from Device mode. In order to see the Meteostation block.

1. go to Stage mode
2. click on Add Extension at the bottom of Category area
3. choose Meteostation extension.



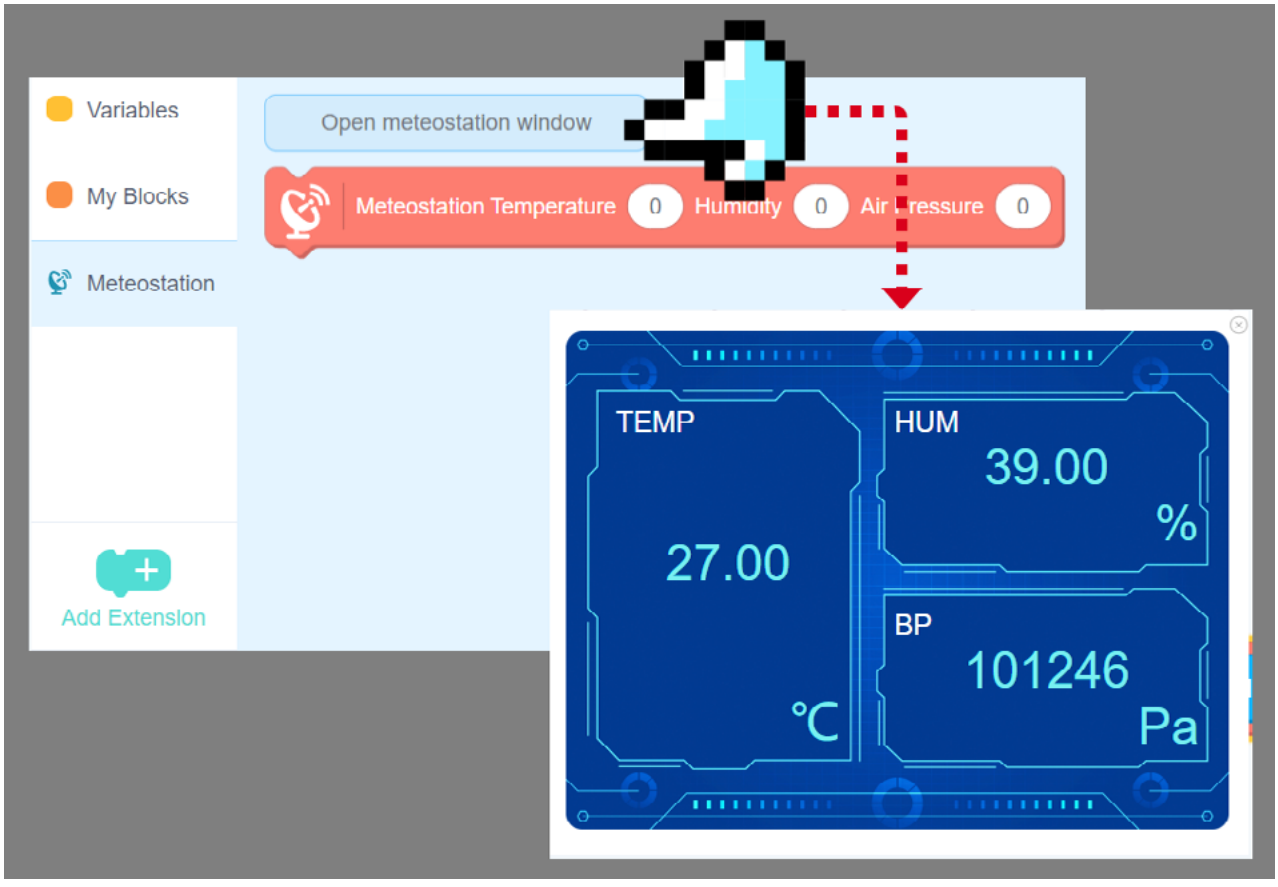


Now there's a meteoration tag.

Broadcast blocks in Stage mode are in Events category.

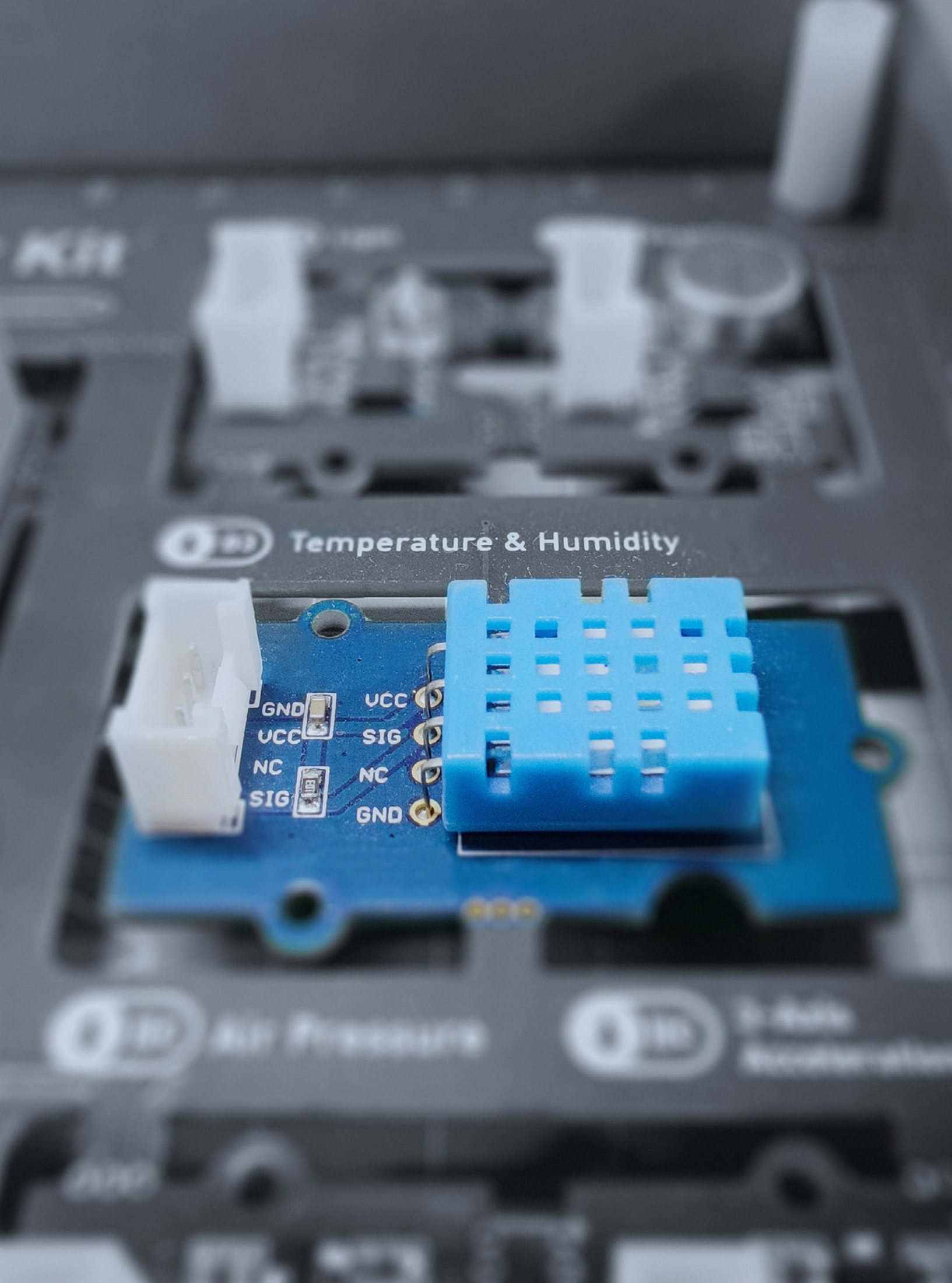
12_3.cdc

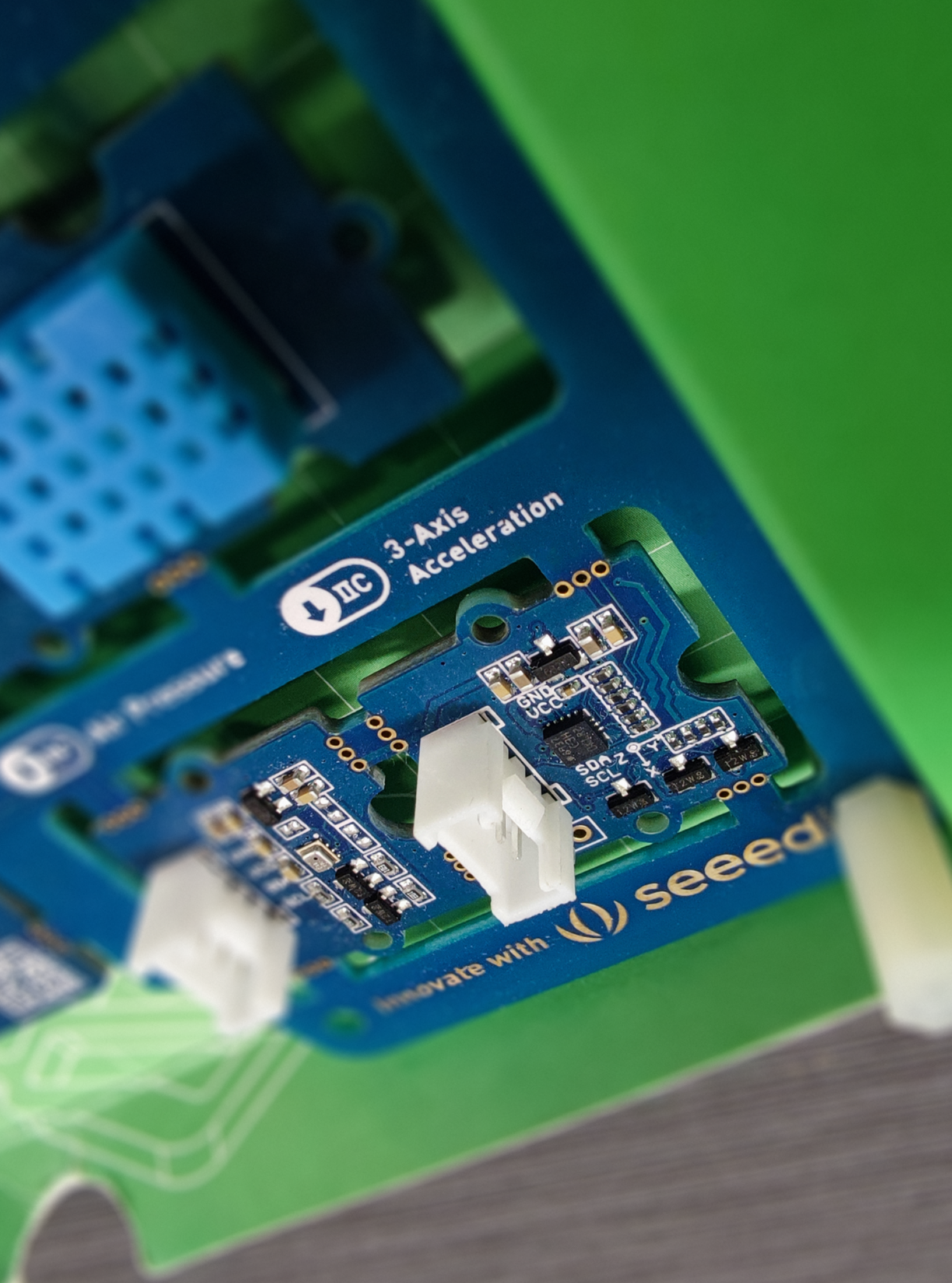
Connect to Arduino in Device mode as you usually do when use Serial monitor or Plotter. Then go to stage mode, press the flag button and click on **Open meteostation window** button in Meteostation category. The data from sensors will be displayed and updated in real time on meteostation interface!



⭐ Outside the Box

- Display temperature and humidity data on OLED screen
- Use button to switch between displaying temperature and humidity on the OLED screen

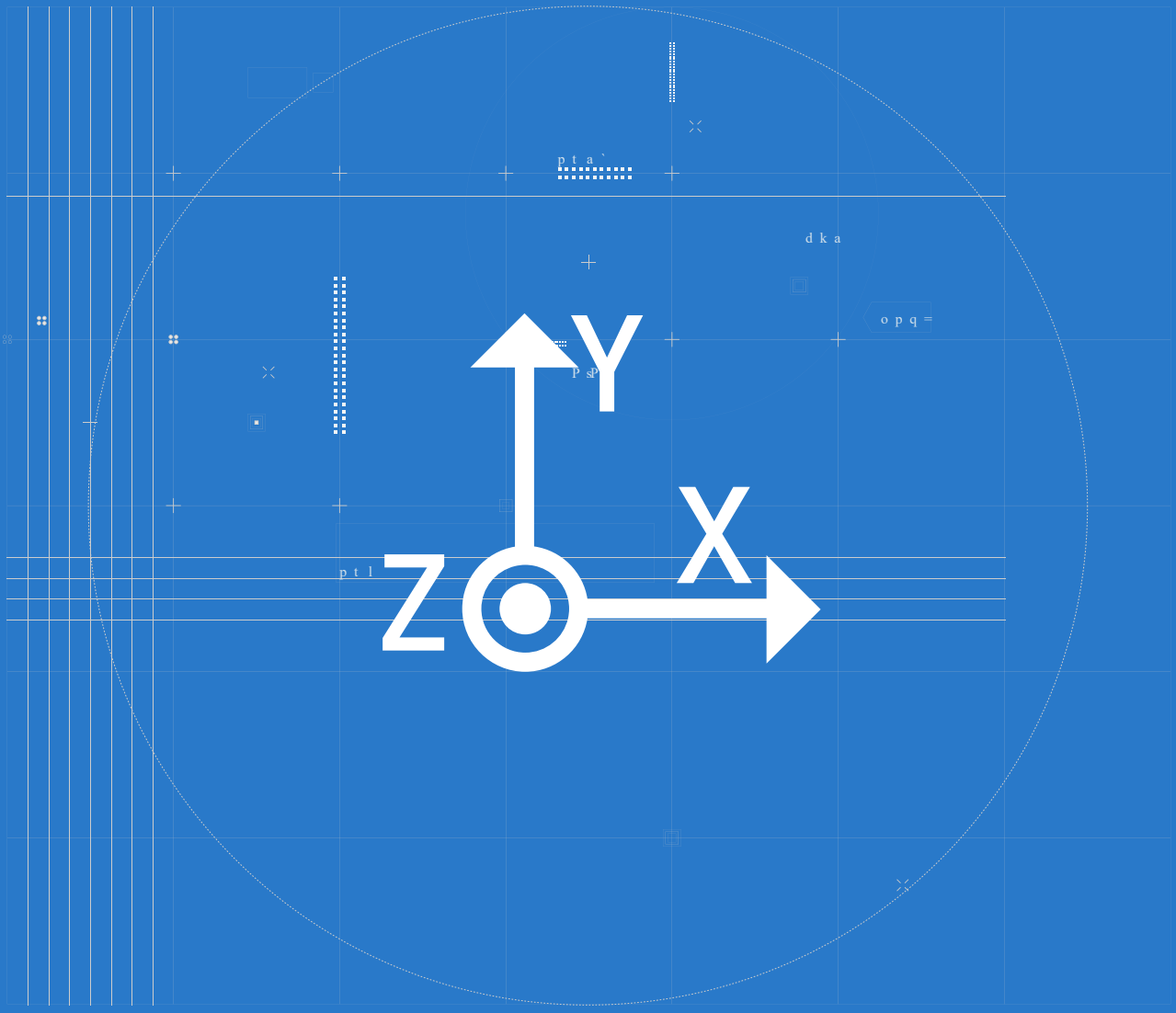

Temperature & Humidity
GND
VCC
NC
SIG
VCC
SIG
NC
GND


I2C
3-Axis
Acceleration
GND
VCC
SDA
SCL
Z
Y
X
Innovate with
seeed

Lesson 13

What Goes Around

When you use a smartphone or a tablet, you've noticed that the screen can change it's orientation automatically - landscape to portrait and back. Similar, you tilt the phone/tablet to control a character or a car when you play games. It means that there is a sensor that can detect the orientation of the device in space - and what sensor could it be? Let's find out!



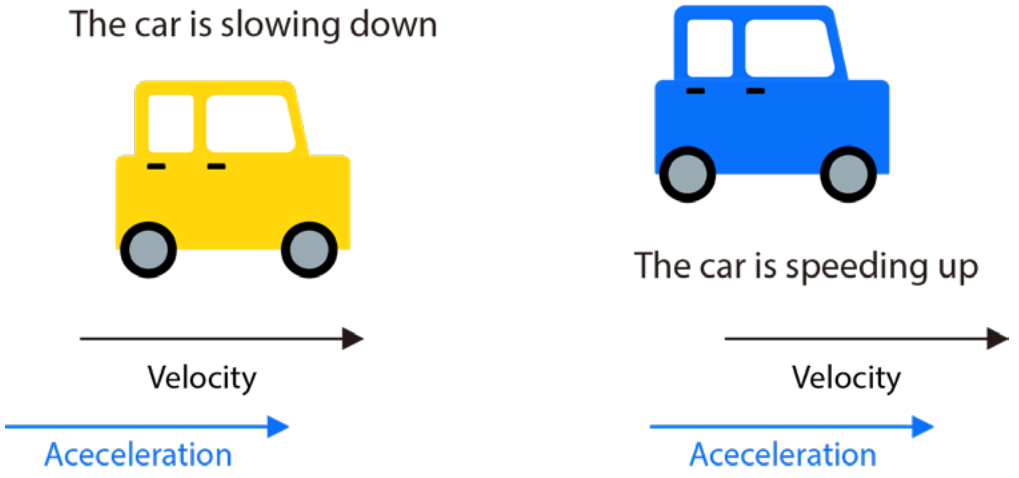
The Big Picture

What is acceleration and how to calculate it

The answer is - accelerometer. As you might guess from the name, accelerometers are devices that measure acceleration, which is the rate of change of the velocity of an object. They measure in meters per second squared (m/s2) or in G-forces (g). A single G-force for us here on planet Earth is equivalent to 9.8 m/s2. One way to find acceleration is from speed(velocity) and time.

Acceleration = speed / time

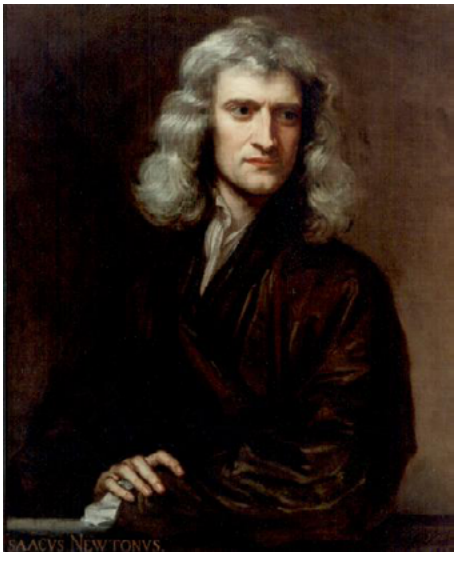
If you have a car that accelerates from a standstill to a speed (or, strictly speaking, velocity) of 100km/h in 5 seconds, the acceleration is the change in velocity or speed divided by the time—so 100/5 or 20 km/h per second. In other words, each second the car is driving, it adds another 20km/h to its speed.



But what if we need to calculate acceleration without waiting for certain time to elapse? If you know about the laws of motion, you'll know that the brilliant English scientist Isaac Newton defined acceleration in a different way by relating it to mass and force:

Acceleration = Force / mass

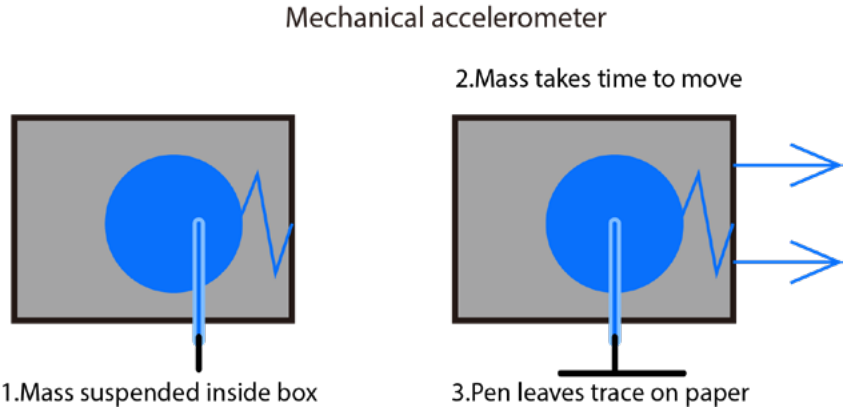
In other words, acceleration is the amount of force we need to move each unit of mass.



Isaac Newton

How to measure acceleration

As with other sensors, there are different kinds of accelerometers and the first ones invented were mechanical ones. The first accelerometer was called the Atwood machine and was invented by the English physicist George Atwood.



