

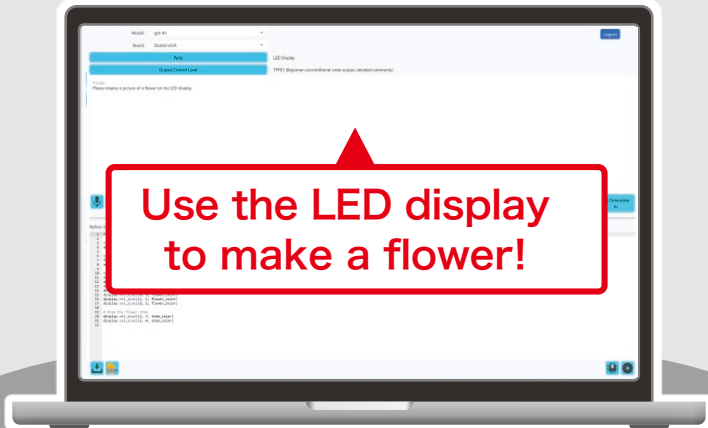
Learn along with **AI!**

Artec Intelligence The AI Basics

Facilitator
Edition

**Artec Intelligence is based on the ideas of
Kazuo Tenra**

Former instructor at Hibiya High School in Tokyo and project
professor at Tokyo Gakugei University

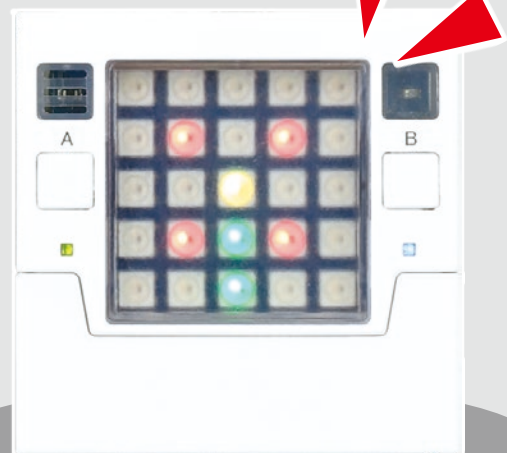


Use the LED display
to make a flower!



For Facilitators

This textbook contains six chapters. Each chapter is expected to take 40-45 minutes, but feel free to adjust your pace depending on the needs of the class.



[Chapter 1] So What's AI?

◆ 1-1. AI in Our World

AI is a word you hear often these days, with each new AI-powered product or service making hard to imagine how we got along without it!

Example AI Services and How They Work



Voice Assistants



Smartphone Security
(face and fingerprint ID)



Self-Driving Cars



Media and Product Recommendations



Auto-Translate

◆ 1-2. The History of AI

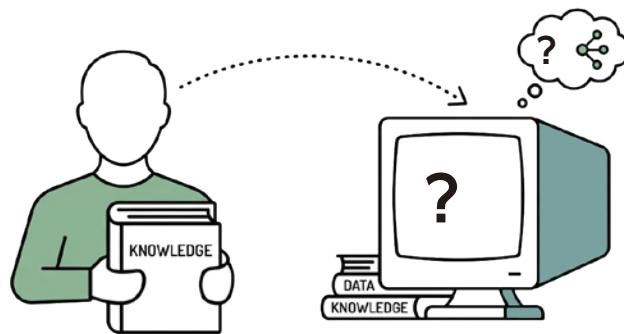
AI stands for Artificial Intelligence, and developers who work on it aim to make an intelligence that can think and learn just like a real person. Developments in the field of AI research can be broken up into three booms starting in the 1950s.

● The First Boom (1950s~1960s)

This period focused heavily on teaching rules which allowed them to infer based on and search through data. While these models are great for solving puzzles or playing games with specific rules like mazes or chess, they have limits when it comes to working with the complex, ambiguous problems we encounter in the modern world!

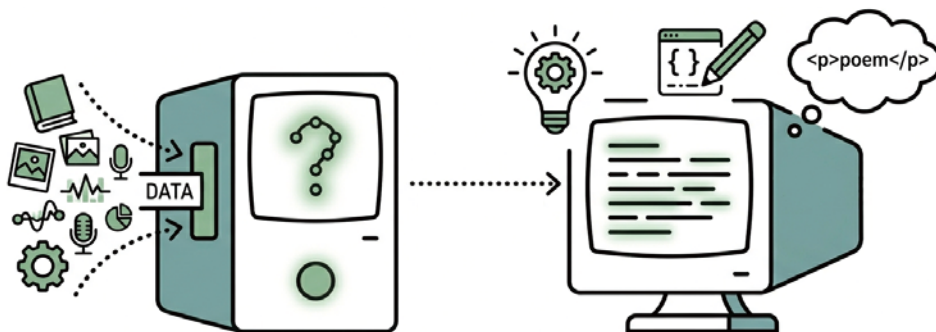
● The Second Boom (1980s)

This era focused on making Expert Systems by feeding computers lots of rules based on knowledge from specialists like doctors and lawyers. One major issue with this is that it would take quite a bit of time to develop rules for every kind of knowledge in today's world



● The Third Boom (2000s~)