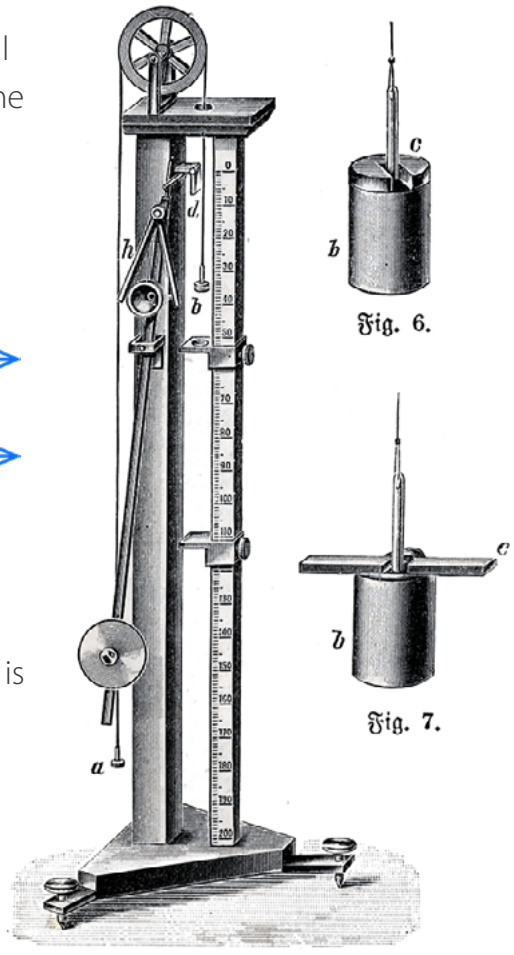


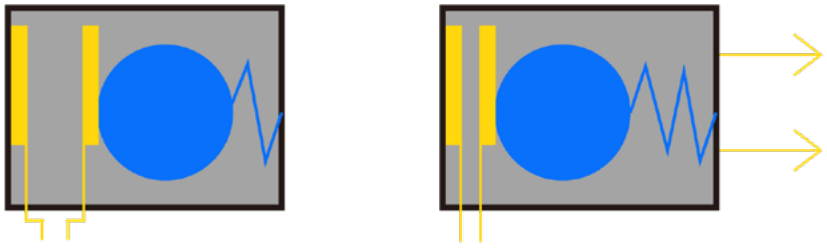
Fig. 6.

Fig. 7.

Fig. 5.
Atwoodsche Fallmaschine

The accelerometer in your phone and in Grove Beginner Kit is a MEMS(Microelectromechanical) accelerometer. The exact module used in Grove Beginner Kit is called 3-Axis Digital Accelerometer(LIS3DHTR).

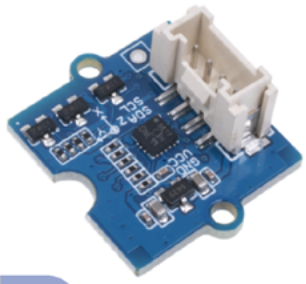
Generally, accelerometers contain capacitive plates internally. Some of these are fixed, while others are attached to minuscule springs that move internally as acceleration forces act upon the sensor.



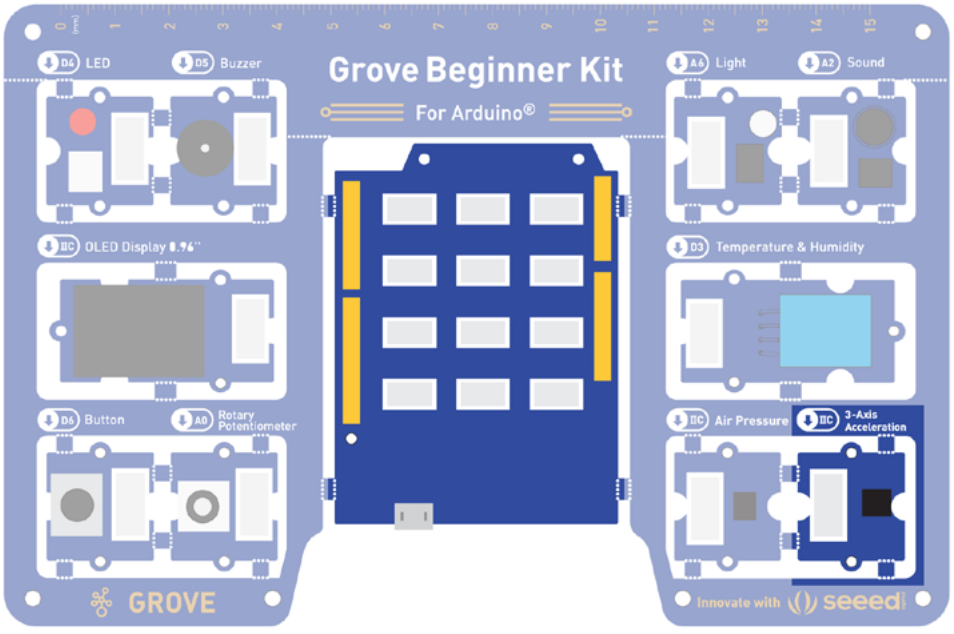
As these plates move in relation to each other, the capacitance between them changes. From these changes in capacitance, the acceleration can be determined.

This is all very well, but if you read everything thoroughly, you might still have a question - how does acceleration relate to device orientation? Even if a device with an accelerometer is not moving, the accelerometer can detect the orientation (tilt) of the device by measuring the acceleration due to Earth's gravity, which is a constant downward force acting on all objects. The accelerometer can determine if the object is parallel to the Earth's surface or if it's tilted. Let's see that ourselves with code!

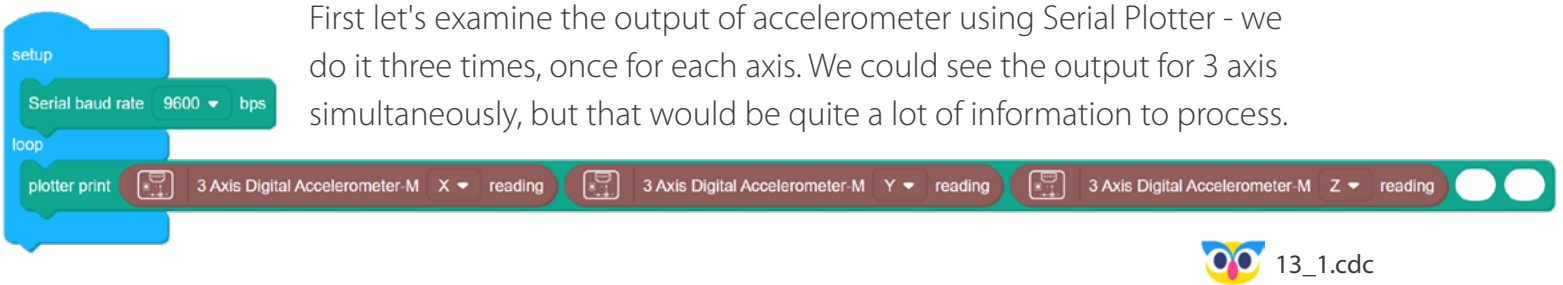
3-Axis Digital Accelerometer sensor module in Grove Beginner Kit



In our Grove Beginner Kit, there is a 3-Axis Digital Accelerometer module.



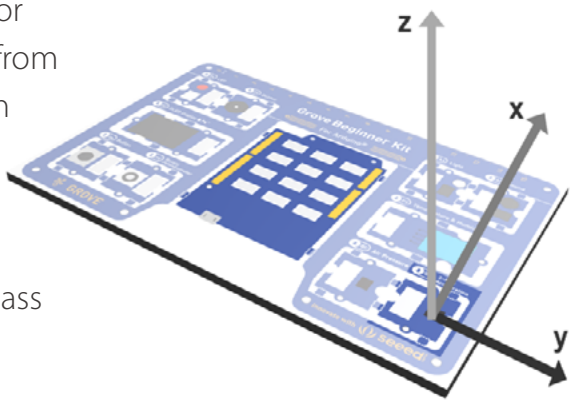
Task 1: Examine the values output by accelerometer with Serial Plotter

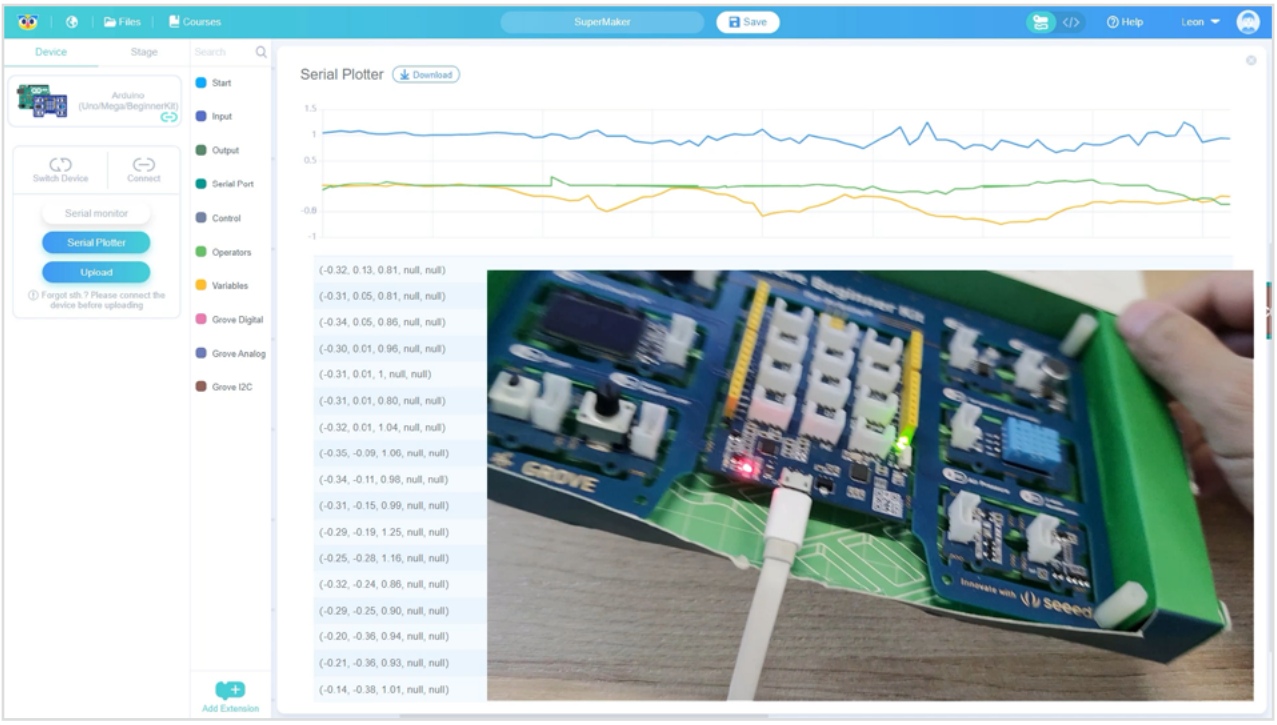


First let's examine the output of accelerometer using Serial Plotter - we do it three times, once for each axis. We could see the output for 3 axis simultaneously, but that would be quite a lot of information to process.

13_1.cdc

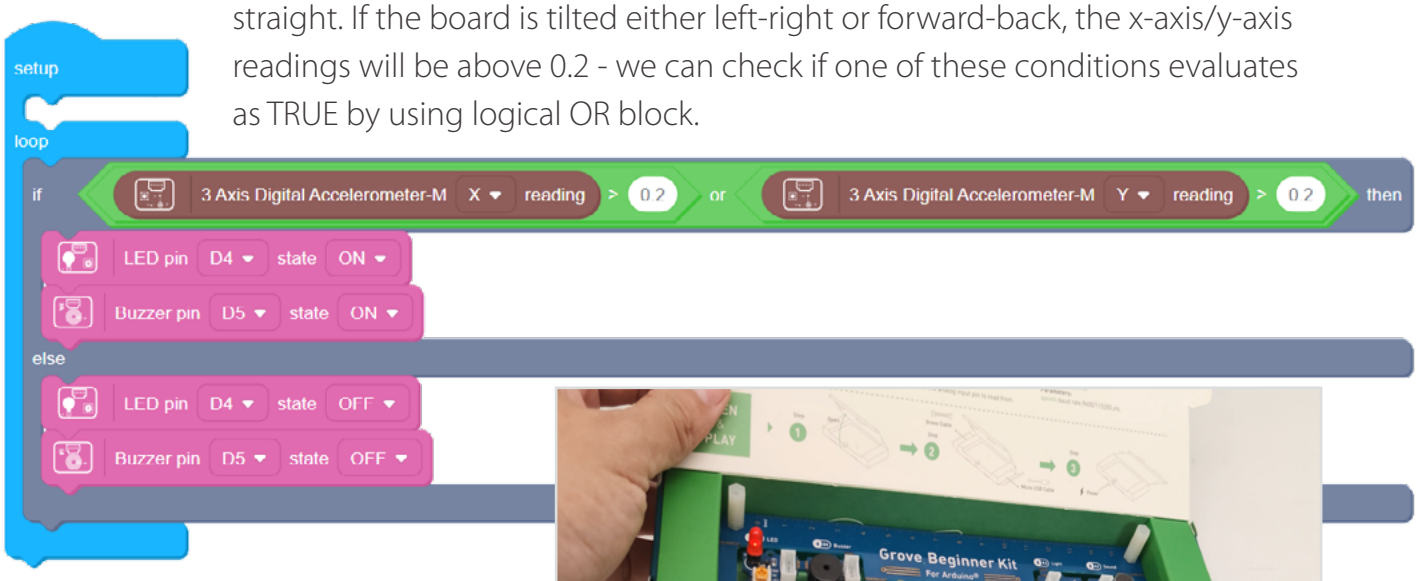
Tilt accelerometer along X-axis(up and down) or Y-axis(right or left) and see the values change from -1 to 1 for that axis. What about Z-axis? Z-axis in our case would be rotation(turning the board left or right) and we cannot measure it with accelerometer, we would need to employ another sensor called magnetometer(or compass for simplicity).





Task 2: Implement tilt checker program

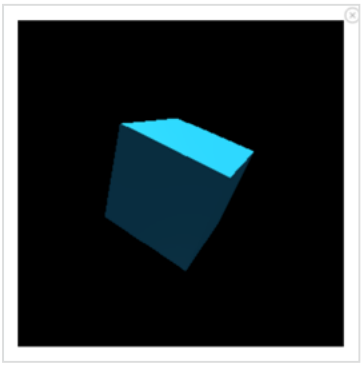
Next, let's write a code that would light an LED and sound the buzzer if the board is not straight. If the board is tilted either left-right or forward-back, the x-axis/y-axis readings will be above 0.2 - we can check if one of these conditions evaluates as TRUE by using logical OR block.

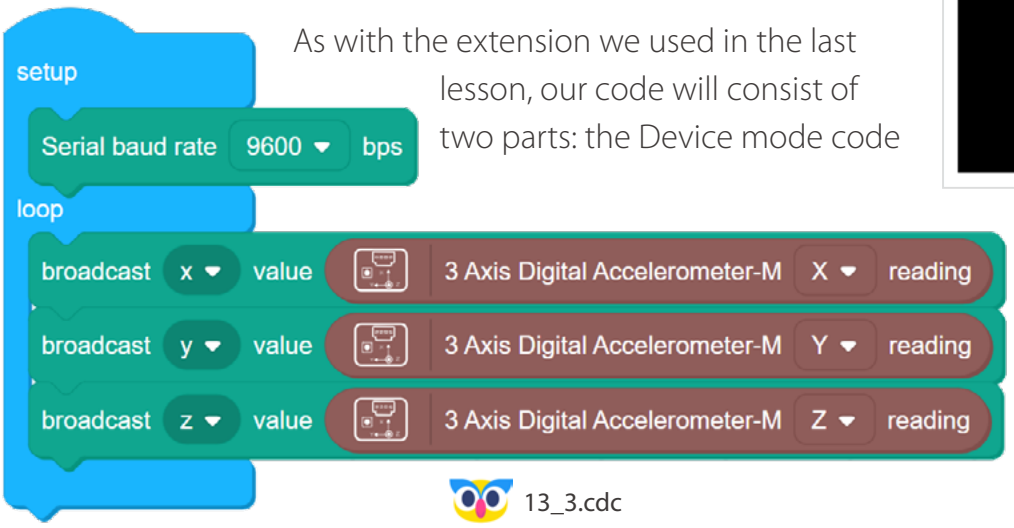


13_2.cdc

Task 3: Visualize Grove Beginner Kit orientation

Finally, let's use another Codecraft Stage mode extension, 3-axis Accelerometer visualization to display rotation of the board using virtual cube, that you can see on your computer screen.

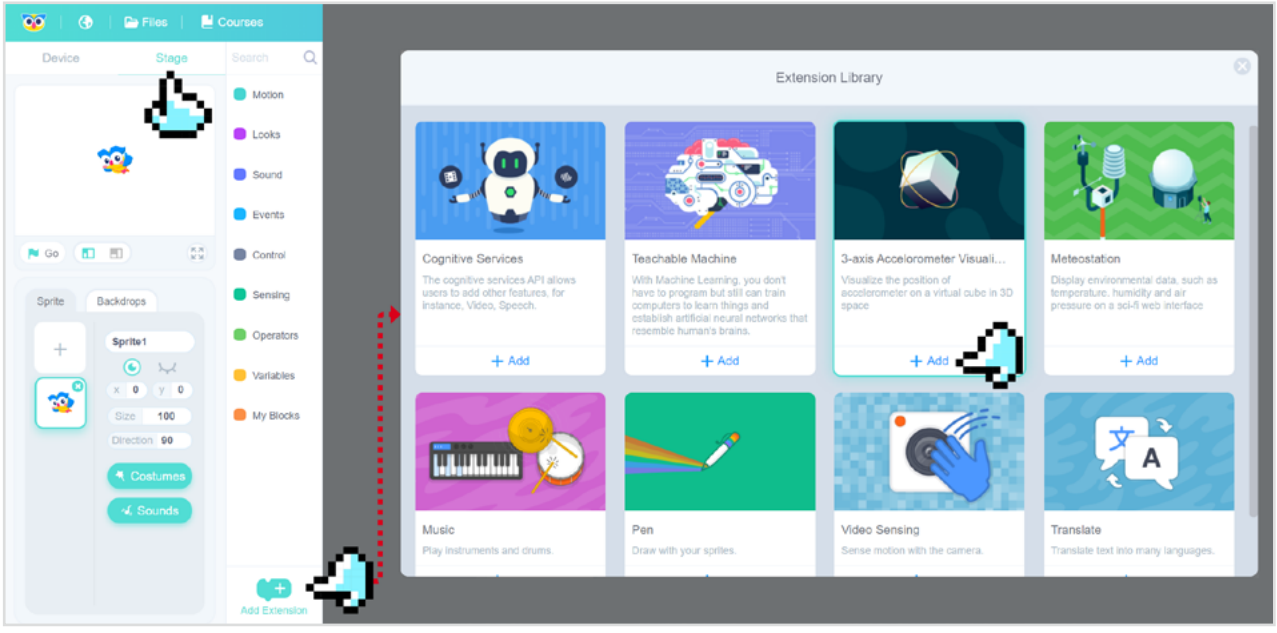




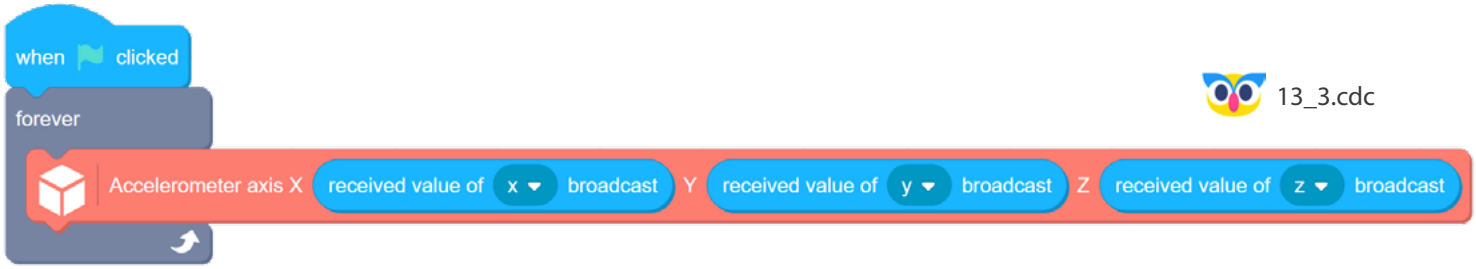
13_3.cdc

As with the extension we used in the last lesson, our code will consist of two parts: the Device mode code and Stage mode code. Make the following program and upload to Grove Beginner Kit as usual.

Second part is for stage mode, where we use 3-Axis Accelerometer visualization extension block to visualize the data received from Device mode. In order to see the 3-Axis Accelerometer visualization block, go to Stage mode, click on Add Extension at the bottom of Category area, then choose 3-Axis Accelerometer visualization extension.

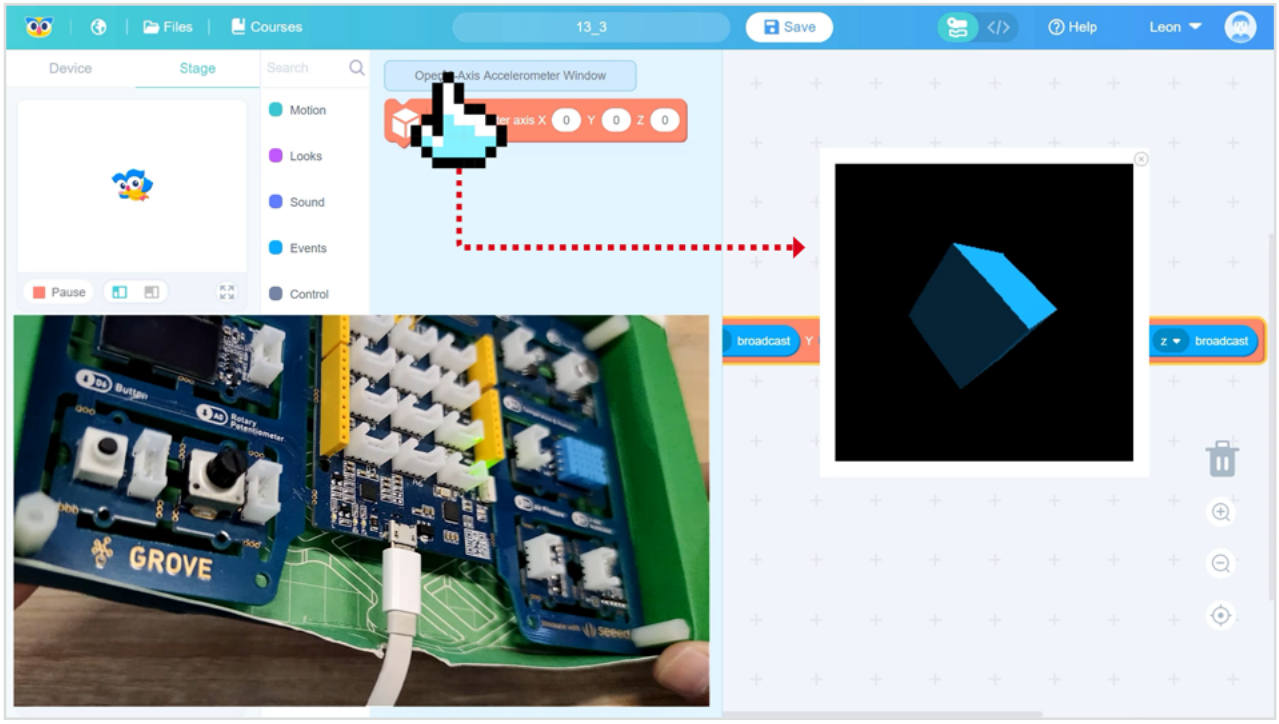


Upload the following code to the board:



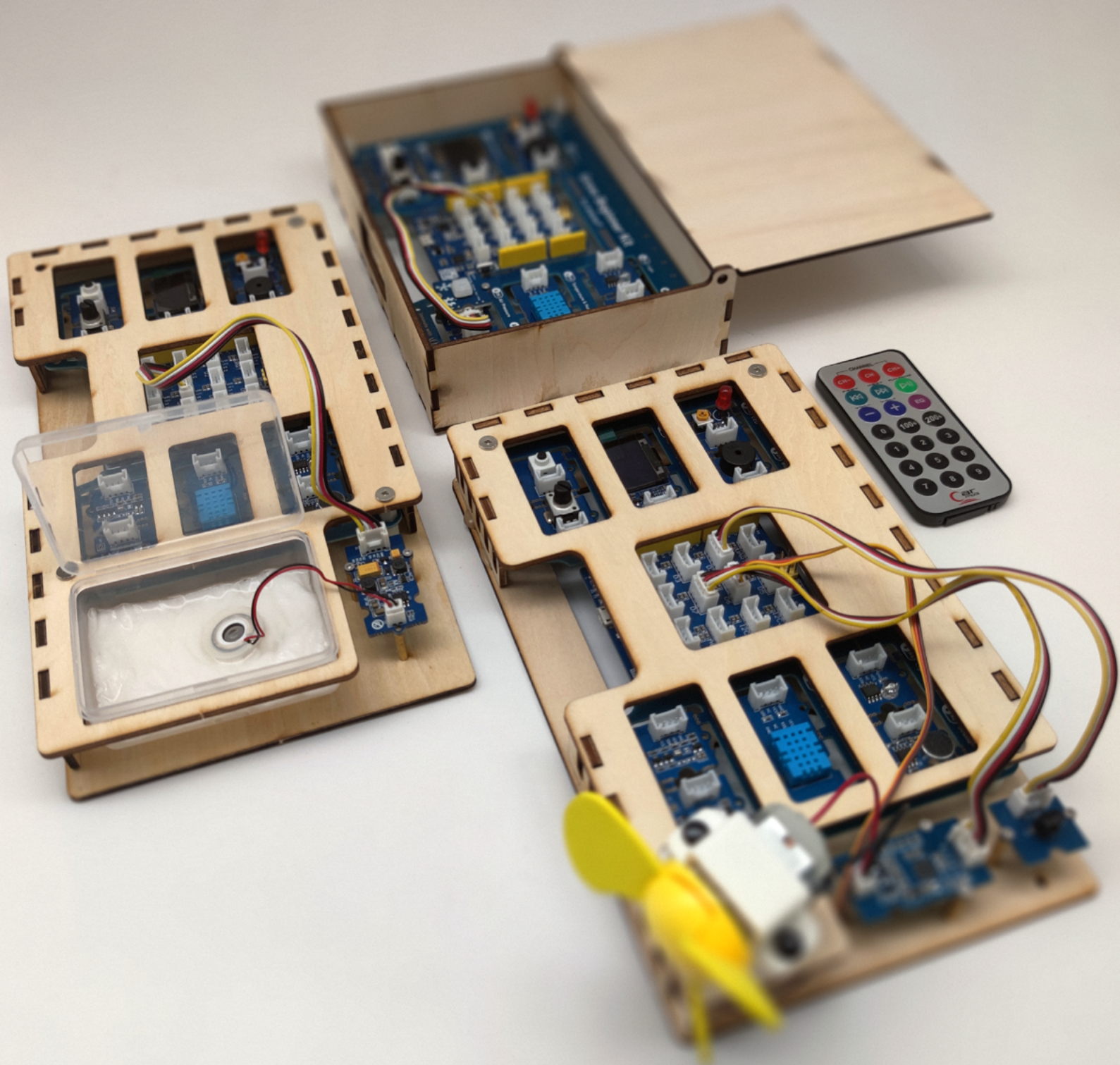
13_3.cdc

Connect to Arduino in Device mode as you usually do when use Serial monitor or Plotter. Then go to stage mode, press the flag button and click on **Open 3-Axis Accelerometer window** button in 3-Axis Accelerometer visualization category. The data flows from the sensor to Grove Zero Beginner kit mainboard and then to computer, where it is visualized in the browser window. Pretty cool isn't it?



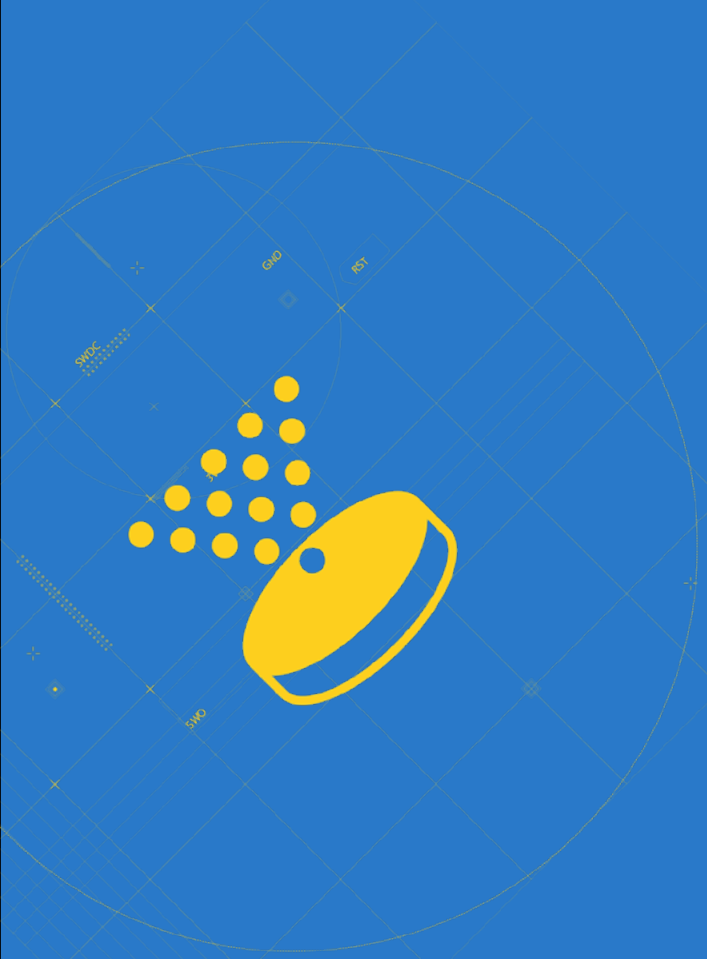
⭐ Outside the Box

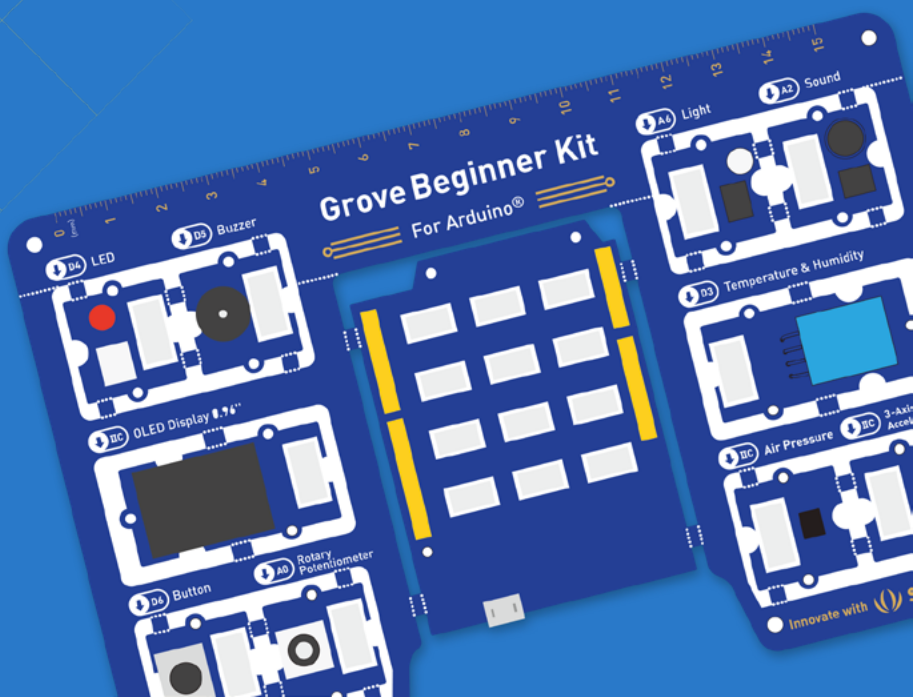
- Use 3-axis accelerometer visualization with manually input values.
- Re-map values of X-axis readings to 0-255 range and control the bluntness of an LED.
- Use multiple if blocks to make Grove Beginner Kit emit different tone sounds depending on board tilt.

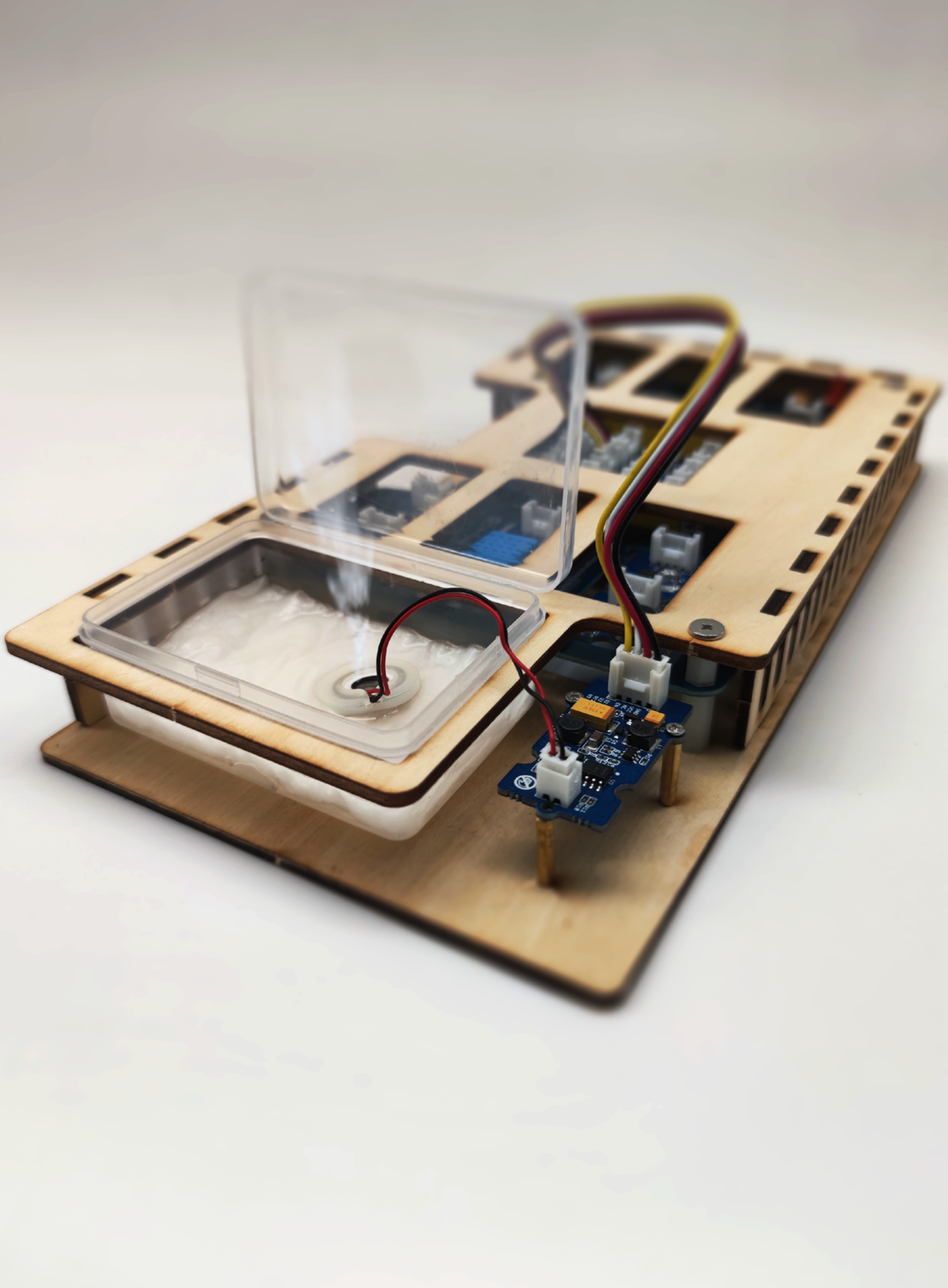


Additional Projects With Grove Beginner Kit Expansion Pack

Lesson 14 Project 1: Humidity Control
Lesson 15 Project 2: Turning Fan
Lesson 16 Project 3: Burglar Alarm







Lesson 14

Humidity Control

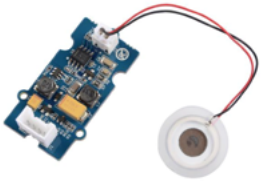
In the next lessons we will learn how to use the additional modules from Grove Beginner Kit for Arduino Education Add-on Pack to make simple and interesting projects, that can be used in everyday life. The extra modules on Grove Beginner Kit for Arduino Education Add-on Pack are not wired to mainboard by default, so you will need to connect them using Grove cables. Additionally, the projects in the following lessons will require you to make structural parts - we provide reference designs that can be made with laser cutter.

The Big Picture

Education Add-on Pack package content

Inside of Grove Beginner Kit for Arduino Education Add-on Pack you can find the following modules.

Actuators



Grove - Water Atomization v1.0 x1

Grove - Mini Fan v1.1 x1

Grove - Servo x1

Sensors

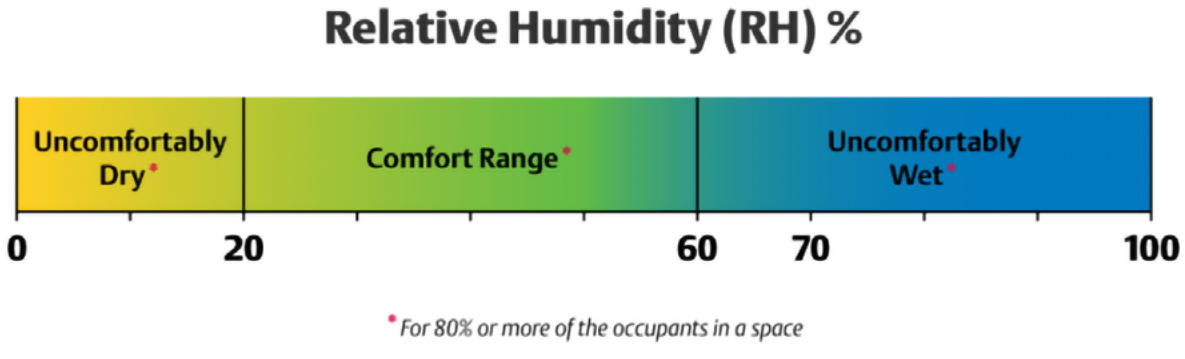
Grove - Ultrasonic Distance Sensor x1

Grove - IR (Infrared) Receiver x1

Grove - mini PIR Motion Sensor x1

Humidity explained in detail

We have learned that humidity (or to be precise a relative humidity) is it is the amount of water vapor in the air. If there is a lot of water vapor in the air, the humidity will be high. As a logical conclusion if we'd like to increase the air humidity we need to put more water vapor in the air. Sufficient humidity levels are important for health and well-being of living beings. Low humidity (dry air) causes eyes to become dry and irritated, skin gets flaky and itchy. Adding to that, humidity inflames and dries out the mucous membrane lining the respiratory tract. As a result, the risk of cold, flu and other infections is substantially increased.

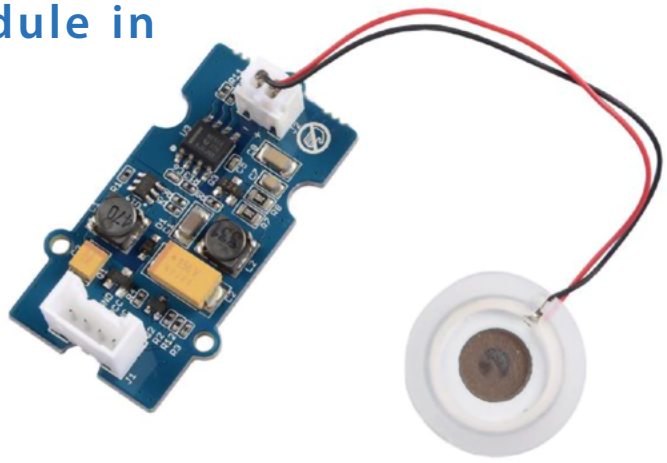


How to humidify the air?

The ideal indoor humidity range is between 40% and 60%. If we detect that it is lower than this range with Temperature & Humidity sensor, how can we raise it? In other words, how can we put more water vapor in the air. One way would be to heat water, so it transforms from its liquid state to gas. Another way is an ultrasonic humidifier, which uses a ceramic diaphragm vibrating at an ultrasonic frequency to create water droplets that silently exit the humidifier in the form of cool fog. Ultrasonic humidifiers use a piezoelectric transducer to create a high frequency (1-2 MHz) mechanical oscillation in a film of water. This forms an extremely fine mist of droplets about one micron in diameter, that is quickly evaporated into the air flow. This is exactly the working principle of Grove Water Atomization module, that we will use to build an intelligent humidifier system project in this lesson.

Grove Water Atomization module in Education Add-on Pack

Grove - Water Atomization is a perfect module for you to quickly DIY a humidifier with Arduino It has Grove interface which makes it easy to be integrated into plenty of applications in plug and play way. Besides a humidifier, you can develop more advanced and interesting projects with digital scent technology and any other situations in which atomization required.