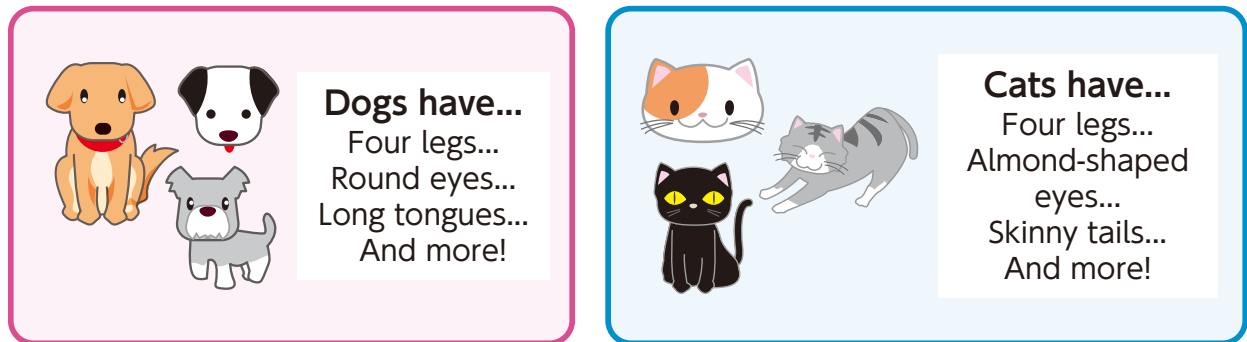
This wave of AI focused on **machine learning**, which lets a computer train itself on data instead of humans teaching it rules. You'll learn more about this on page 4. The introduction of **deep learning** models which mimic the human brain led to a vast increase in an AI models ability to recognize patterns in complex data as well as distinguish between pictures and audio. You'll learn more about this on page 6! This technology has continued to evolve, leading to generative AI which can create sentences, images, and even programs based on accumulated data. You'll learn more about this on page 7!



◆ 1-3. Secrets of Machine Learning

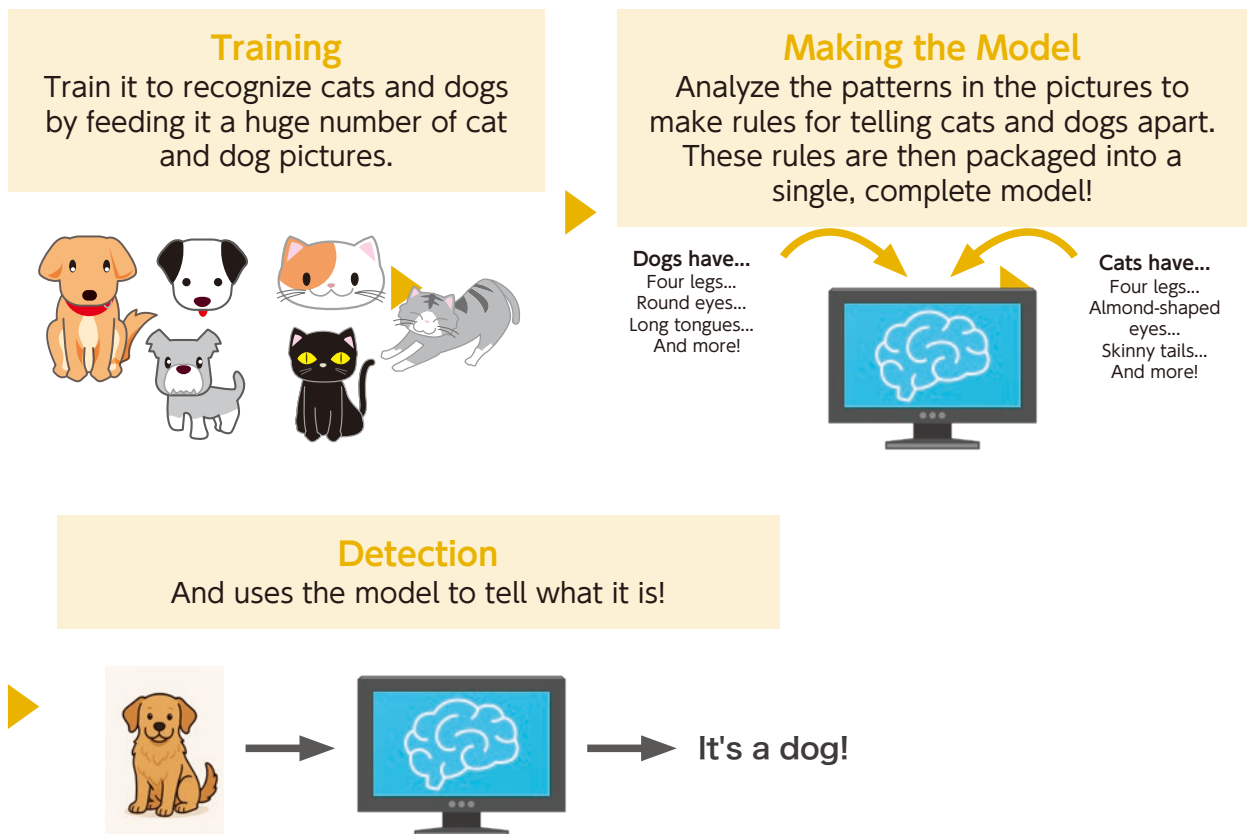
The word artificial intelligence covers lots of technologies, and one way we use this intelligence is in machine learning. In machine learning, a computer teaches itself to find rules and patterns in a large data set so it can make decisions by itself.

A good comparison is how we teach ourselves to recognize and tell apart the dogs and cats we see every day in the real world!



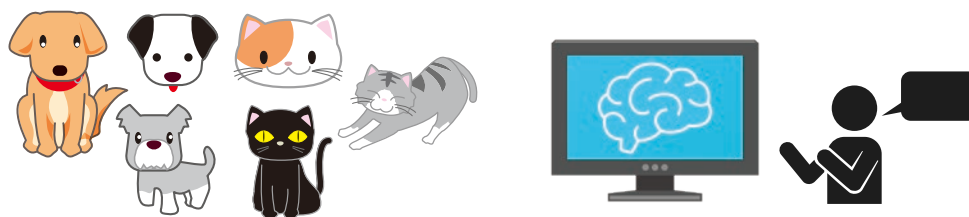
Just like a human, giving a computer lots of pictures of dogs and cats allows it to train itself to recognize the differences between them!

Example Recognizing Pictures with AI



● The Three Styles of Machine Learning

You can divide the types of machine learning into three major styles based on what kind of data you give the AI and how much you help it!



① Supervised Learning

This is the most common type of learning method where a human gives the AI a set of data with labels which tell it the correct answers!

How it Works	Start by giving the AI a picture of a dog along with a label that says, "This is a dog!" Adjusting the instructions by giving it more concrete d attention to the ears," as
The Pros	Supervised learning lets the model guess and pick a dog correctly with 90% certainty!