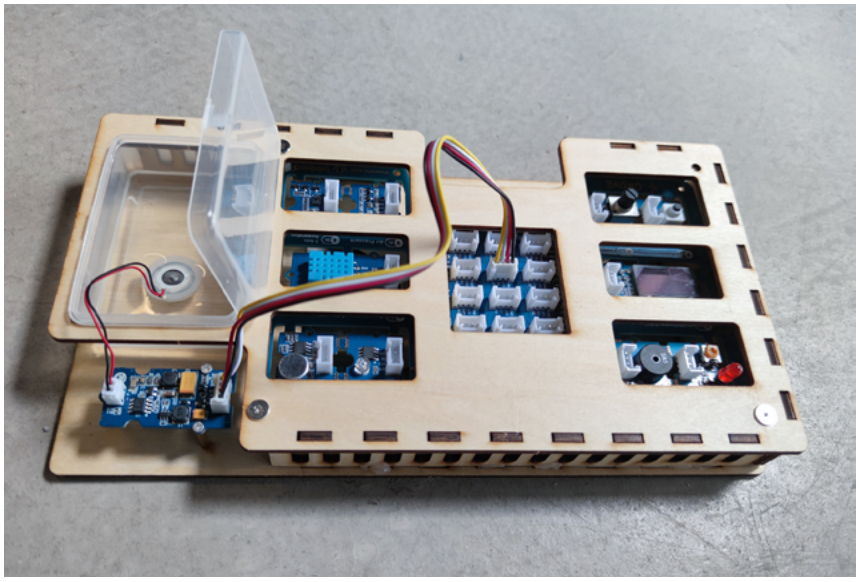
Comprehensive Project: Smart Humidifier

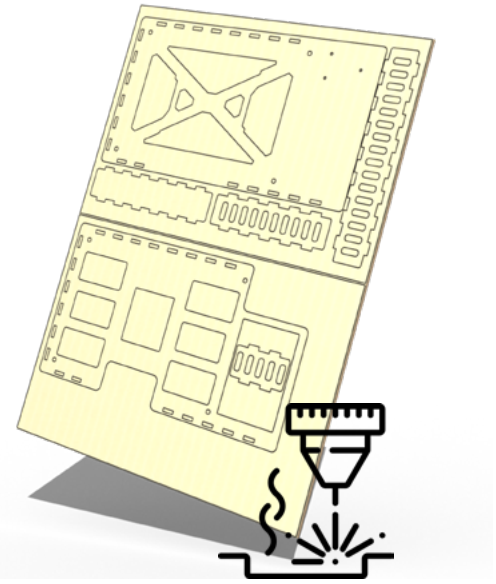
Assemble the structure

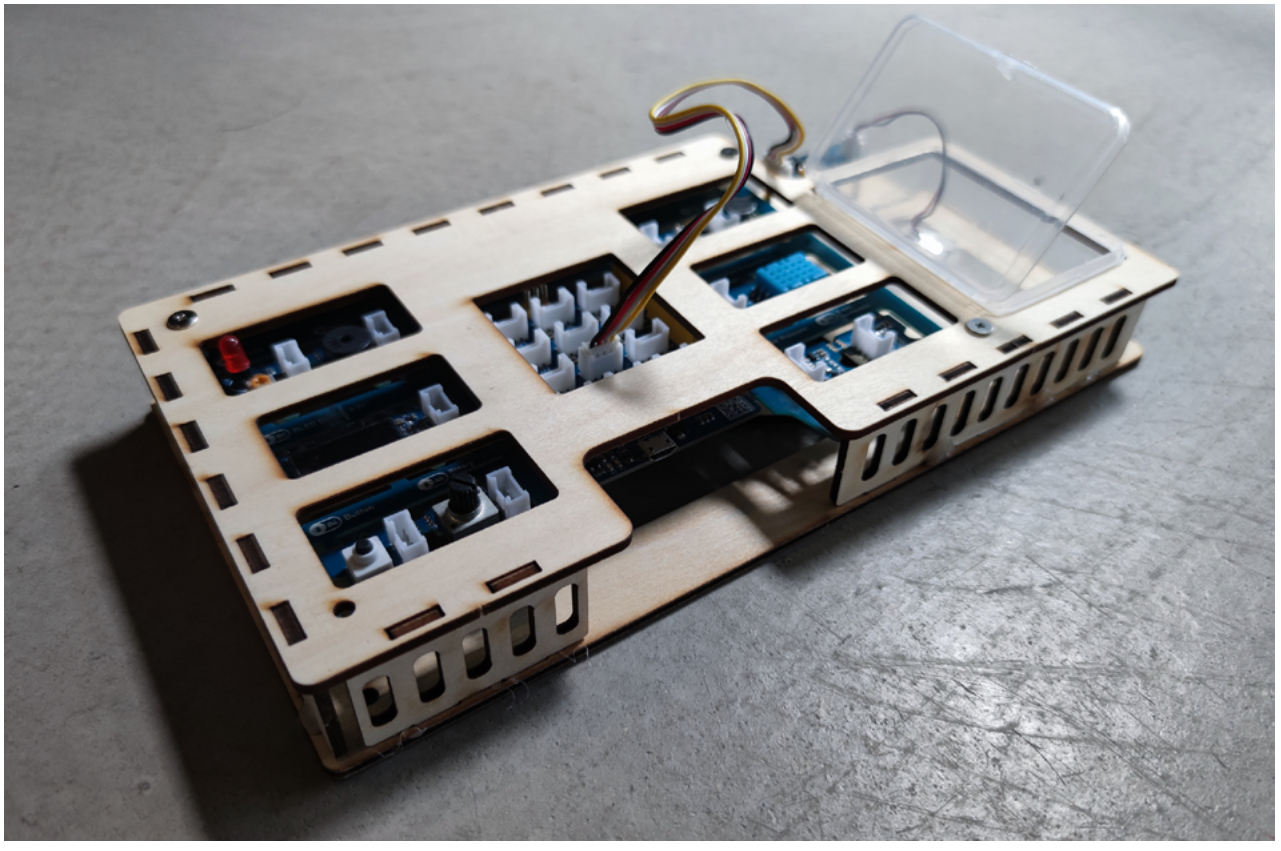
First of all assemble together the project structure from laser cut wooden parts, designs for which you can find here:

Grove Beginner Kit-L14-Humidity Control.dfx

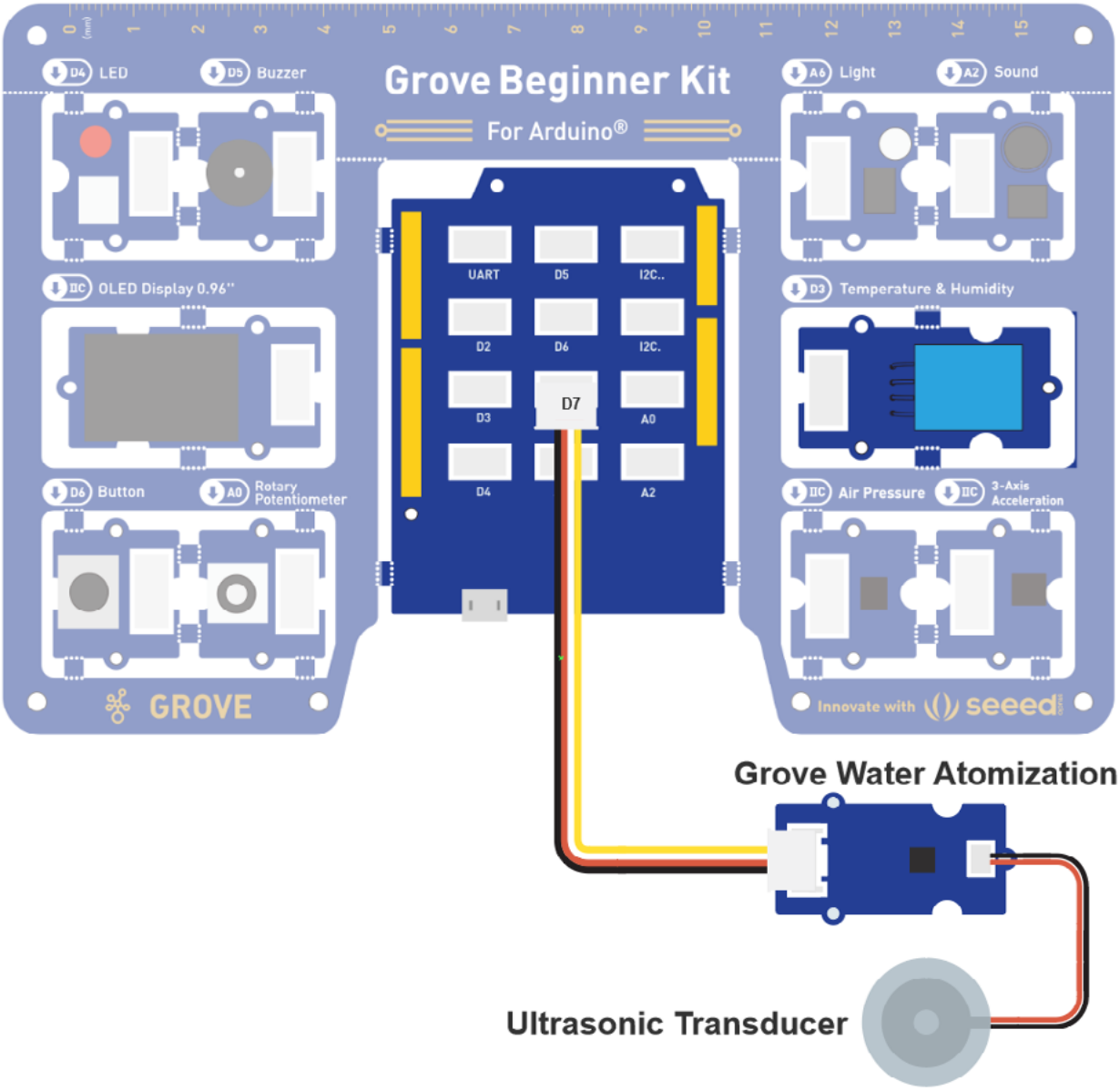
You can see the final assembly result in below pictures:





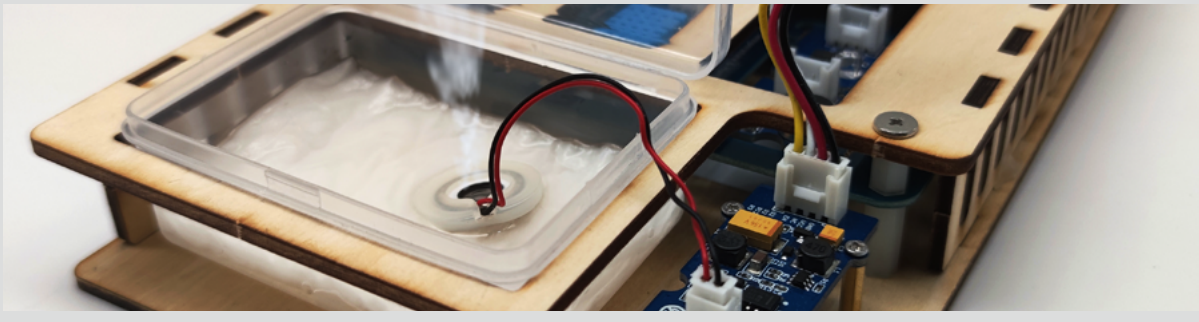


Connect humidifier as shown in this diagram.



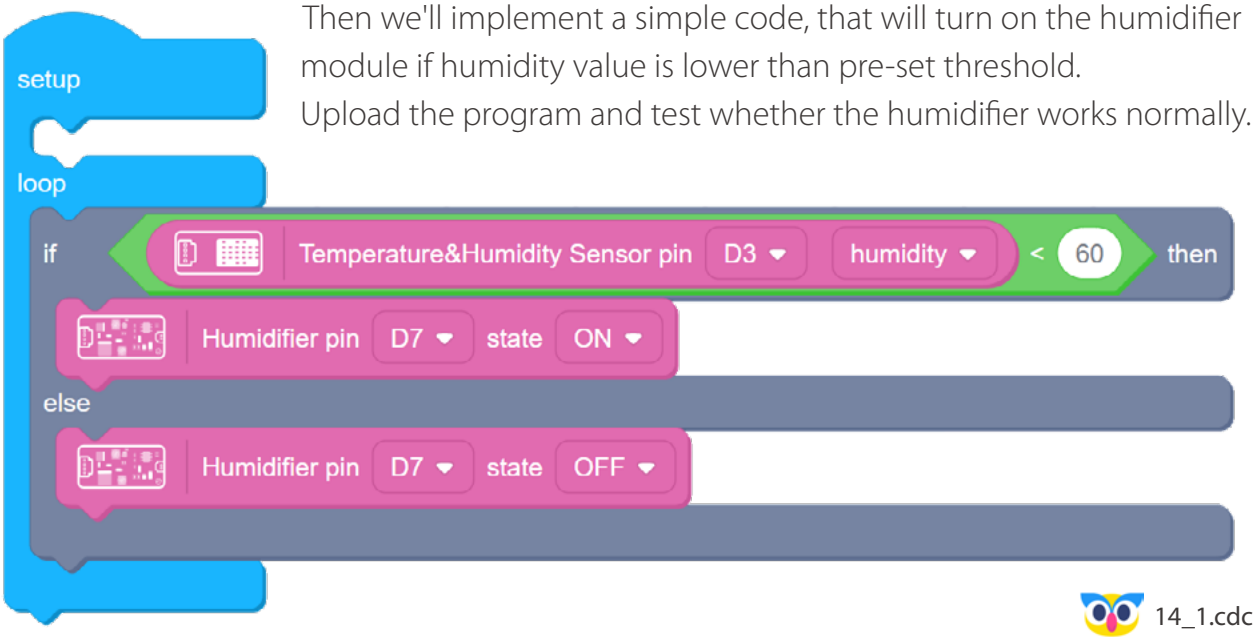
Note

The bottom side is the side with hollow which is supposed to face downside. Let bottom of transducer plate sink into the water and keep top side above water. The function of tissue is lead water to the transducer and keep upper side of transducer above water.



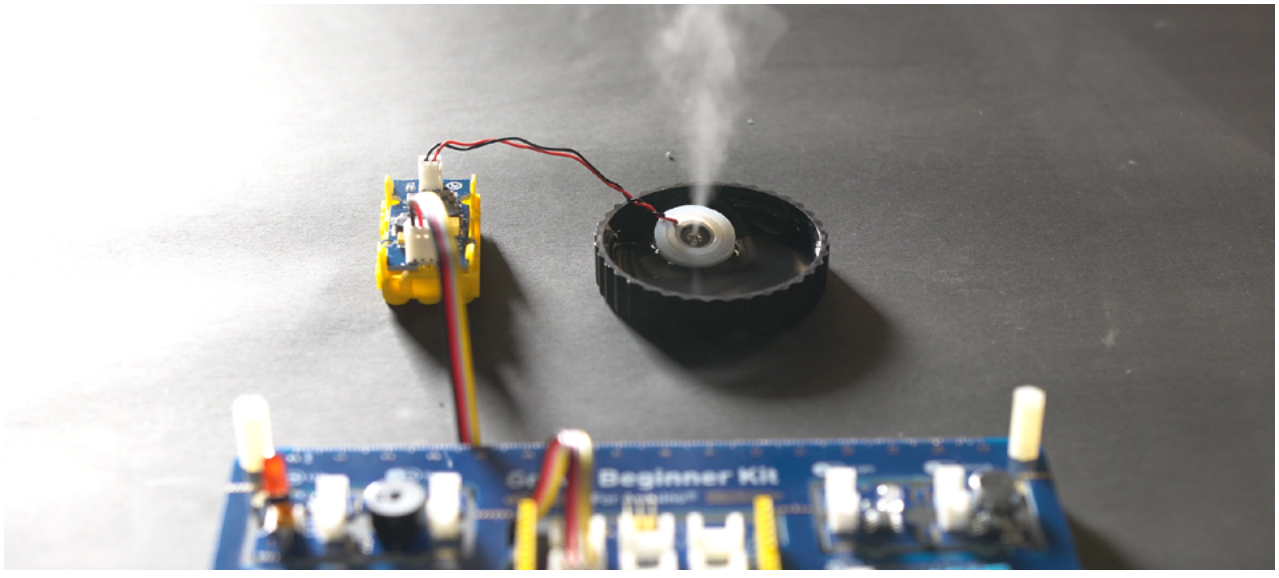
Coding the Solution

Step 1: Switch humidifier on and off depending on environment humidity



Then we'll implement a simple code, that will turn on the humidifier module if humidity value is lower than pre-set threshold.
Upload the program and test whether the humidifier works normally.

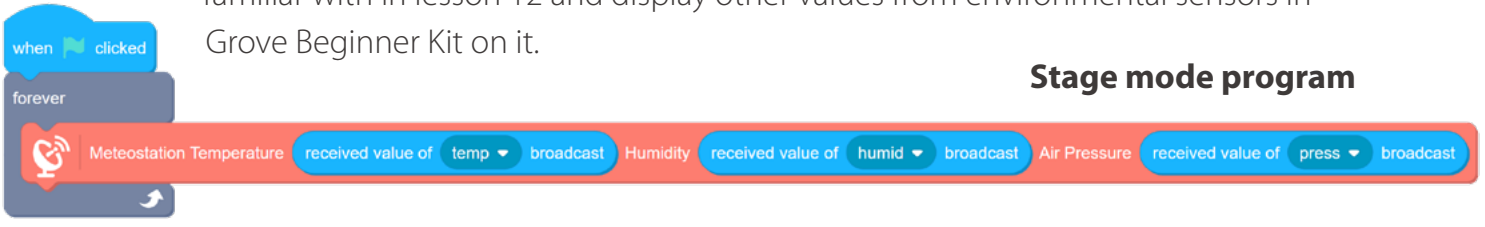
14_1.cdc



Step 2: Add meteostation interface

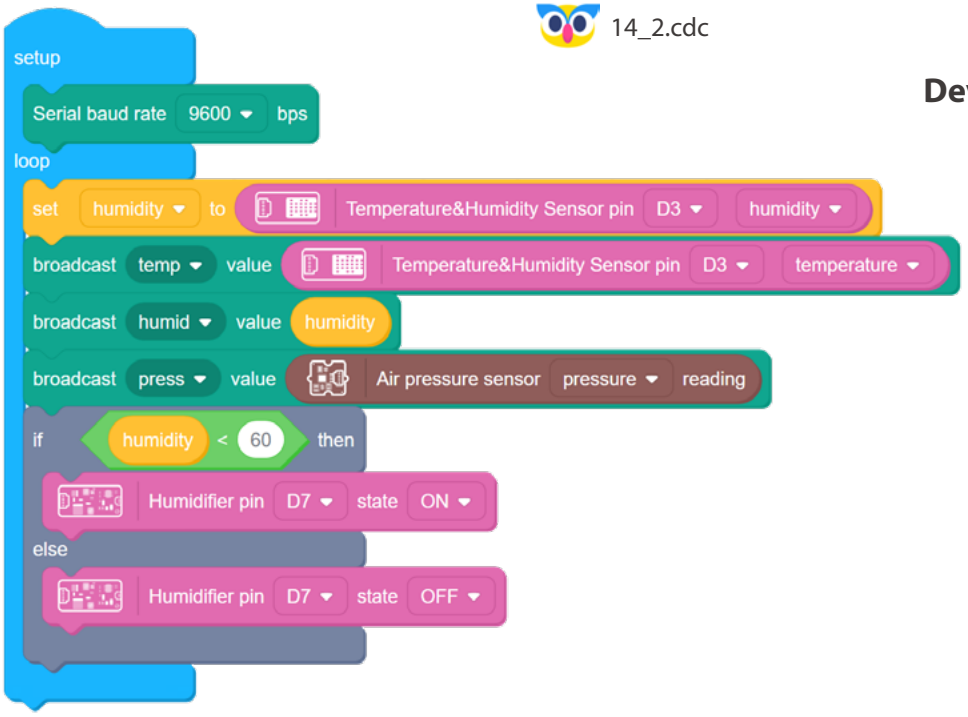
And finally, for the sake of aesthetics, let's use that meteostation web interface that we got familiar with in lesson 12 and display other values from environmental sensors in Grove Beginner Kit on it.

Stage mode program



14_2.cdc

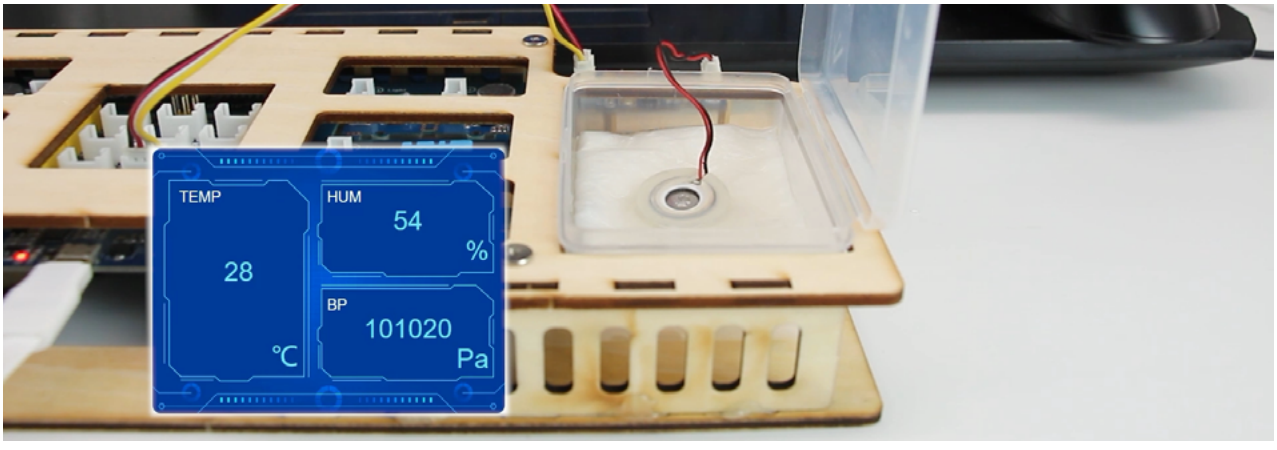
Device mode program



Step 3: Upload the program and open the "Meteostation"

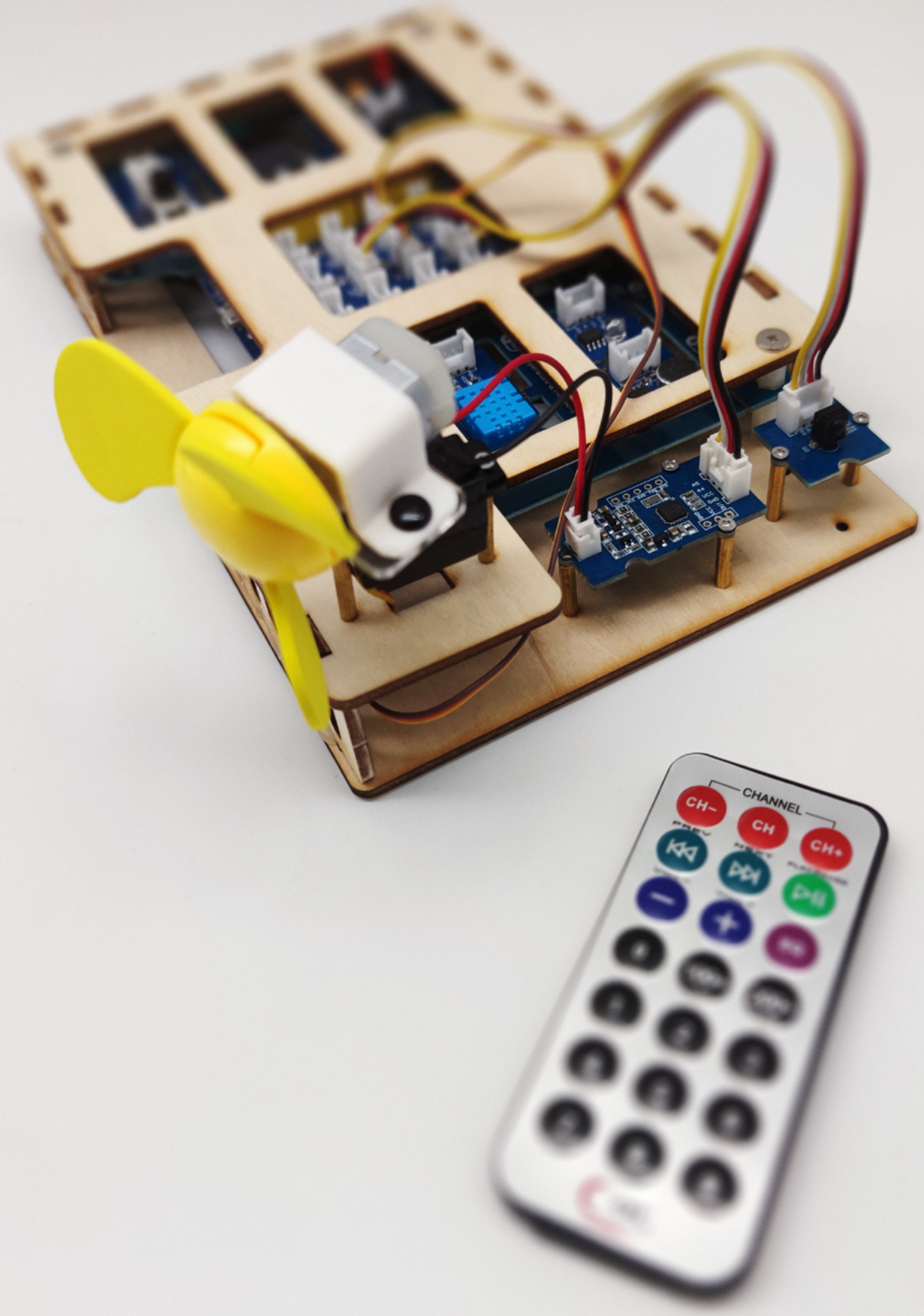
After programming, follow the steps below:

- Use the USB cable to connect the starter kit to the computer
- Upload programs in Codecraft's device mode
- Connect the device in Codecraft's device mode
- Run the program in the stage mode of Codecraft
- In the stage mode of Codecraft, open the Meteostation window



⭐ Outside the Box

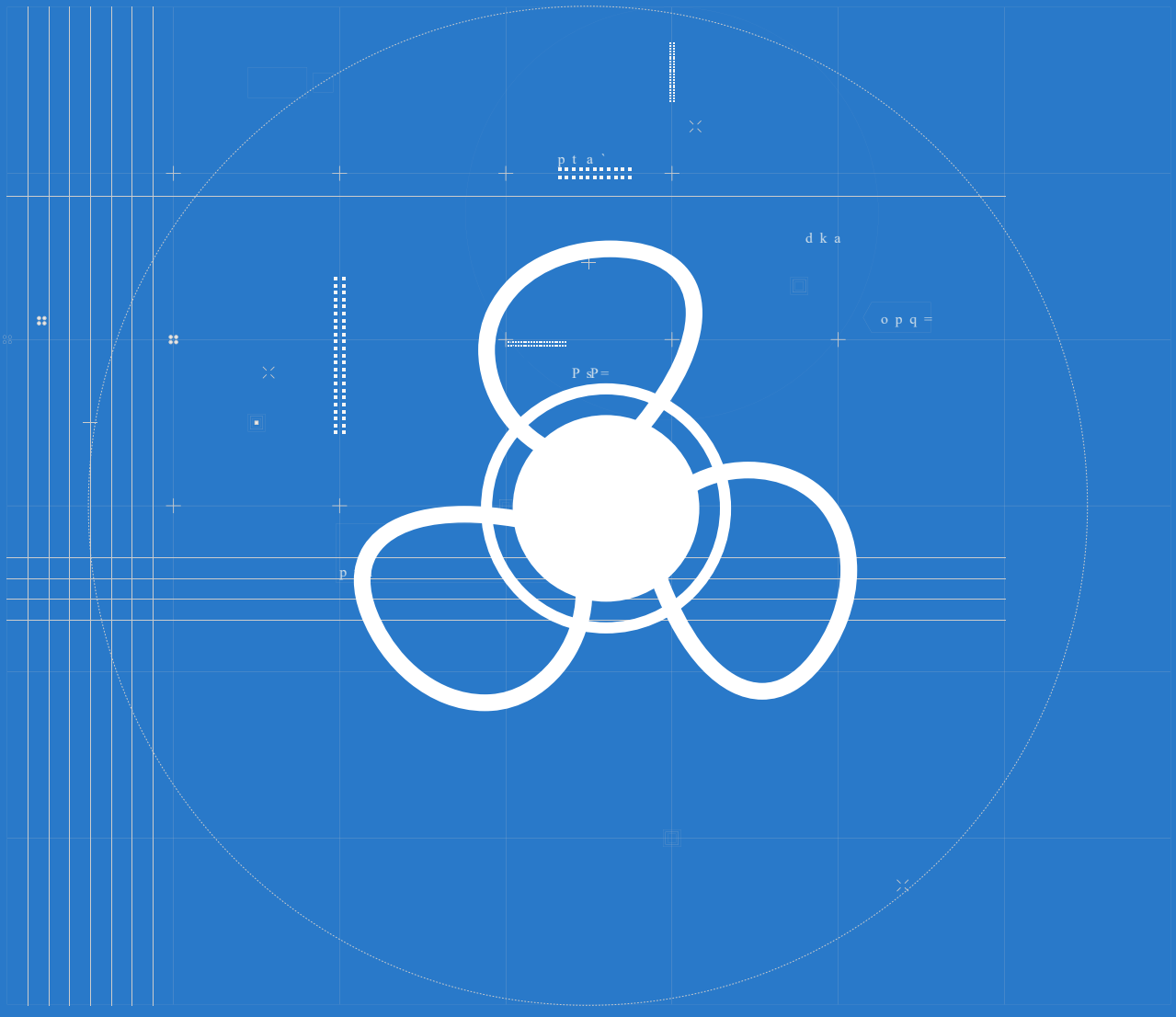
- Use Rotary Angle Sensor to control the humidifier activation threshold
- Use the button module to switch on/off the system
- Display the humidity values on the OLED screen


CHANNEL
CH-
CH
CH+

Lesson 15

Turning Fan

In our next project we will try to re-create a device some of you often see and use in everyday life - a remote controlled fan. For our project we are going to use three modules fro Grove Beginner Kit Expansion Pack - a Mini-Fan, a Infrared Receiver and a Servo.

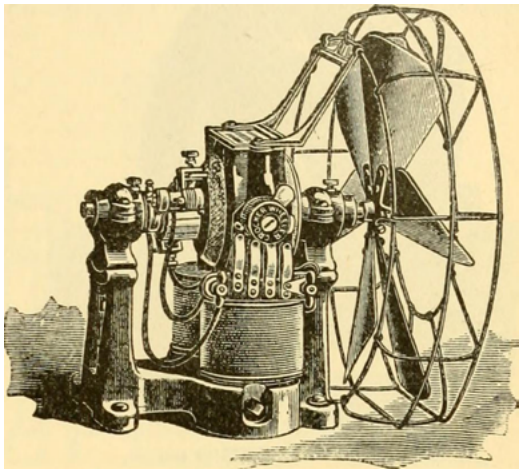


The Big Picture

How do fans work

Fan is an old invention and it's hard to say when did people came up with the idea that waving with a hand or an object in the hand can make one feel cooler. The designs for rotary fan existed as early as year 180 CE (common era, same meaning as AD), but first practical applications for ventilation came with the advent of practical steam power. Between 1882 and 1886 Schuyler Wheeler invented a fan powered by electricity.

Mechanically, a fan can be any revolving vane, or vanes used for producing currents of air. Fans produce air flows with high volume and low pressure.

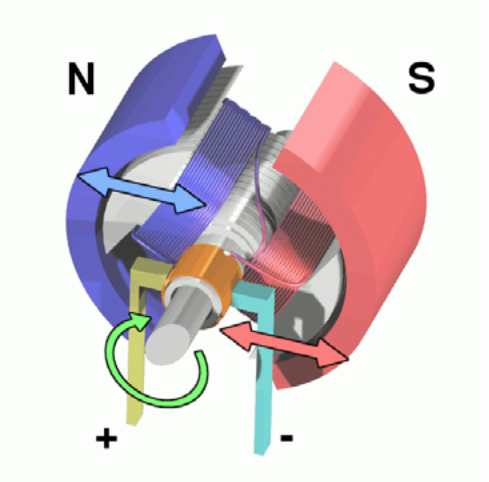


While fans are often used to cool people, they do not cool air (electric fans may warm it slightly due to the warming of their motors), but work by evaporative cooling of sweat and increased heat convection into the surrounding air, due to the airflow from the fans.

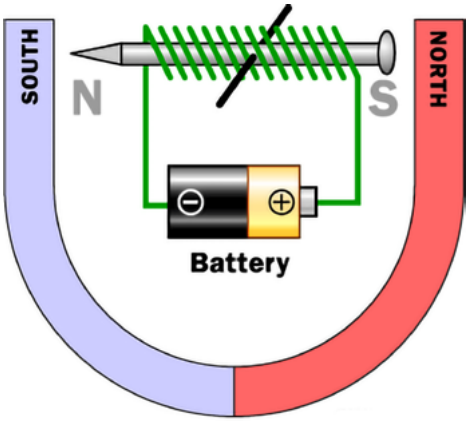
What is the working principle of this now common device and how can we make it remote controlled? Let's find out!

The working principle of DC motor

Mini-fan module consists of a motor with a motor driver and a brightly colored soft blades. The basic idea of an electric motor is really simple: you put electricity into it at one end and an axle (metal rod) rotates at the other end giving you the power to drive a machine of some kind.



An electromagnet is the basis of an electric motor. Say that you created a simple electromagnet by wrapping 100 loops of wire around a nail and connecting it to a battery. The nail would become a magnet and have a north and south pole while the battery is connected.



Now say that you take your nail electromagnet, run an axle through the middle of it and suspend it in the middle of a horseshoe magnet as shown in the figure above. If you were to attach a battery to the electromagnet so that the north end of the nail appeared as shown, the basic law of magnetism tells you what would happen: The north end of the electromagnet would be repelled from the north end of the horseshoe magnet and attracted to the south end of the horseshoe magnet. The south end of the electromagnet would be repelled in a similar way. The nail would move about half a turn and then stop in the position shown. The key to an electric motor is to then go one step further so that, at the moment that this half-turn of motion completes, the field of the electromagnet flips. The flip causes the electromagnet to complete another half-turn of motion. If the field of the electromagnet were flipped at precisely the right moment at the end of each half-turn of motion, the electric motor would spin freely.

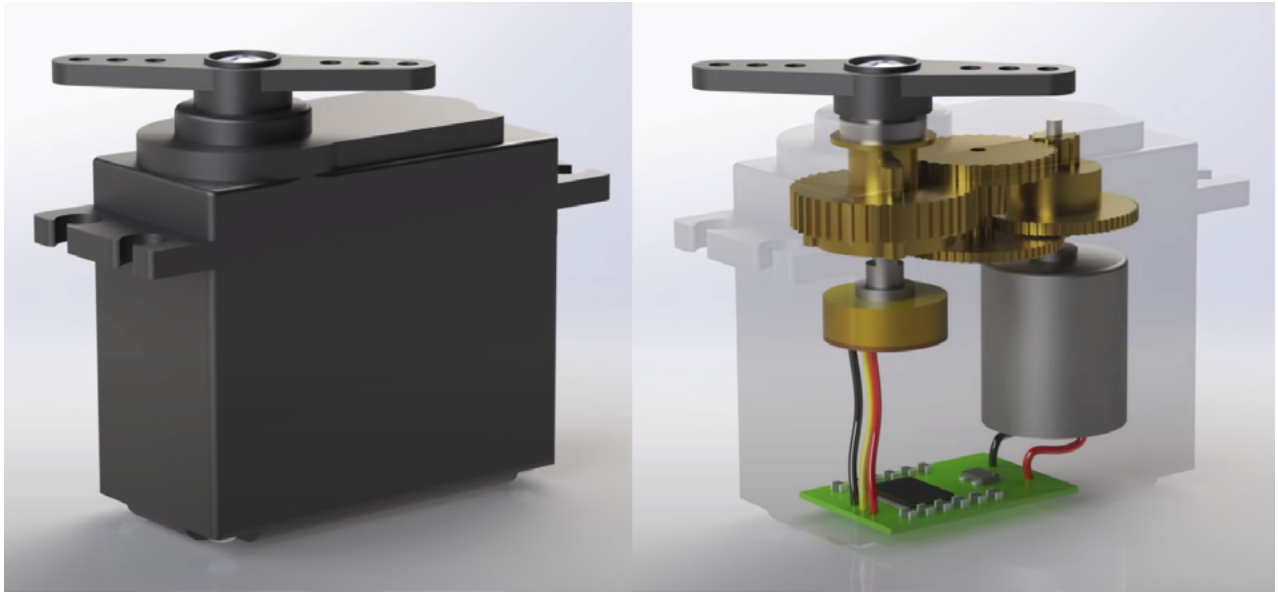
Grove Mini Fan in Education Add-on Pack

The Grove - Mini Fan module is a DC motor driver based on the AVR Atmega168 microcontroller. The module also provides a breakout through which you can change the microcontroller code. For example, the code can be changed so that the module can be used to drive a servomotor. By default, the module is set up to run the DC motor that is included in your mixer pack. The soft-leaved fan also included in the pack can be attached to the motor to make a fun project with kids. Being soft-leaved, the fan is completely safe and there is no chance of any injury even if it is moving at a high speed.

It looks like this after they are assembled.

What is inside of servo

A servo is very similar to electric motor - in fact it IS an electric motor with a controller chip, a potentiometer and gears for reducing speed all packed inside of plastic casing.



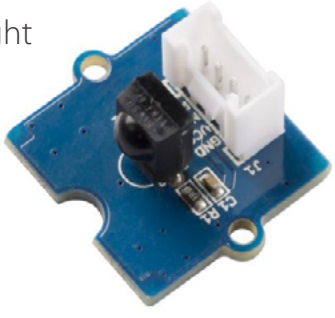
As we remember potentiometer can be used to measure the angle of rotation of the shaft - and using control circuit we can move servo shaft to precise degree, unlike with simple DC motor, where we can only control the direction and speed.

Grove Servo in Education Add-on Pack

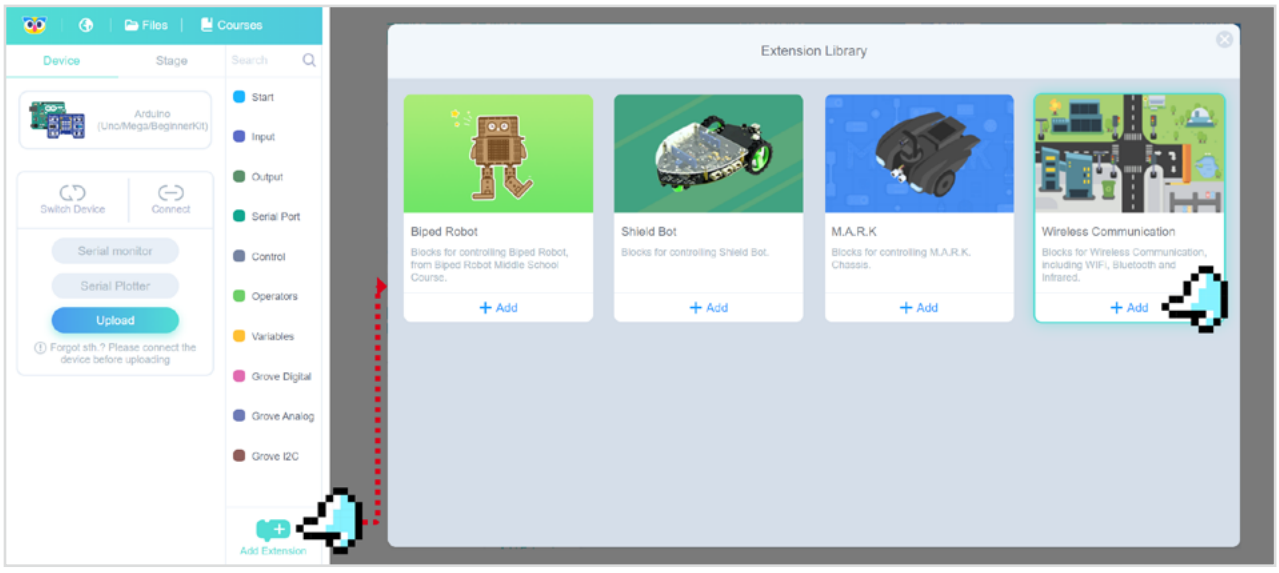
Grove - Servo is DC motor with gearing and feedback system. It is used in driving mechanism of robots. The module is a bonus product for Grove lovers. We regulated the three-wire servo into a Grove standard connector. You can plug and play it as a typical Grove module now, without jumper wires clutter.

Grove IR(Infrared) Receiver in Education Add-on Pack

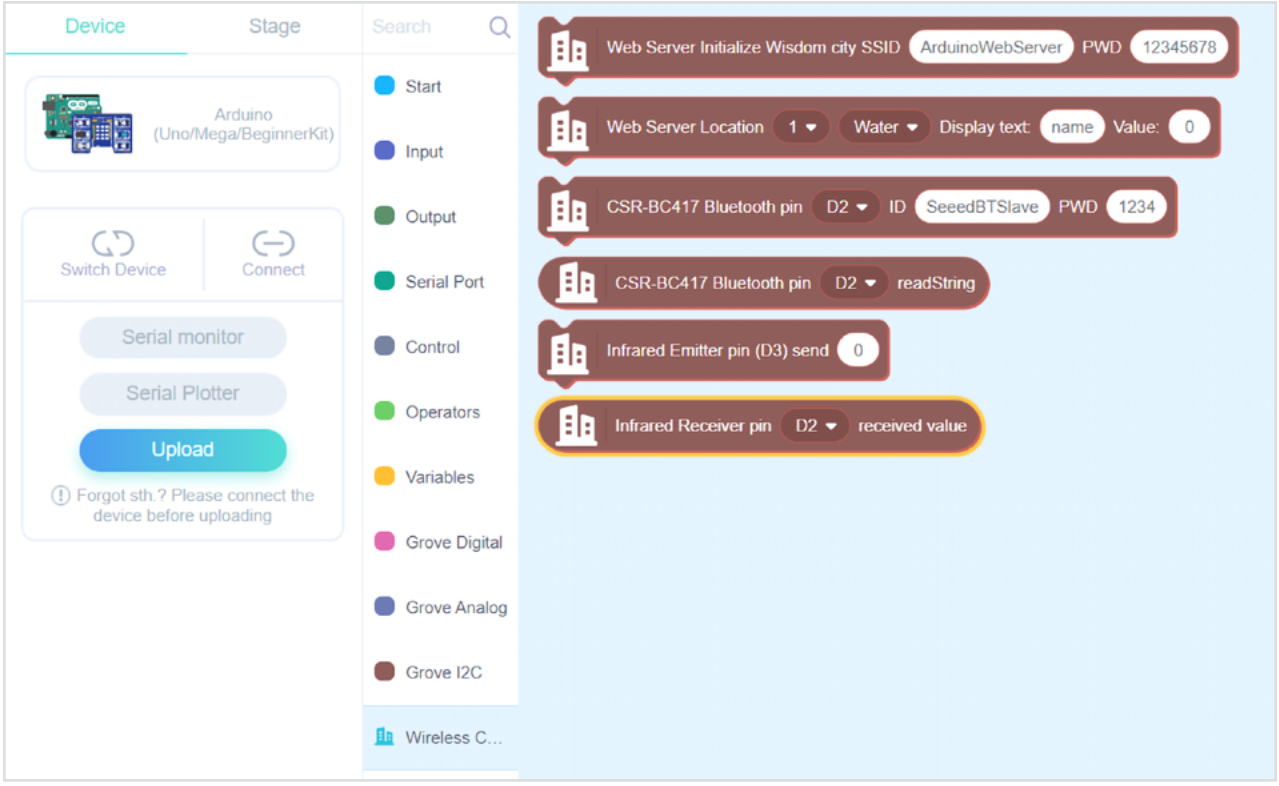
The Infrared Receiver is used to receive infrared signals and also used for remote control detection. There is an IR detector on the Infrared Receiver which is used to get the infrared light emitted by the Infrared Emitter. The IR detector have a demodulator inside that looks for modulated IR at 38 KHz. The Infrared Receiver can receive signals well within 10 meters. If more than 10 meters , the receiver may not get the signals.



To add building blocks for infrared receivers in Codecraft, you need to add the extension of "Wireless Communication" first, as shown in the figure below.



After adding, you can find the blocks related to the send and received of the infrared transmitter module in the block classification area of "Wireless Communication".