② Unsupervised Learning

With this style, humans pass the AI a huge amount of data without telling it what the correct answer is!

How it Works	Give the AI the data and it will train itself. Instead of giving the AI answers to find, it's given an analysis framework of grouping similar items together.
The Pros	Unsupervised learning can lead to the AI discovering data structures and groups (known as clusters) that a human wouldn't notice themselves!

③ Reinforcement Learning

This style relies on giving the AI points as a reward to train it instead of giving it the answer directly.

How it Works	It starts by telling the AI that it has to do its last high score by 100 points. The AI then uses trial and error to train itself based on this framework!
The Pros	Reinforcement learning lets the AI train itself for maximum performance by using a set of predetermined rules!

◆ 1-4. Deep Learning

Until recent years, deep learning relied on having humans carefully pick answers and things to pay attention to in order to train the AI as efficiently as possible. Deep learning, on the other hand, allows the AI to automatically learn from a huge amount of data by using the rules given to it by humans and by finding detailed patterns that humans wouldn't be able to find themselves!

Machine Learning	Deep Learning
A human picks and adjusts the data...	The AI trains itself!
	

● The Systems Behind Deep Learning

Deep learning use models known as artificial neural networks built to mimic the neural circuits of the human brain. The key to this system is processing information by stacking it into multiple “layers.”

[Input Layer] This organize data by its surface-level features like points and lines.

[Hidden Layers] The AI combines these to understand complex shapes and patterns.

Creating a deep stack of these layers allows an AI to spot small differences, complex rules, and other relationships that a human can't explain in words!

Example

Image Recognition

These models can distinguish the objects in a photo with higher precision than a human!

Audio and Translation

Accurately listening to human speech lets the model create natural sentences in any language!

Generative AI is the result of the developments which let deep learning models gain a deeper understanding of a data set's characteristics!

◆ 1-5. What's Generative AI?

Generative AI will use the vast amount of data it was trained on to generate new content based on instructions from a human being. While AI up to this point specialized in sorting the data given to it, generative AI can create text, images, audio, and video which no one has ever seen before!



Text



Images



Audio



Video

● Two Types of GenAI

Generative AI comes in two major types depending on how they handle data and how we use them!

Single Modal

Single modal generative AI focuses on creating one type of data at a time like text or images. Being made for a specific field means fewer wasteful calculations and an incredible level of precision and speed!

Examples Translation and image generation

Multimodal

Multimodal models can handle multiple kinds of data at once, including text, images, audio, and more. These models are great for when you want to process multiple pieces of data at once, like using text to ask the name of an object in a photo!

Examples General purpose AI models like ChatGPT and Gemini which can chat with you and analyze images

The secret to the intelligence of multimodal AI is its ability to process a vast number of words. What gives these models the ability to understand pictures or converse naturally with humans is because they can process words along with any other data. Next, let's take a closer look at the **large language models** which take advantage of all this intelligence!

◆ 1-6. Large Language Models

Large language models are programs which are trained on huge amounts of data from the Internet to talk with humans in a natural way. This data includes millions of books as well as articles and social media posts, allowing them to record not just the meanings of words but tone and the knowledge inside of the data, too!



The reason we can talk to generative AI just like we would a real human being is because it uses large language models!



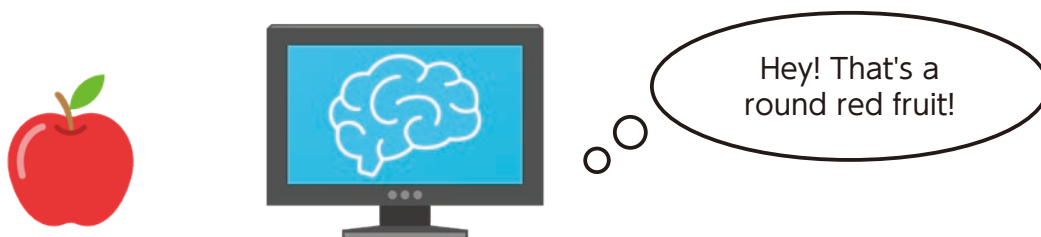
● The Drawbacks

An important thing to remember is that generative AI doesn't understand the meaning of words like humans do. It can only make patterns based on its training data! This means that, while it can use that data to create something pleasantly surprising, it might make obvious mistakes or even give you different replies to the same question!

● Handling Other Data

Large language models are programmed to work with language. But why can a multimodal generative AI model which includes an LLM process other kinds of data like video and audio?

The truth is that a multimodal generative AI handles this data just like it processes words! This means that if you give the model a picture of an apple, it can associate it with the description, “round red fruit,” internally. Processing data in the same way as words lets it work with all sorts of data!



Try it Out Processing Data the AI Way

How would a generative AI describe the things on your table or desk? Pick an object and use the table below to describe it!

② Breaking it Down

Pick an object on your table or desk, then fill out the table with words that generative AI would use to describe it.

What is it?	Ballpoint Pen
What does it look like?	○ cm long, black...
How would you describe it?	Almost empty, laying on desk...

② Getting Descriptive

Let's say we send the text generated in ① to another AI and ask it to make a picture. Did you use enough words to make a clear description which lets the AI recognize the object?



If you didn't, there's no way for the AI to know what it's missing! Putting your data into words that an AI can understand is the best way to get the results you want.