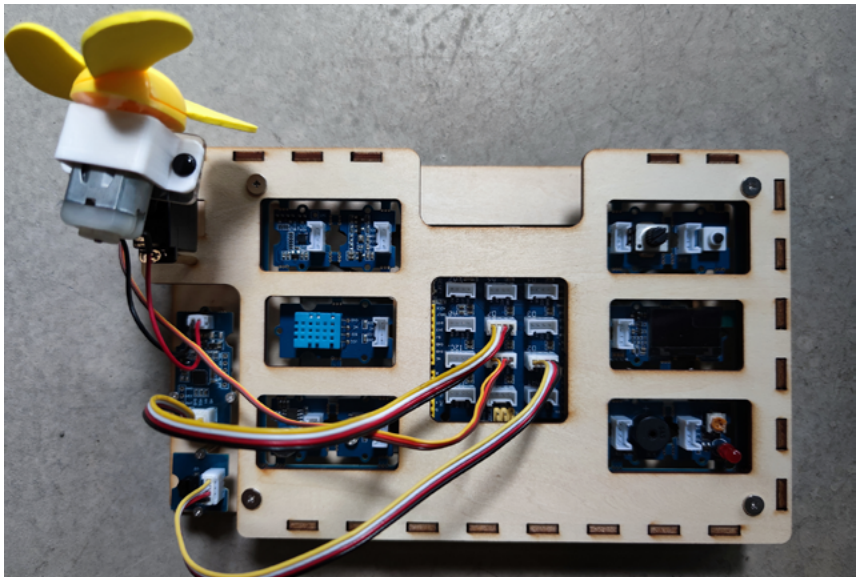
Comprehensive Project: Turning Fan

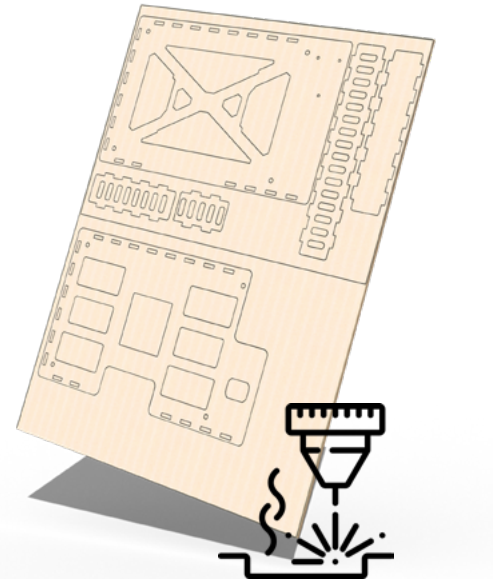
Assemble the structure

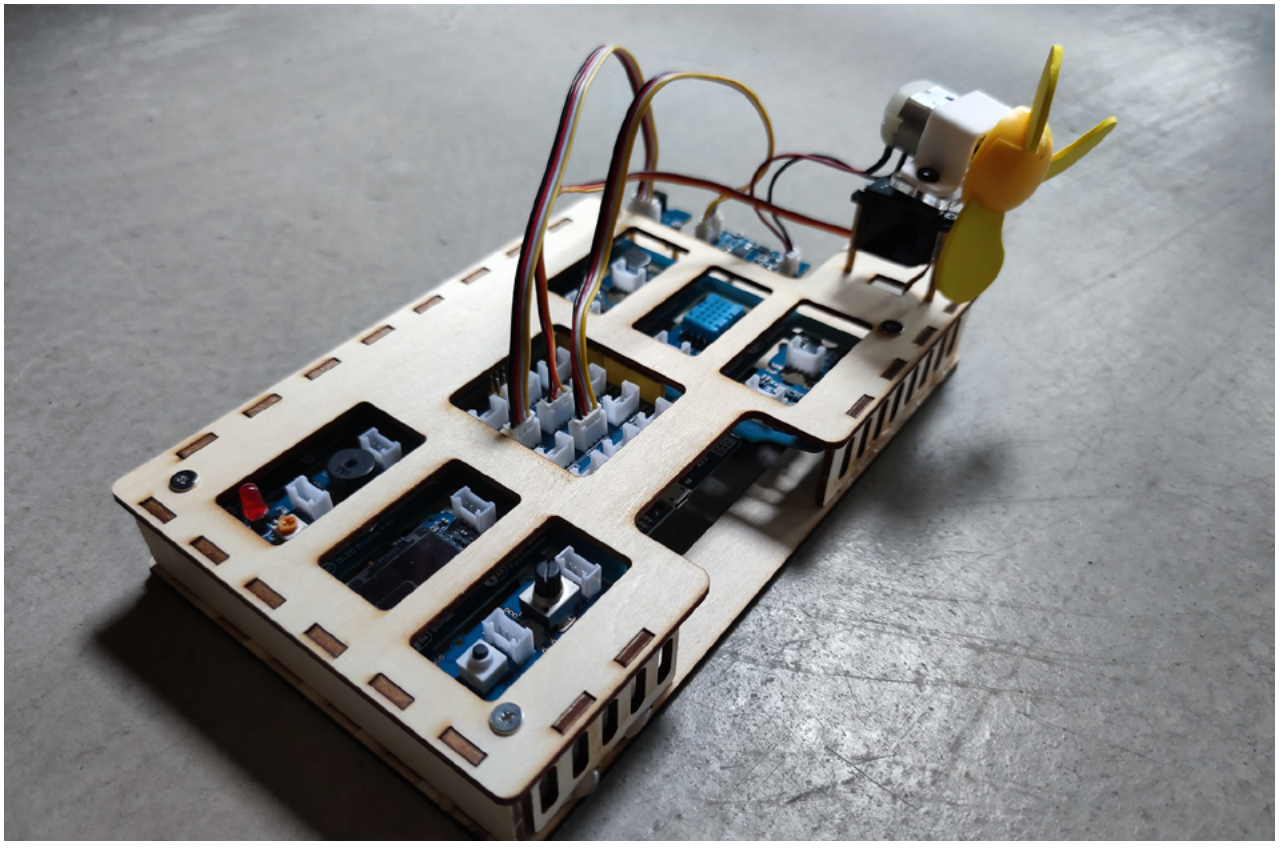
First of all assemble together the project structure from laser cut wooden parts, designs for which you can find here.

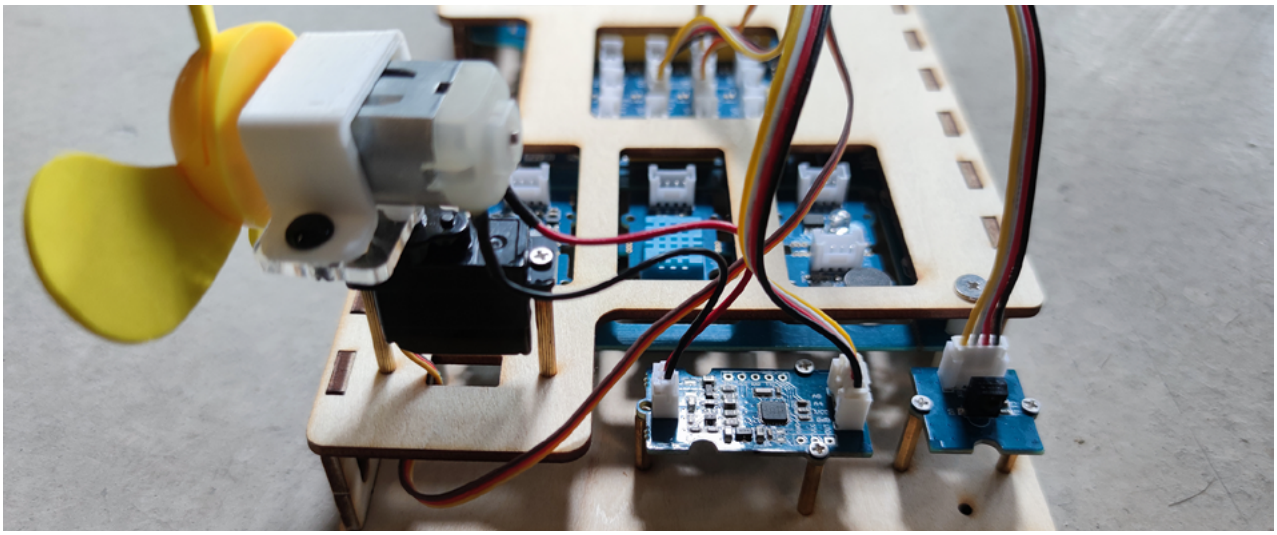
Grove Beginner Kit-L15-Turning fan.dxf

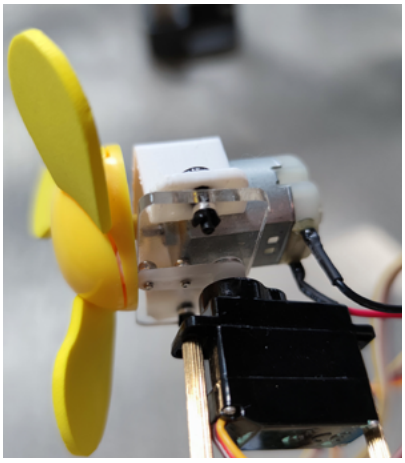
You can see the final assembly result in below pictures.





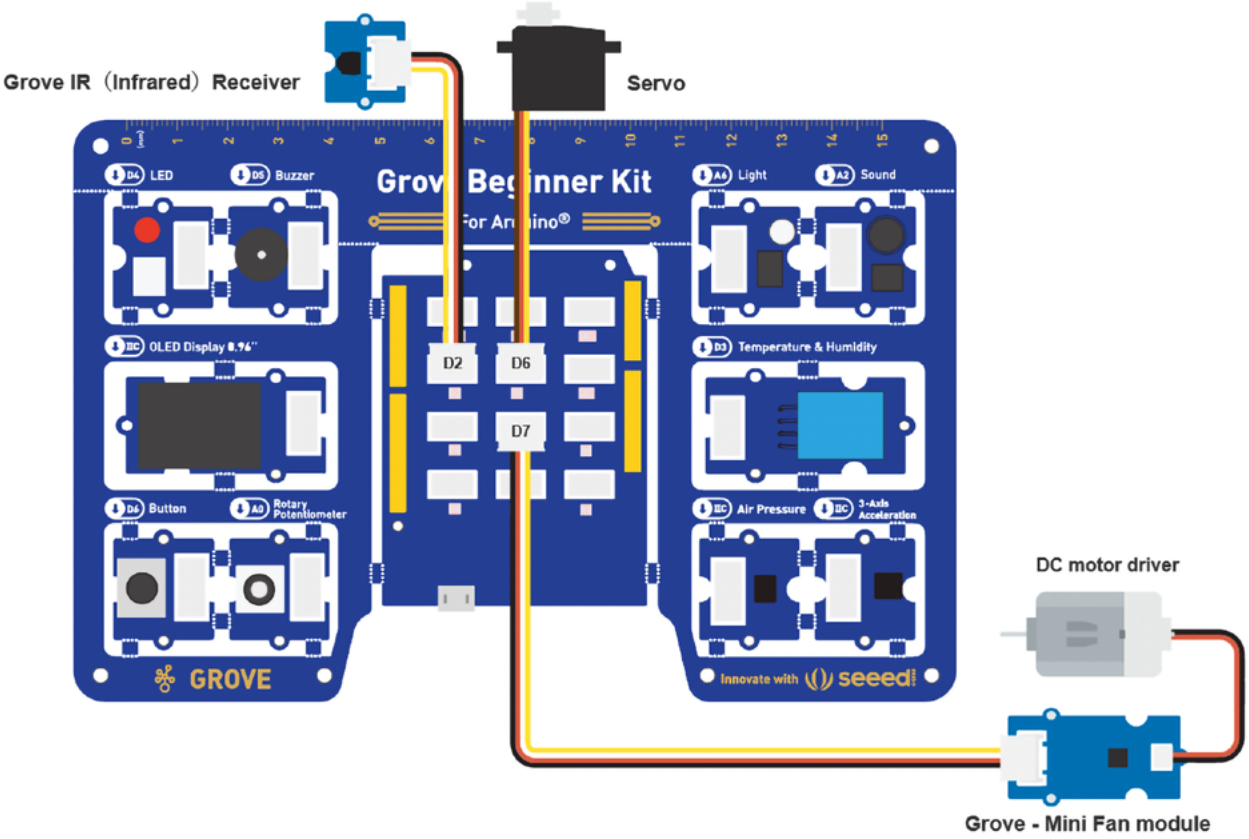






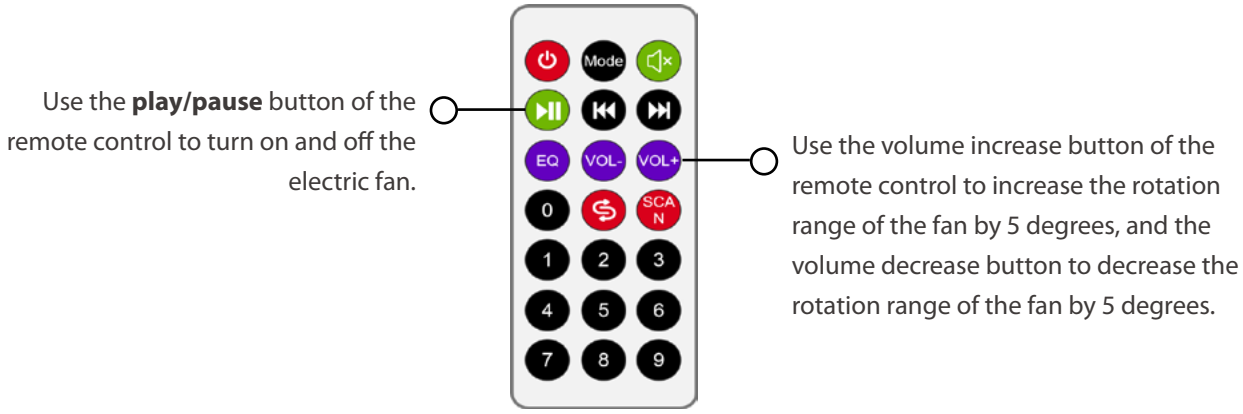
Pay attention to the connection between the servo and the motor. Use a small acrylic plate to fix the motor on the servo arm.

Then connect the additional modules with Grove wires to Grove Beginner Kit pins:
Servo to D6
Grove IR(Infrared) Receiver to D2
Grove Mini Fan driver to D7



Coding the Solution

The functions that need to be realized by programming are as follows:

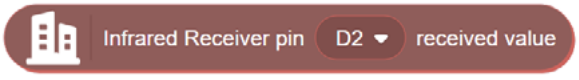


Step 1: Test the motor and understand working principle of IR receiver

Let's write a simple code now to test the motor:
That will switch the fan on and off with period of 1 second. After checking that Mini fan works and wired properly.

Step 2: Get the return value of the remote control button

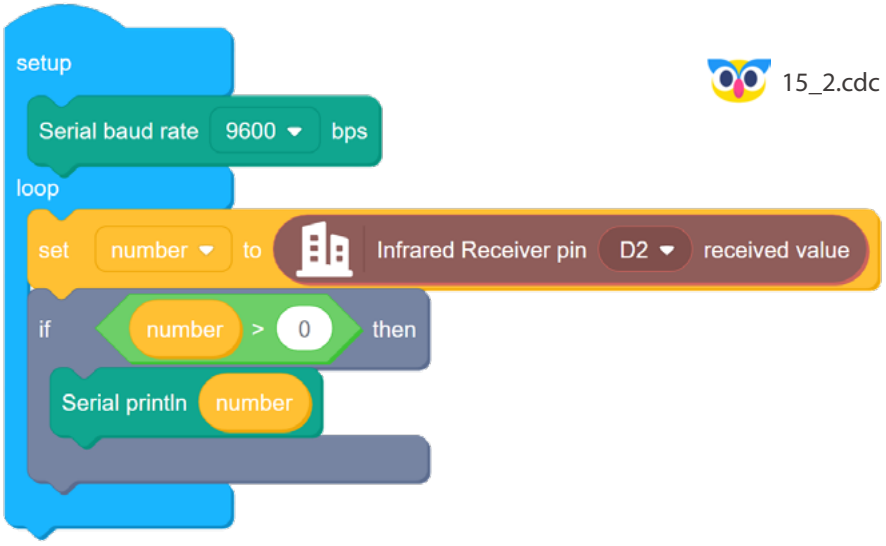
let's move on to the next stage of adding the remote control. We will need to Add Extension for Wireless communication, which contains the blocks for using Infrared Receiver.



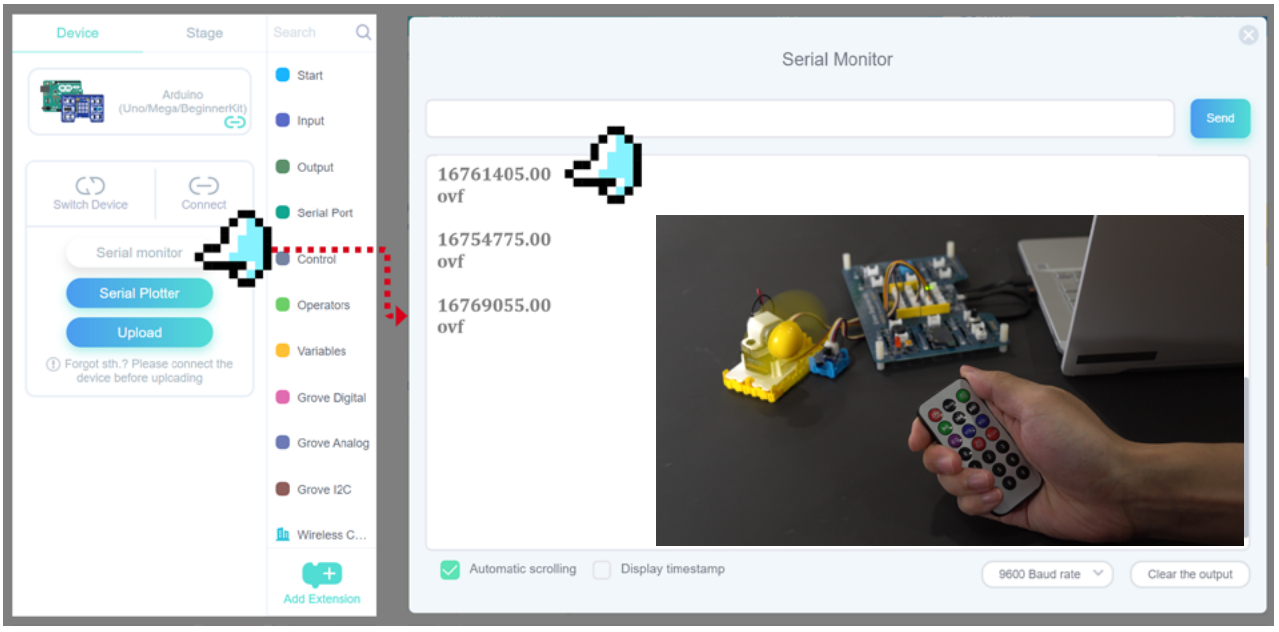
Our infrared receiver module comes bundles with a common IR remote control.
When we press a button on the remote, the IR LED in the remote emits a specific sequence of infrared pulses -we cannot see them with our naked eye, but if it was regular LED, we would see them as very fast flashes. Each sequence when it reaches IR receiver module can be decoded by the Grove Beginner Kit mainboard as sequence of numbers.

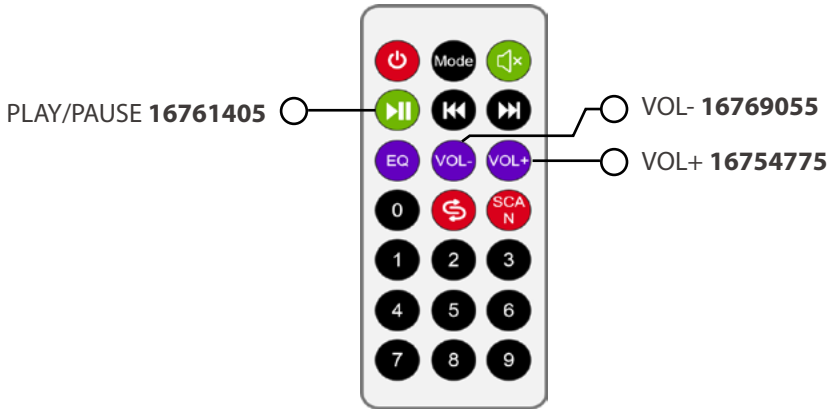
You can check these and values for other buttons by using this code.

Then connect to your board, open Serial monitor and press some buttons! Remember to note which button correspond to which number.