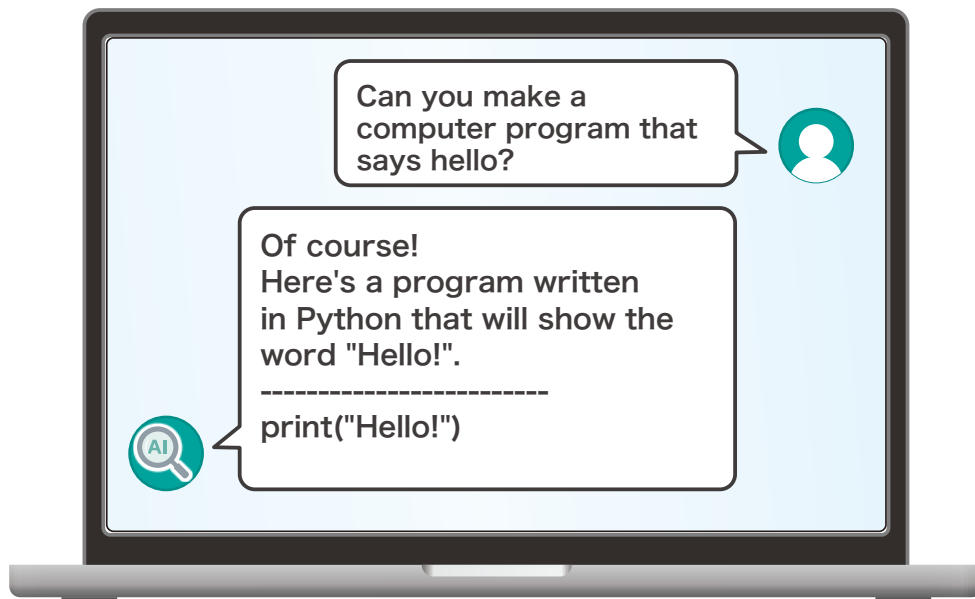


A pencil with an eraser
laying on a wooden
table. It's 15 cm long
and...

[Chapter 2] Programming with GenAI

There are lots of uses for generative AI, but it's especially popular when it comes to programming. A good amount of the data used to train AI to program comes from the same programs used to make websites and video games! This means that anyone can program just by asking questions, even if they aren't an expert!

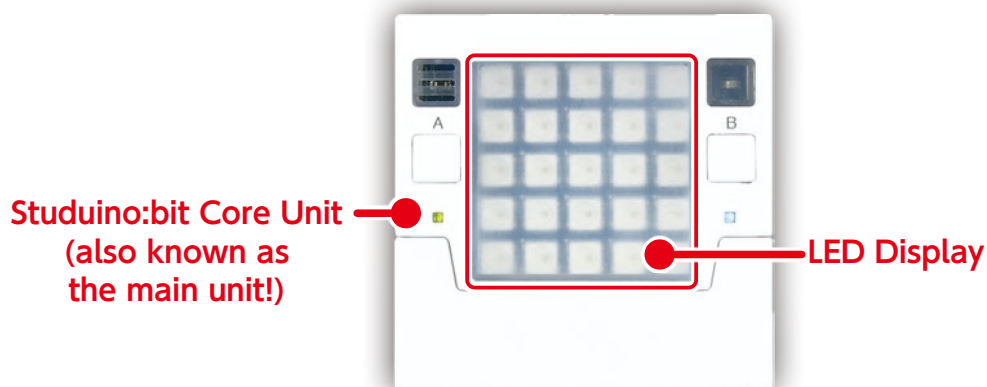
Example Programming Hello



This series of textbooks teaches you how to use generative AI to program robots and systems. Doing this will help you learn the characteristics of AI as well as how to use it best!

● The Hardware

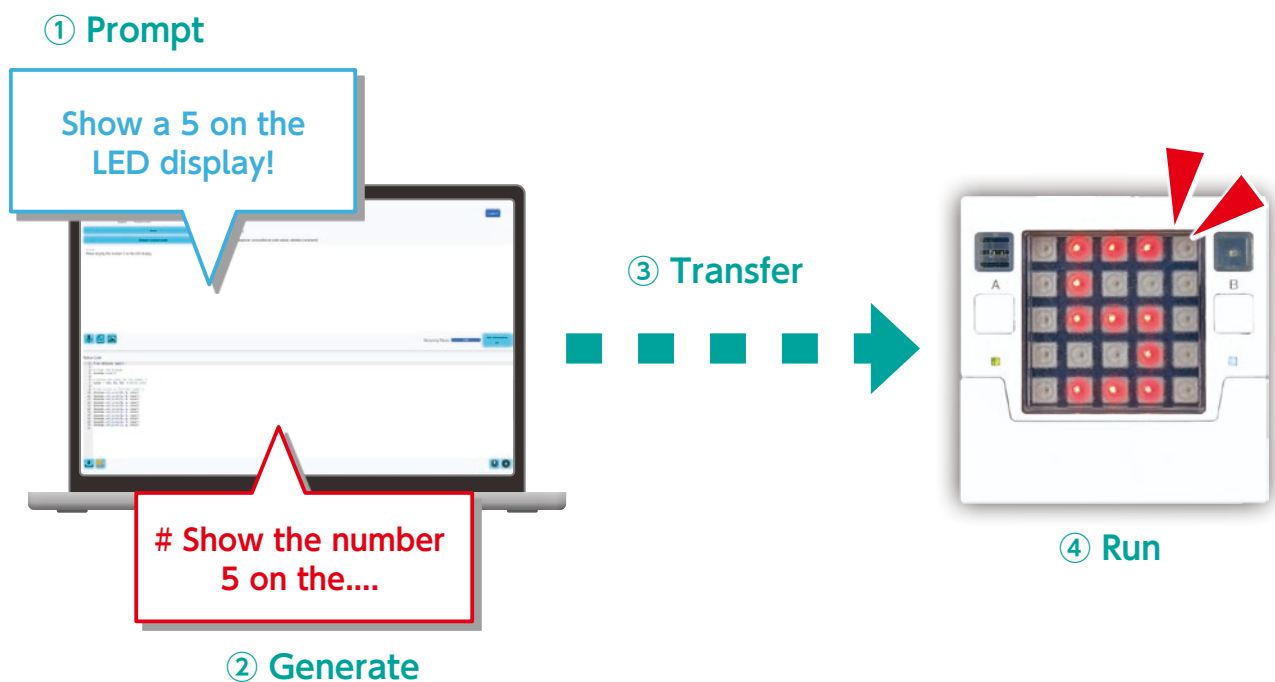
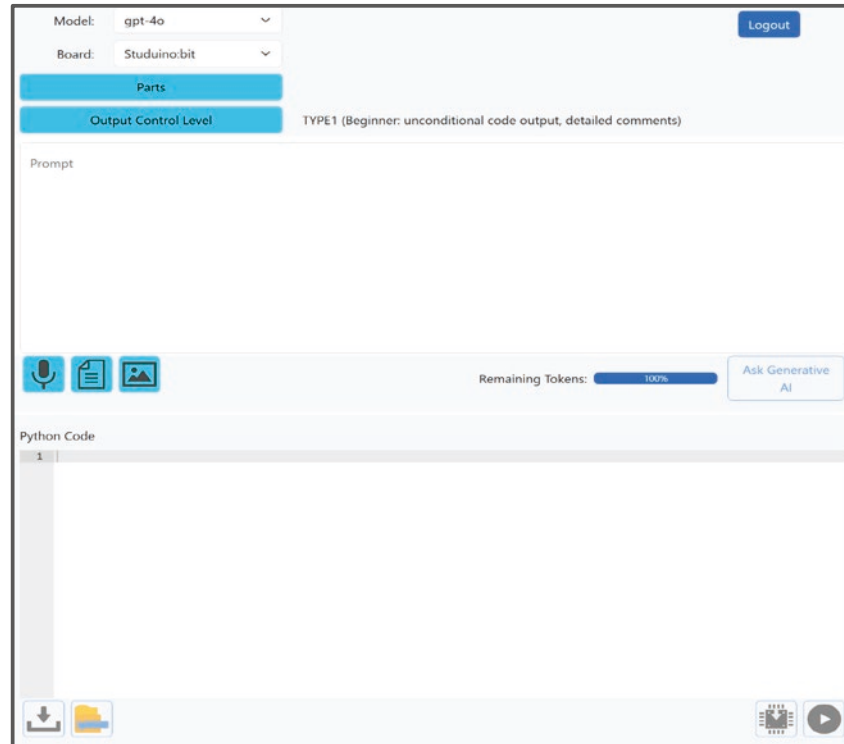
The programs you're going to make will run on hardware from the **ArtecRobo 2.0** programming platform.