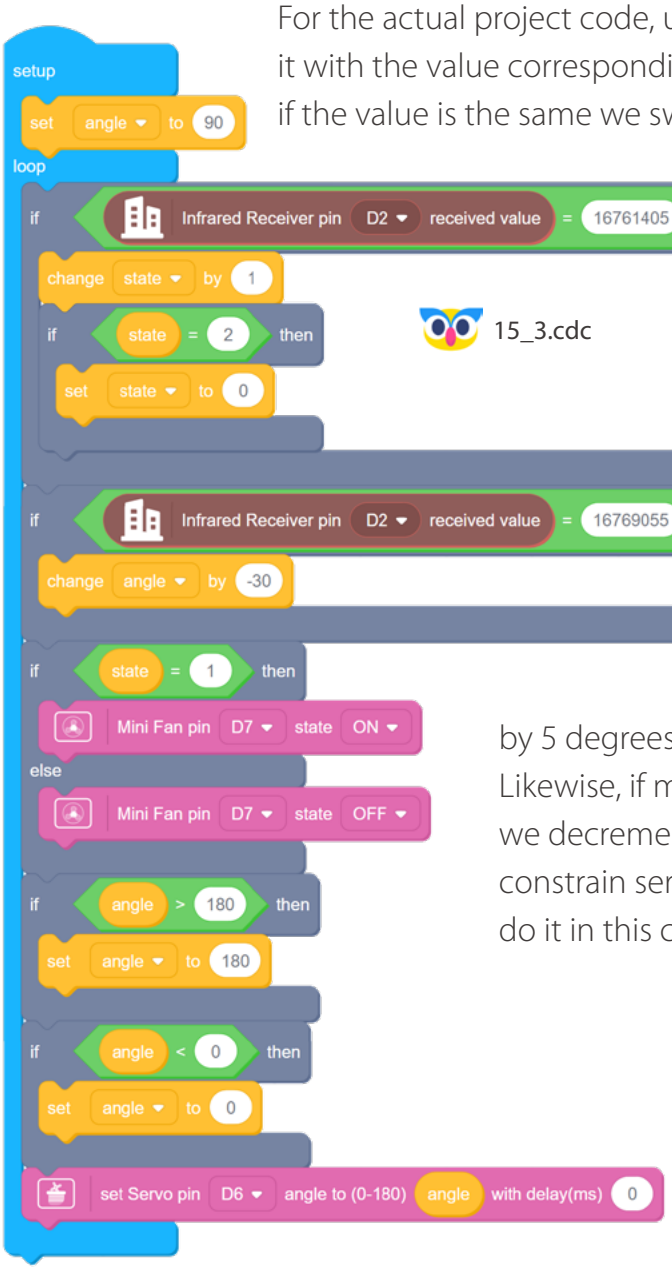


15_2.cdc





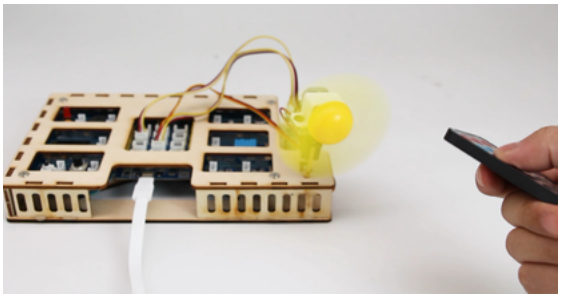
Step 3: Make a remote controlled turning fan



15_3.cdc

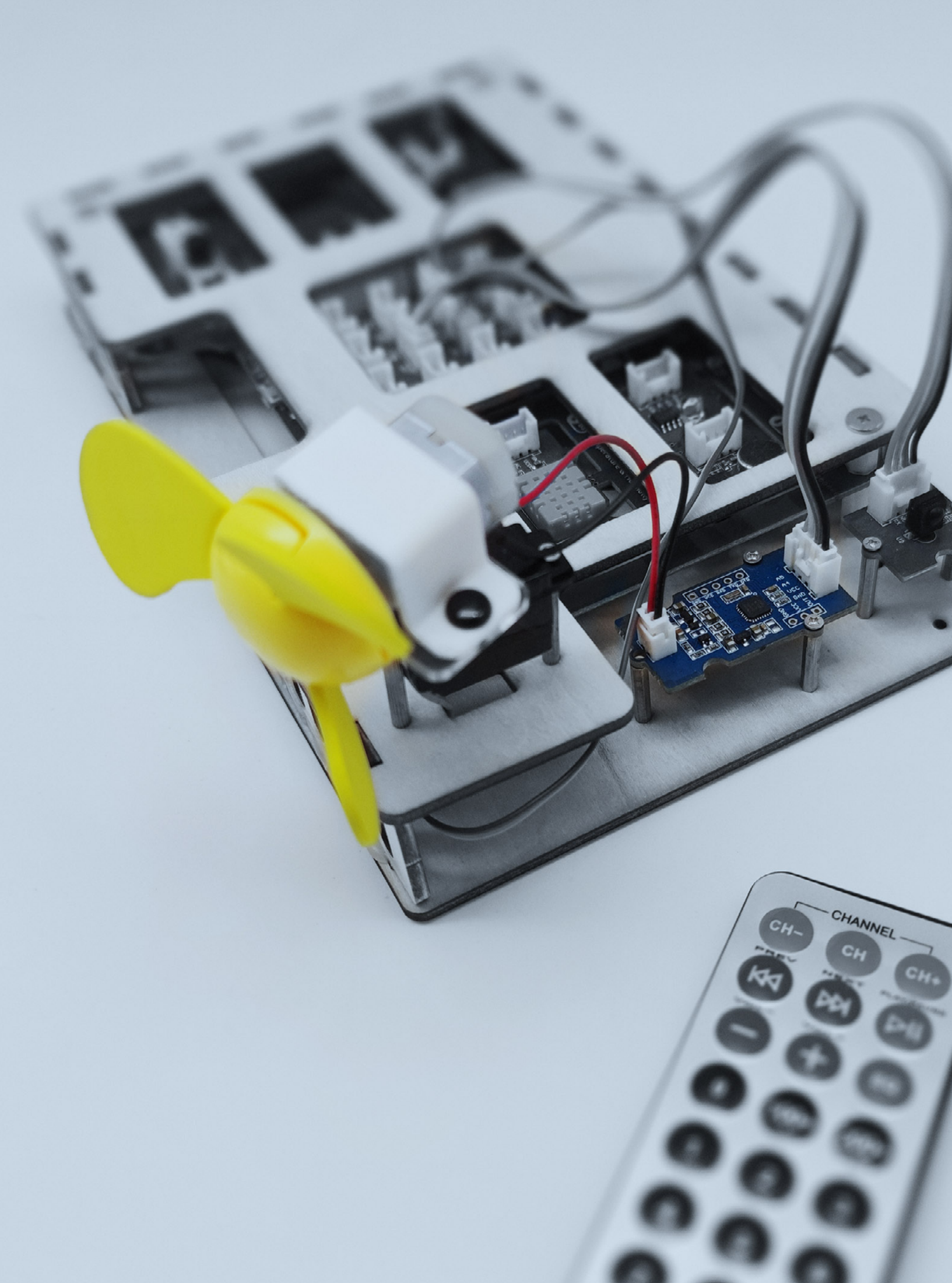
For the actual project code, upon receiving a value from we compare it with the value corresponding to the button PLAY/PAUSE - and if the value is the same we switch the fan on/off by changing the corresponding variable. For the final task let's add the servo in the mix and add the remote control of the fan angle!

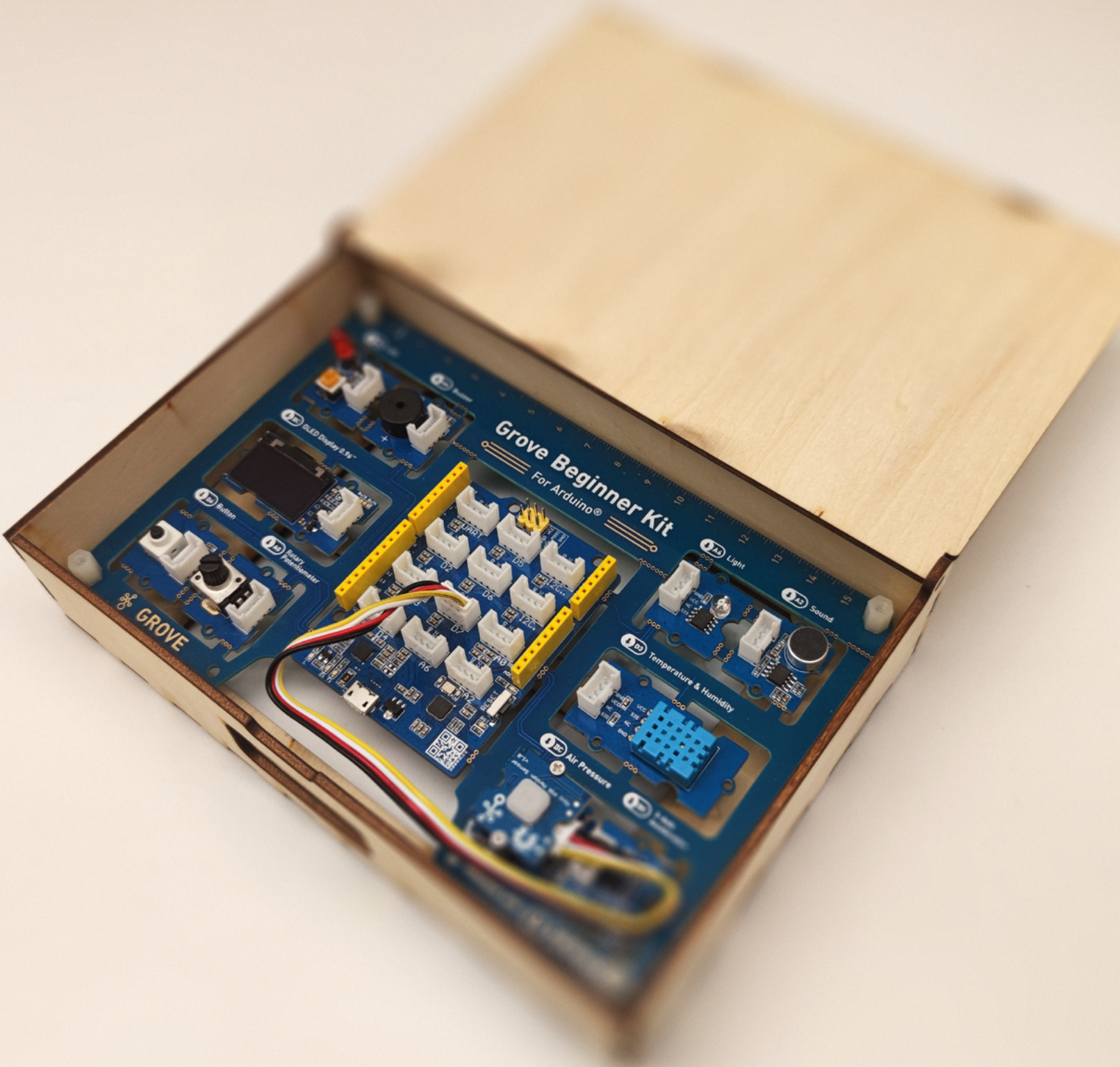
Here apart from comparing the value received from remote to corresponding to the button PLAY/PAUSE we additionally compare it to values of buttons + and -. If plus command is received from remote we increment servo angle variable by 5 degrees and move servo to the resulting degree. Likewise, if minus command is received from remote we decrement angle by 5 degrees. We also need to constrain servo angle variable between 0 and 180 - we do it in this code block.



⭐ Outside the Box

- Add the automation to the project! Use Temperature and Humidity sensor to trigger the fan on (and start rotating it) when temperature is above certain threshold.
- Display current state of the fan on OLED screen.
- Make the fan sound activated by using Sound sensor. When sound level is above threshold(for example the volume of hands clapping) trigger the rotating fan on/off.

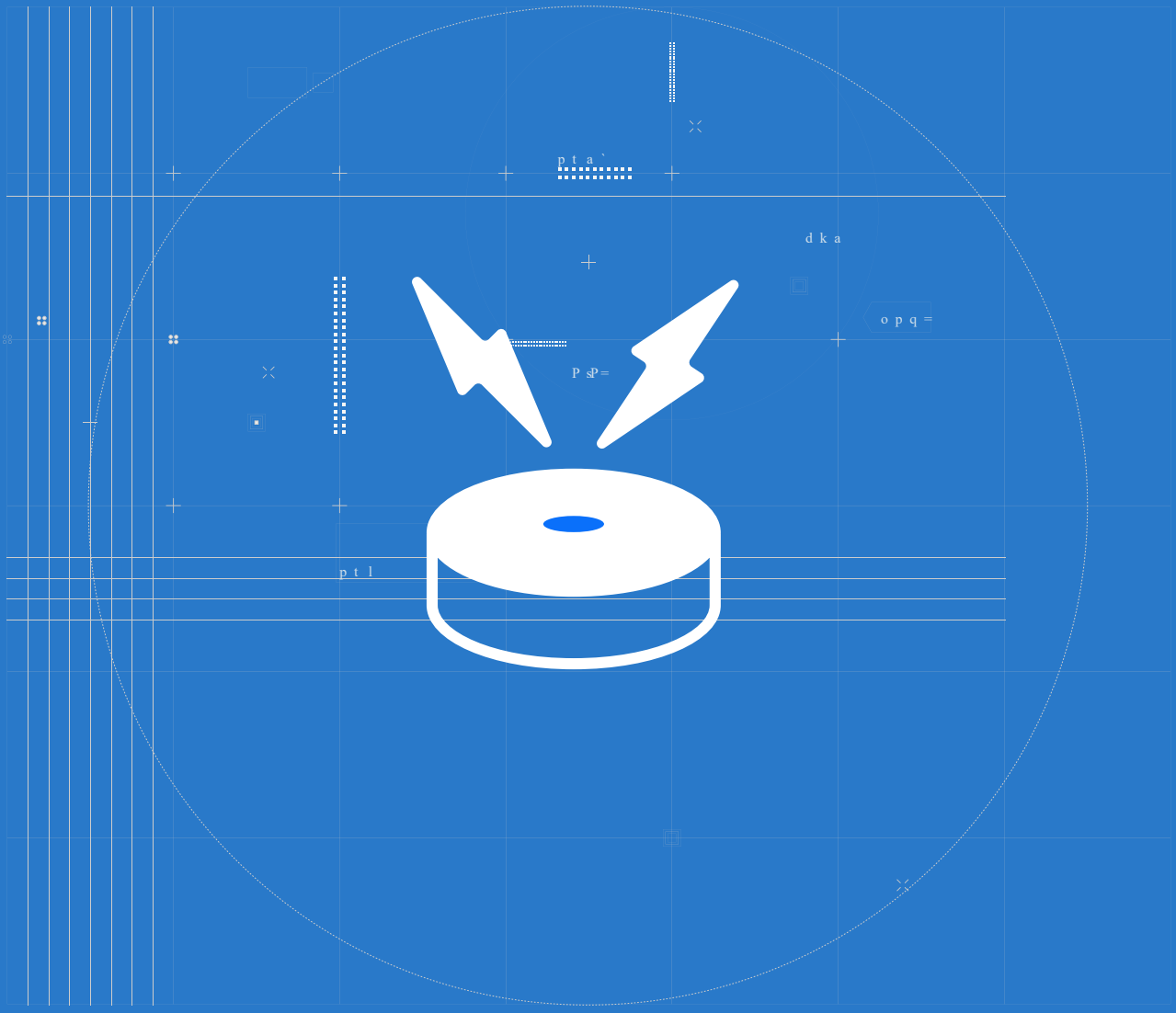

CHANNEL
CH-
CH
CH+


Grove Beginner Kit
For Arduino®
GROVE
Button
Rotary Potentiometer
Light
Sound
Temperature & Humidity
Air Pressure

Lesson 16

Burglar Alarm

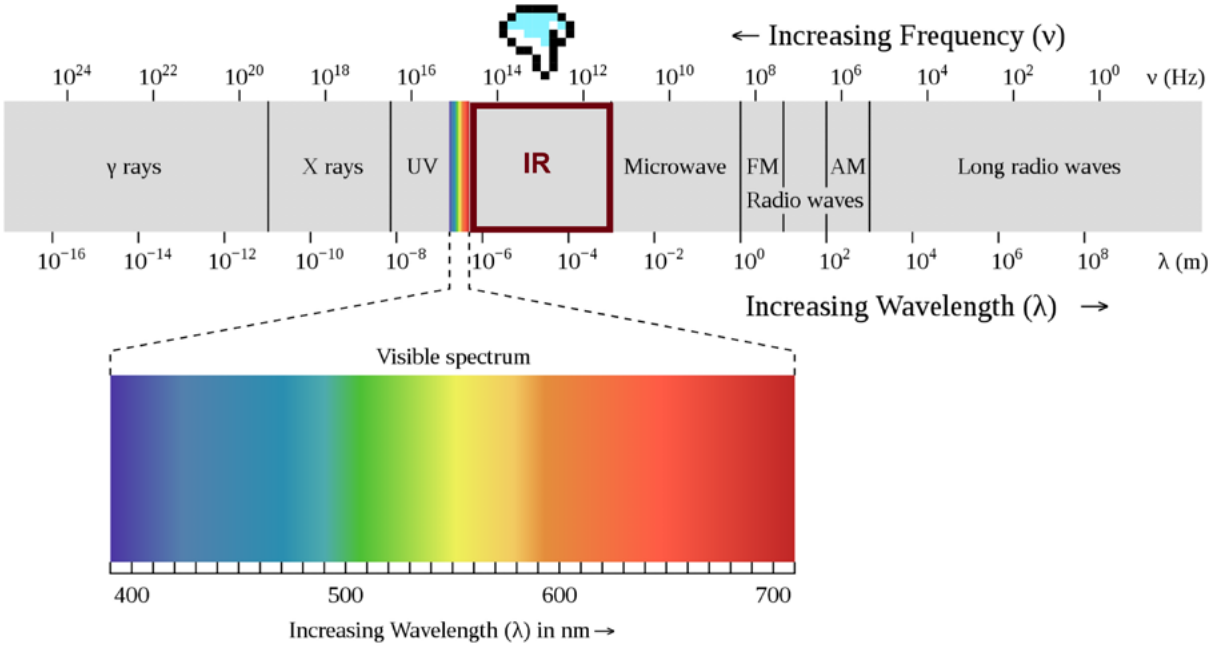
For the last project in our Grove Beginner Kit course we will make a simple, but genuinely useful device that will help to keep your stash of cookies (or other valuable things) well guarded. We will use Passive Infrared Motion Sensor module to detect the movement and sound the alarm when it is detected. And to make sure that the alarm is only triggered when necessary we will place that module inside the wooden box were you can put your valuables.



The Big Picture

What is Infrared Light or Infrared Radiation?

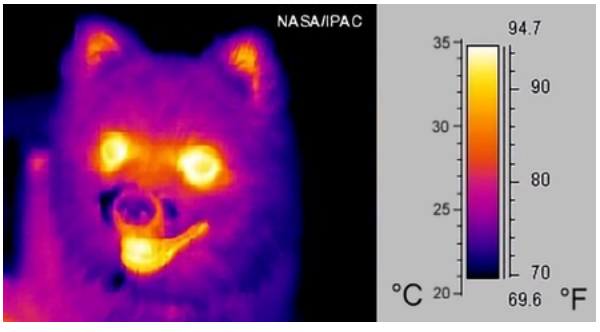
In earlier lessons we have already familiarized ourselves with concept of visible light - that it is a form of electromagnetic radiation that our eyes can perceive.



Inferred spectrum waves are longer than light which humans can see and shorter than microwaves. The word infrared means below red. It comes from the Latin word infra (meaning below) and the English word red. Despite you cannot see infrared radiation you can feel it - as heat. The hotter something is the more infrared waves it emits.

This for example is the image of a dog in infrared.

You can see that the parts that are hotter, such as mouth and eyes are brighter and nose (which is normally cold in healthy dogs) is darker. Mammals regulate their body temperature and keep it at constant level.



Grove mini PIR Motion Sensor in Education Add-on Pack

In this project we will only use one extra module - Passive Infrared Motion Sensor, sometimes abbreviated as mini PIR Motion Sensor.

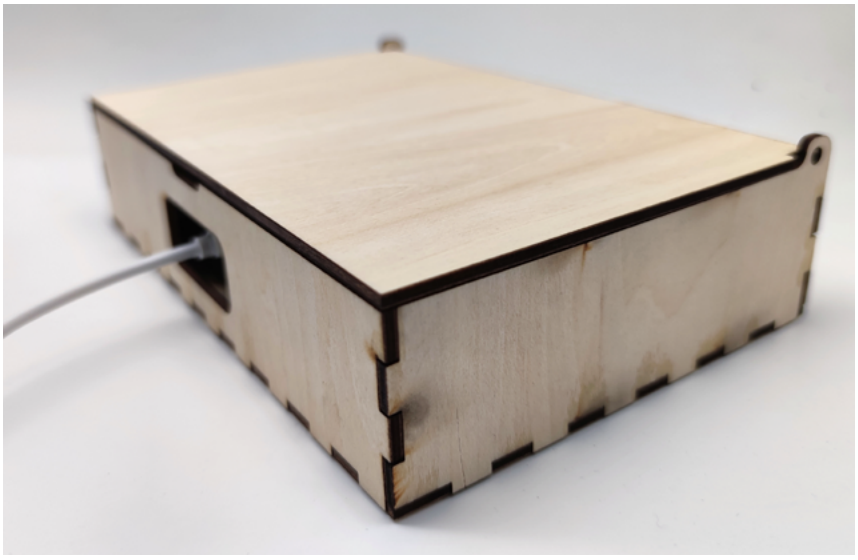
Comprehensive Project: Burglar Alarm

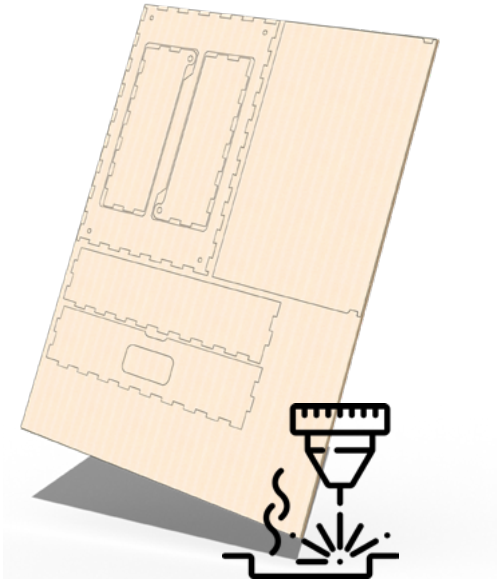
Assemble the structure

First of all assemble together the project structure from laser cut wooden parts, designs for which you can find here.

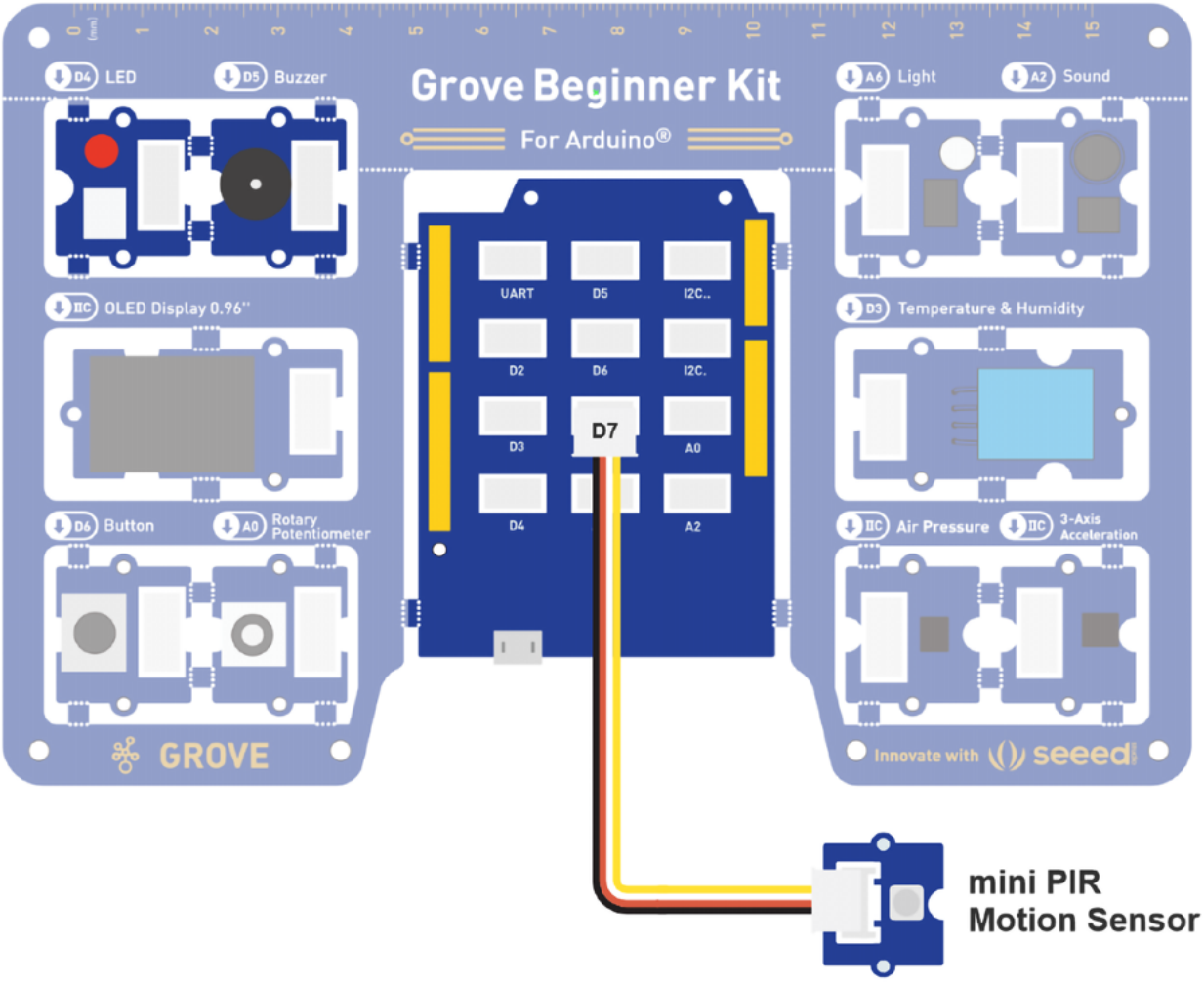
Grove Beginner Kit-L16-Burglar alarm.dxf

You can see the final assembly result in below pictures.





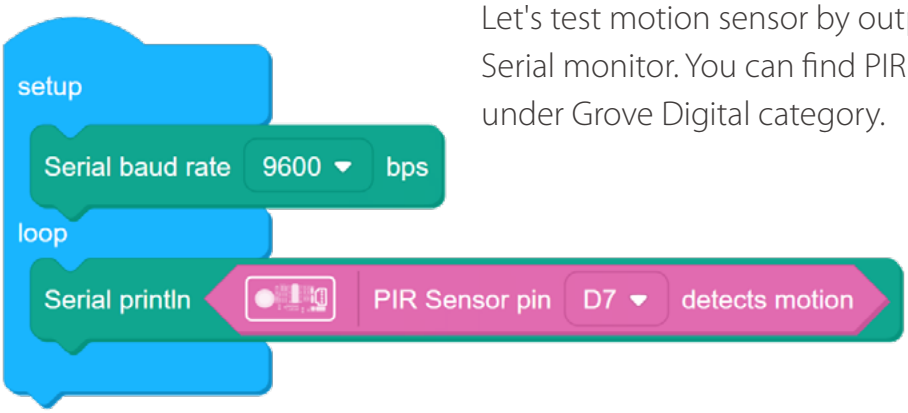
Then connect the additional module with Grove wires to Grove Beginner Kit pins:
mini PIR Motion Sensor - Pin D7



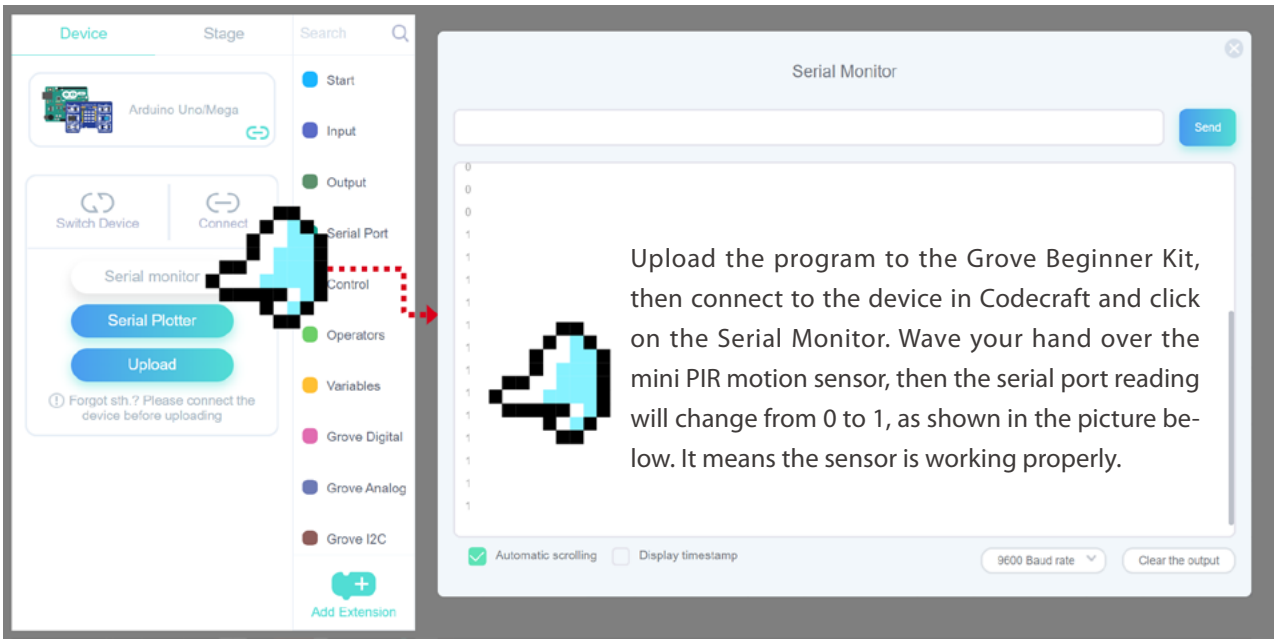
Coding the Solution

Step 1: Test the sensor with Serial Monitor

Let's test motion sensor by outputting it's values to Serial monitor. You can find PIR Motion Sensor block under Grove Digital category.

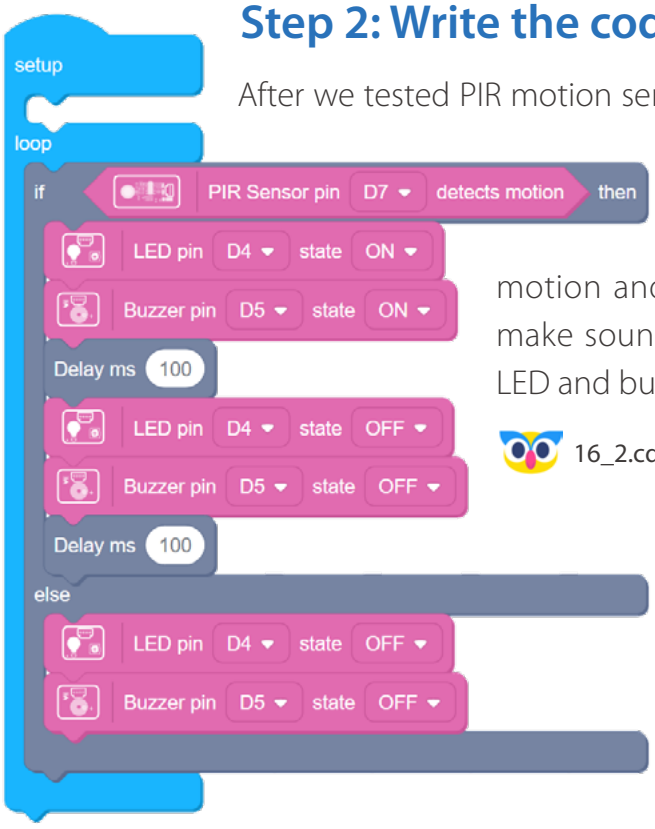


16_1.cdc



Upload the program to the Grove Beginner Kit, then connect to the device in Codecraft and click on the Serial Monitor. Wave your hand over the mini PIR motion sensor, then the serial port reading will change from 0 to 1, as shown in the picture below. It means the sensor is working properly.

Step 2: Write the code for Burglar Alarm



After we tested PIR motion sensor and made sure it is working, let's make a code that will alert us about an intruder, trying to break in the safe box. For that we will check if PIR motion sensor detects motion and if it does we will start blinking LED and make sounds with buzzer. Otherwise we turn off the LED and buzzer.

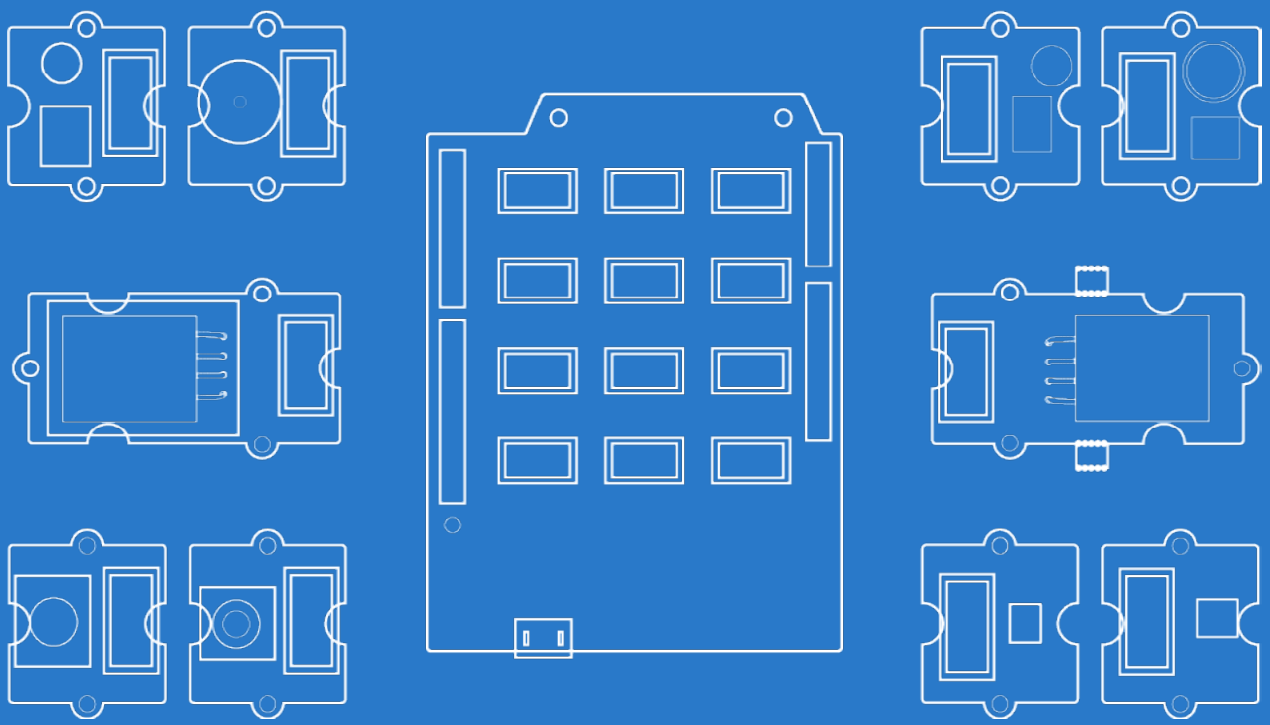
16_2.cdc

⭐ Outside the Box

- Use a button to switch the alarm on/off.
- Add a delay after triggering PIR motion sensor during which a user can turn off the alarm with a button press.

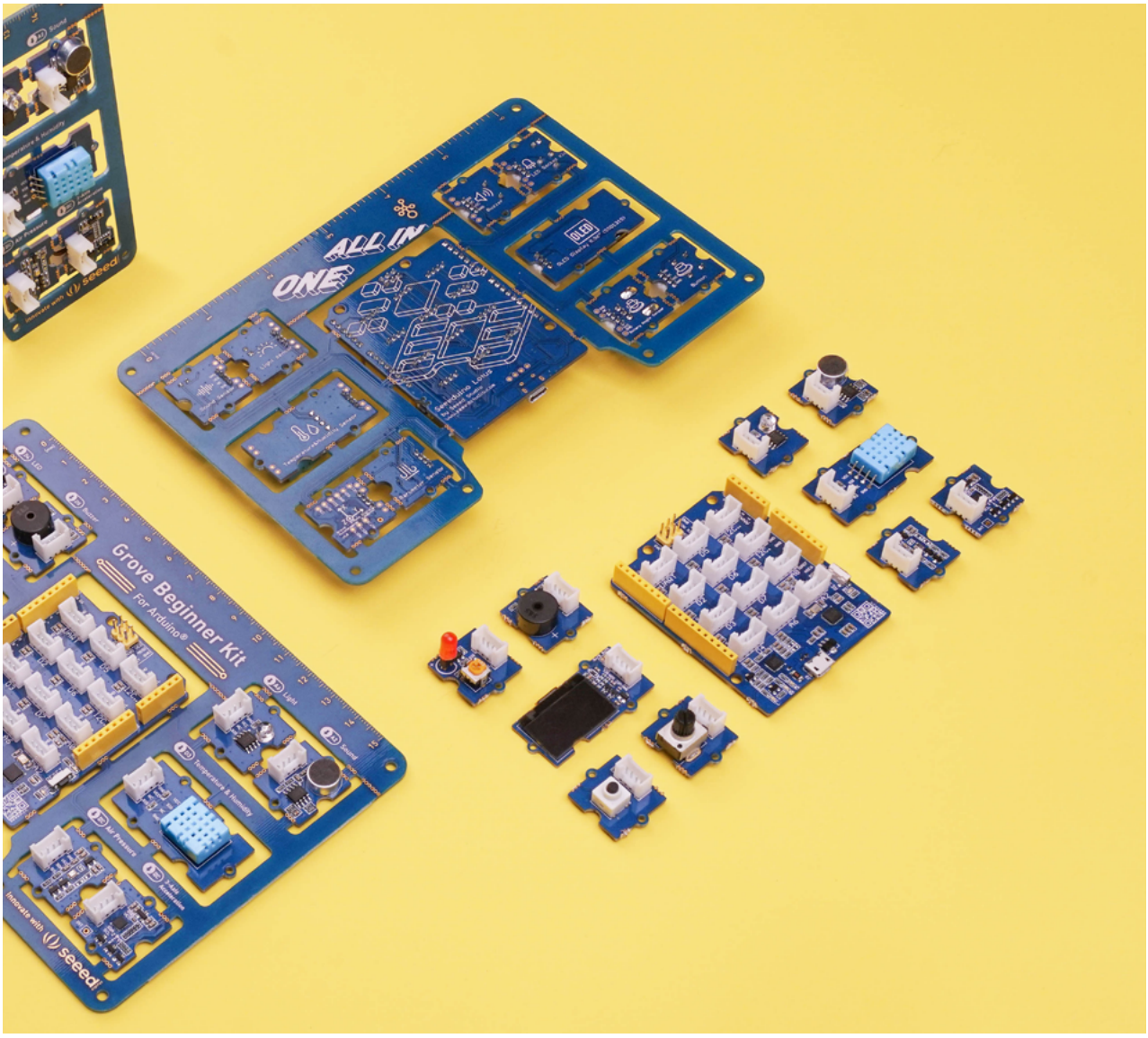
Afterword

Resources, downloads, links and instructions related to this lesson.



Breakout Instruction

If you have completed the course, you can also split the Grove Beginner Kit modules from the main board for creating your own projects.



Note

Please be careful not to cut your hands when using a knife.

If you prefer to use the modules in elsewhere then you can simply follow the procedures to break the modules out!

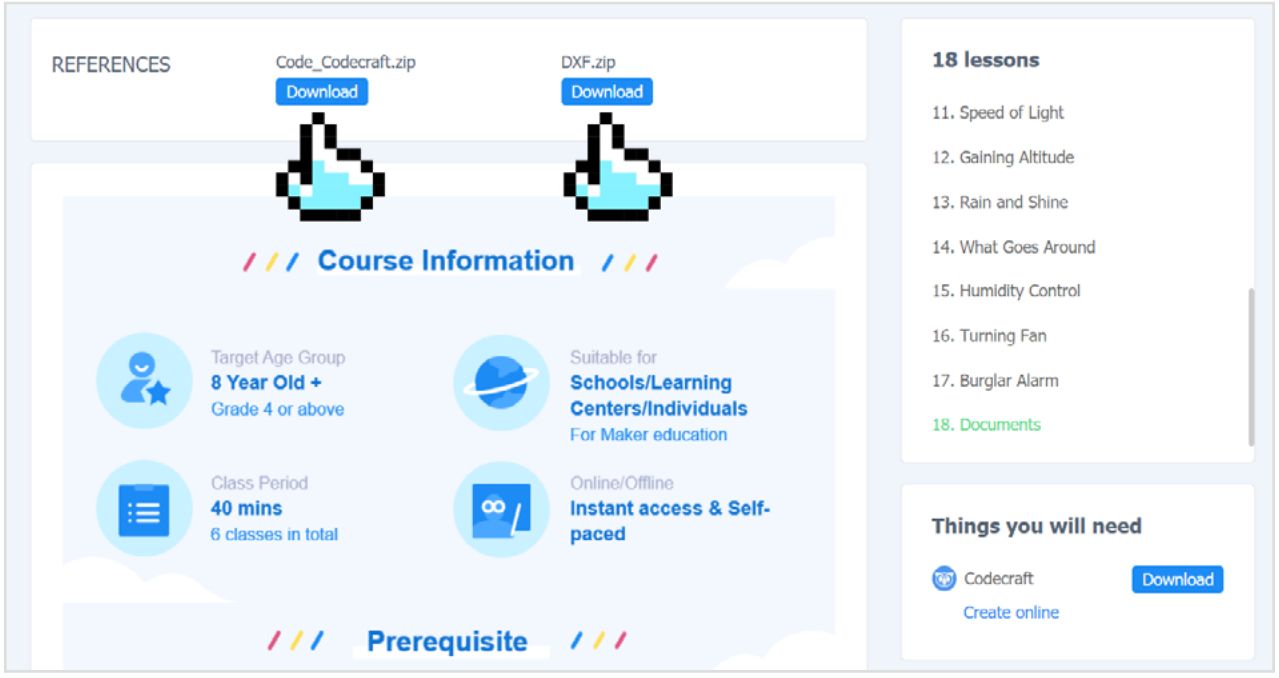
Step 1 : Use a knife or a sharp object to cut at the stamp holes that connect the sensors together.

Step 2 : Shake the modules up and down and it should come out quite easily!

Online resources

Visit the following URL to get the online version of this course.
https://make2learn.tinkergen.com/course/?sku=604182009

You can download the code samples and the .DXF files required for Lessons 14, 15, and 16 in Documents section.



Codecraft

https://www.tinkergen.com/codecraft
Programming Online: https://ide.tinkergen.com/
Download: https://ide.tinkergen.com/download/en/
Codecraft Documentation: https://www.yuque.com/tinkergen-help-en/codecraft

Grove Beginner Kit For Arduino for Seeed Wiki

https://wiki.seeedstudio.com/Grove-Beginner-Kit-For-Arduino/

Purchase link

Grove Beginner Kit for Arduino
https://www.seeedstudio.com/Grove-Beginner-Kit-for-Arduino-p-4549.html
https://shop.tinkergen.com/grove-beginner-kit.html
Grove Beginner Kit for Arduino Education Add-on Pack
Coming soon...

Support for schools or training institutions

This course also can be used for teaching in public schools or training centers. If you want to use this course in a public school or training center, you can contact us to obtain:

- Teacher discount for product purchase
- Teacher manual PPT files
- Technical support for course development

Call for multilingual translation volunteers!

If you wish to translate course content into a language version other than English or Chinese, you can also contact us for support.

CONTACTS

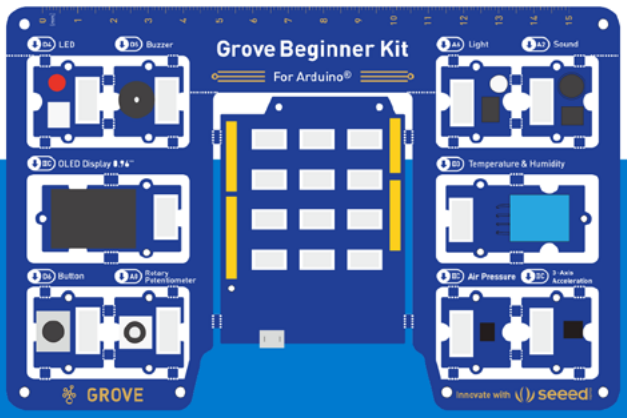
Tel: +86-0755-86716703
Address: 1002, G3 Building, TCL International E City, 1001 Zhongshan Park Road, Nanshan District, Shenzhen
General: contact@chaihuo.org
www.tinkergen.com
www.seeedstudio.com

Course Developers

This course was co-authored by the following TinkerGen employees:
Dmitry Maslov, Feng Lei, Meng Yihui, Liu Haixu, Wang Qun, Tang Ruiqian
SeeedStudio employees:
Yang Jia Mou, Liang Jiawei

TinkerGen

STEM made simple



TinkerGen
STEM made simple | seeed studio

To know more about TinkerGen

Please contact us:

contact@chaihuo.org

86-0755-86716703

www.tinkergen.com