The Studuino:bit Core Unit features a 5 x 5 full color LED display. Each LED in the display can light up individually to make pictures and even animations!

● The Software

We'll use the **Artec Intelligence** software to program the hardware with the power of generative AI. Just type a prompt describing what you want your robot or system to do and the generative AI will make a program that does it. Once done, just transfer the program to make your robot or system work!



◆ 2-1. Using Artec Intelligence

Try it Out Generating a Program

Let's use the Artec Intelligence software to program your Core Unit's LED display.

① Opening the Software

There are different versions of the software depending on your connection type. Use the QR codes or URLs below to open the right version for you!

USB



 <https://app.artec-kk.co.jp/artecintelligence/usb/>

Bluetooth



 <https://app.artec-kk.co.jp/artecintelligence/bt/>

iPad (Bluetooth)



 <https://apps.apple.com/us/app/artec-intelligence/id6755988678>

★ You need to download the app to use it on iPad!

② Logging In

Type in your login details to log in to the software!

Artec Intelligence Login

User ID

Password

Login



Model: gpt-4o
Board: StudioBit
Parts
Output Control Level
TYPE1 (beginner: unconditional code output, detailed comments)

Prompt



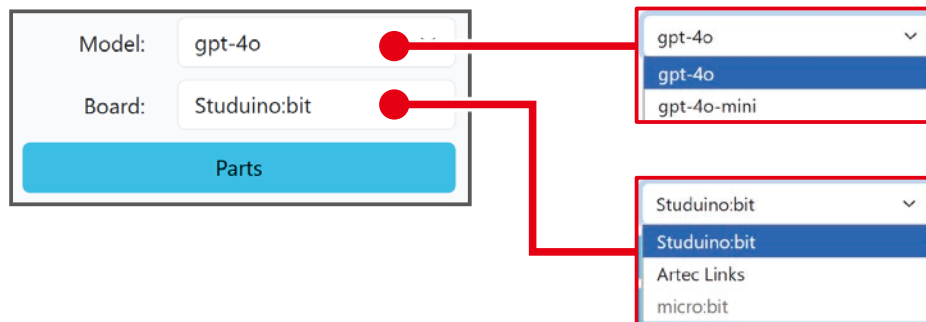
Remaining Tokens: 1000

Ask Generative AI

Python Code

③ Settings

You need to make the right settings before to type your prompt!
Choose gpt-4o^{★1} under Model and Studuino:bit^{★2} under Board.

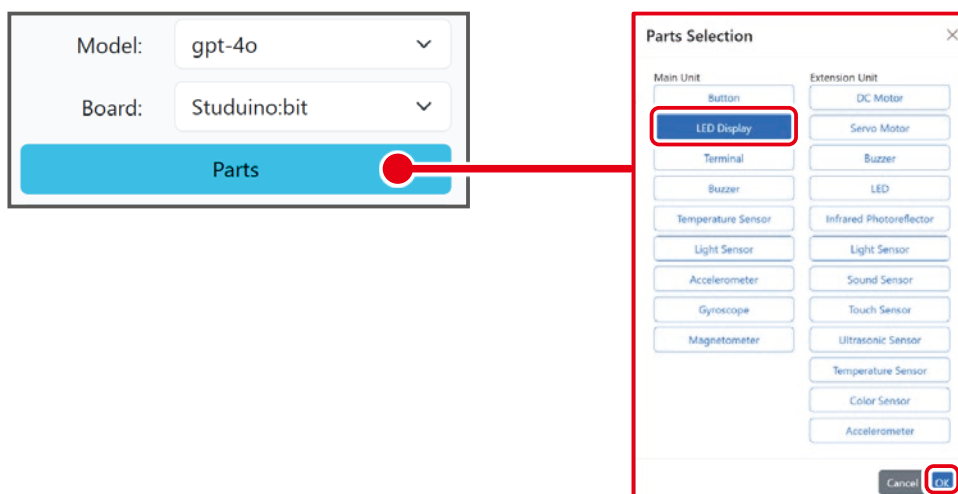


★ 1 gpt-4o is a generative AI model made by OpenAI.

★ 2 The Studuino:bit Core Unit is the microcontroller used in ArtcRobo 2.0!

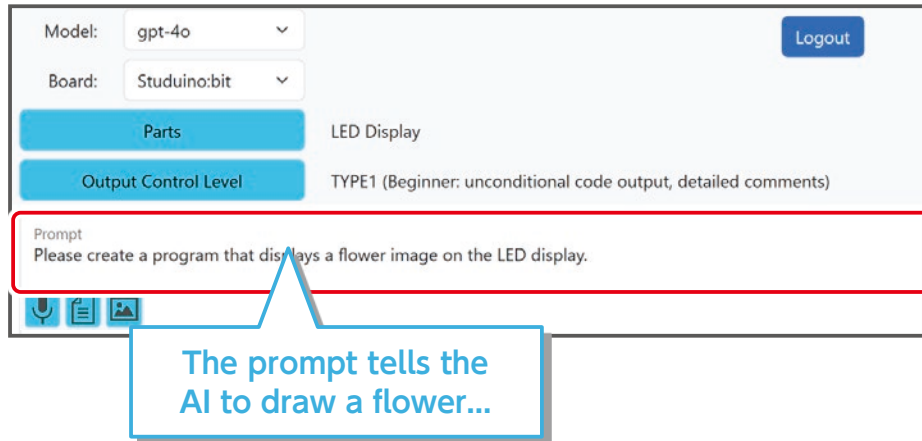
④ Choosing Parts

This is where you select the ArtecRobo 2.0 parts you want to program. Pick the LED display here!



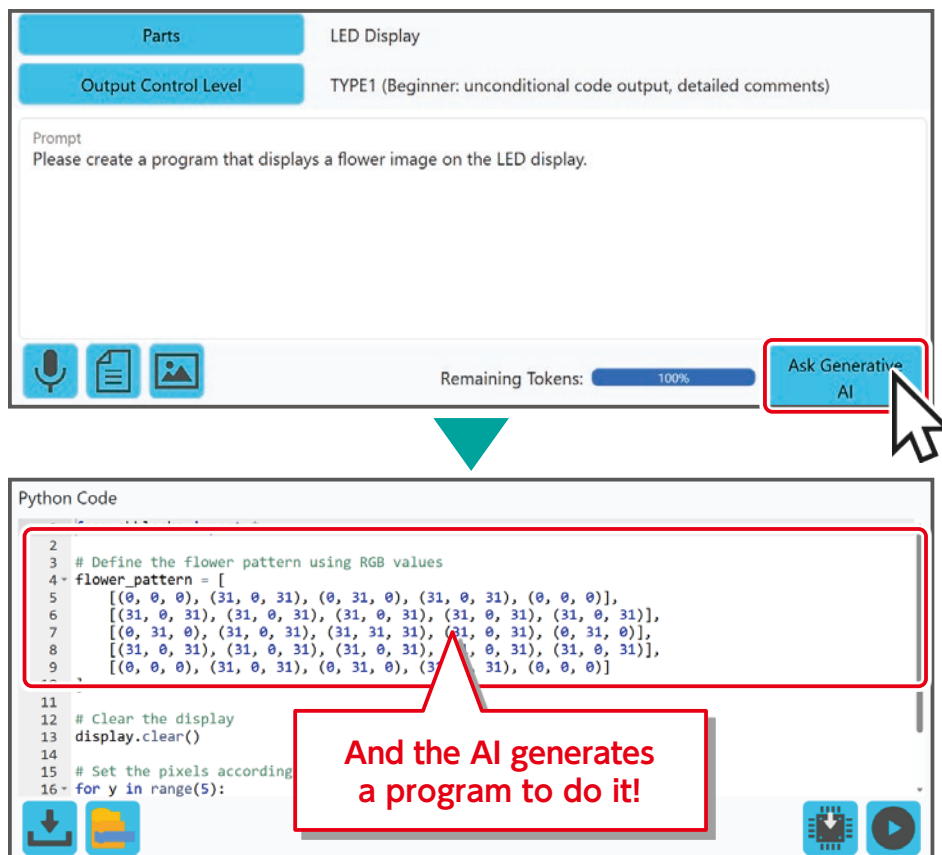
⑤ Typing the Prompt

The instructions you pass to the AI are known as **prompts**. You type these prompts into Artec Intelligence using your keyboard. Try using your keyboard to ask it to draw a flower!



⑥ Generate the Program

Just click the Ask Generative AI button and the generative AI will read your prompt and make the program you ask for!



The programs your software generates are written in a language called Python. Clicking anywhere on the code will let you use your keyboard to edit the program yourself!

⑦ Running Programs

To run your program, transfer it to your Core Unit.

★ The USB and Bluetooth versions of the software connect in different ways. Follow the steps below for your connection type!

```
Python Code
1 from sbblocks import *
2
3 # Define the flower pattern using RGB values
4 flower_pattern = [
5     [(0, 0, 0), (31, 0, 31), (0, 31, 0), (31, 0, 31), (0, 0, 0)],
6     [(31, 0, 31), (31, 0, 31), (31, 0, 31), (31, 0, 31), (31, 0, 31)],
7     [(0, 31, 0), (31, 0, 31), (31, 31, 31), (31, 0, 31), (0, 31, 0)],
8     [(31, 0, 31), (31, 0, 31), (31, 0, 31), (31, 0, 31), (31, 0, 31)],
9     [(0, 0, 0), (31, 0, 31), (0, 31, 0), (31, 0, 31), (0, 0, 0)]
10 ]
11
12 # Clear the display
13 display.clear()
14
15 # Set the pixels according to the flower pattern
16 for y in range(5):
```

Click here to transfer!

★ You can also click  to run a program for non-iPad devices.
Find out more on page 17 of the official Instruction Manual!

https://www.artec-kk.co.jp/artecintelligence/en/pdf/Artec%20Intelligence_Instruction%20Manual.pdf

! USB

① Use a USB cable to connect your microcontroller to your device.



② Click  and the following menu will appear.
Choose your microcontroller, then click Connect!

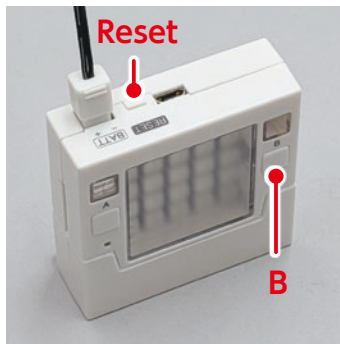


! Bluetooth

- ① Turn your device Bluetooth on.
- ② Plug in your Battery Box or Power Bank.

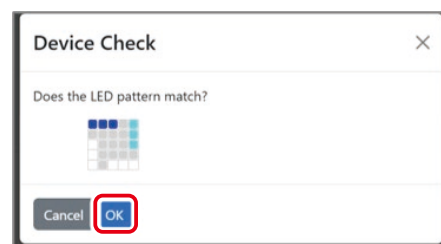
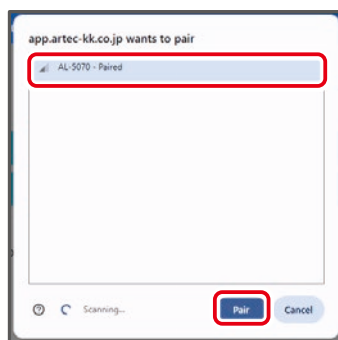


- ③ Hold down the B button and press Reset once. The LED display will show a pattern! This pattern is unique to each Core Unit.

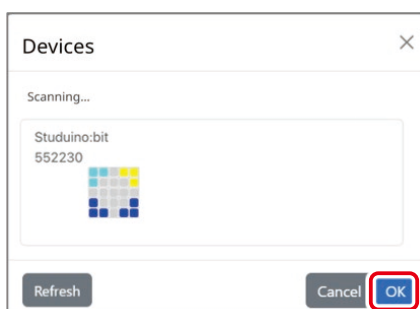


- ④ Click  and the following menu will appear. Choose the same pattern as the one on the microcontroller, then click Pair or OK.

Windows / Chrome OS / Android



iPadOS



If the transfer worked you'll see a flower on the LED display!

★ If the transfer doesn't work, see page 25 of the official Instruction Manual to troubleshoot!

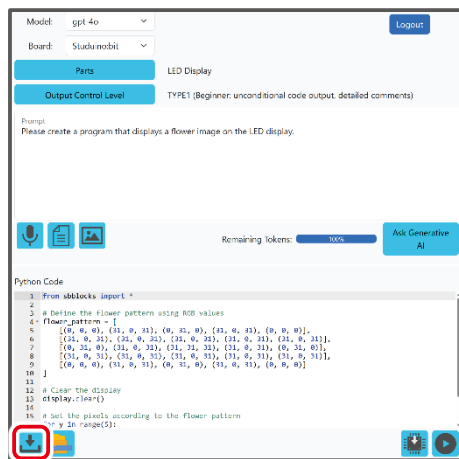
https://www.artec-kk.co.jp/artecintelligence/en/pdf/Artec%20Intelligence_Instruction%20Manual.pdf



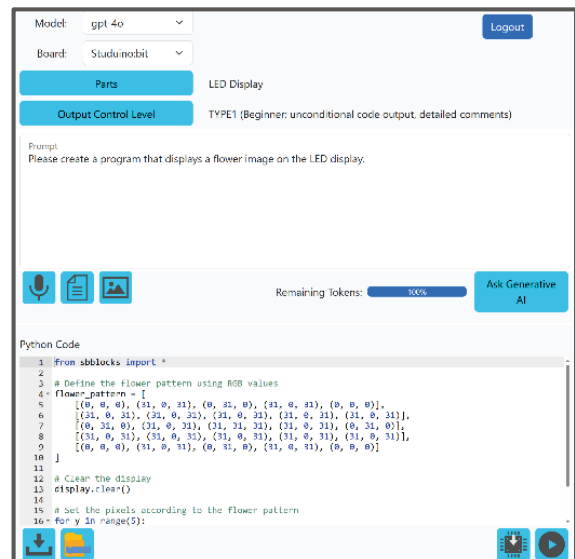
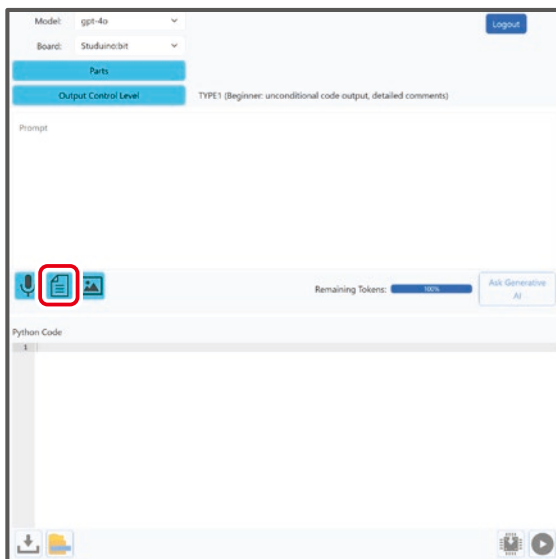
⑧ Saving the Program

You can save programs as a single project file which includes software settings and the prompt you used.

Click  below Python Code to download an .aipz file which contains your software settings, prompt, and the generated Python code.

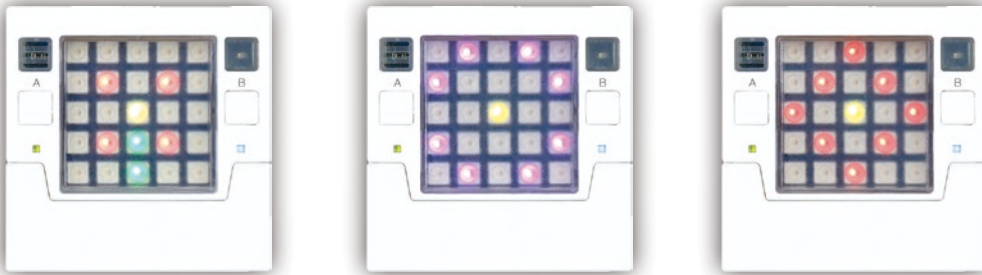


You can then use  to load it!

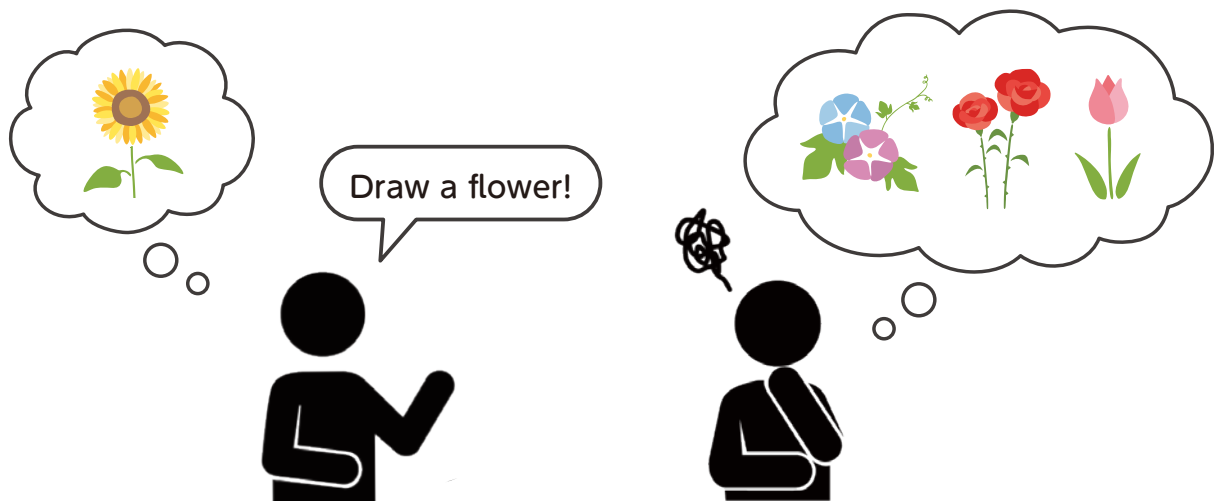


◆ 2-2. Comparing Results

Try comparing the flower on your LED display with other flowers. You'll see that every flower is different even if you used the same prompt! But why are they different?



Let's try thinking about it from a human perspective. If someone told you to draw a flower, what kind of flower would you draw? Anyone you ask to do this is going to choose one flower to draw out of all the flowers they know. Generative AI picks a flower at random from its training data in the exact same way!



Just like the exercise from chapter 1 where you described an object on your desk or table, you need to be as clear as possible when prompting with AI. For a flower, this means describing the color, shape, size, and anything else!

Try it Out **Getting the Right Flower**

Try making your prompt more descriptive to make the kind of flower you want!

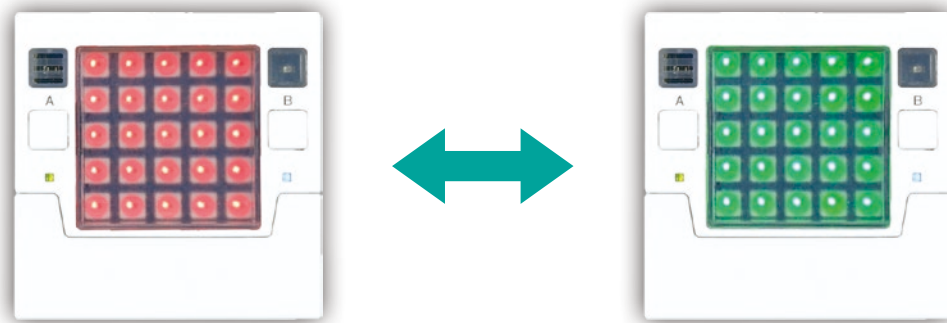
Prompt	Result

WARNING

Your prompts may be used as training data for generative AI models. Make sure that that they don't contain personal or confidential data before sending them!

Try it Out Red and Green

The flower you asked the AI to make is one, simple picture. Now let's try making something a little more complex by making the LED flash red and green in order!



Prompt	Result

In chapter 3, we'll start thinking about how to make prompts which tell AI exactly what kind of program we need!

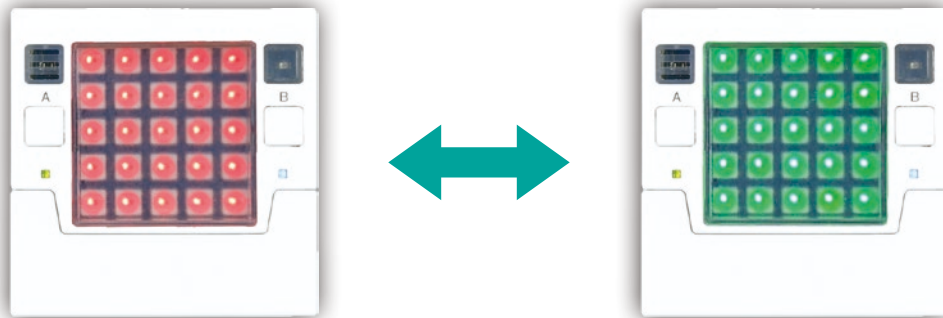
[Chapter 3] Engineering Prompts

In chapter 2, you used prompts to get generative AI to make programs for your Core Unit. You also thought about how to fine tune the words in your prompt to make it easy for the AI to understand. The process of making a prompt which gets the exact results you want is known as **prompt engineering**!

◆ 3-1. Elements of a Prompt

Engineering is the field of using technology to design and build a whole range of things. The best prompt engineers mix and match parts to make sure that the AI understands it as clearly as possible!

Just like you need basic elements like walls and a roof to make a house, you need **four elements** in order to make a good prompt. Take the prompt you made on the last page as an example.



Element ① Clear Direction

The most important element is clearly telling the AI what you want it to do. Make sure you avoid any ambiguous phrasing or words, being as clear as possible to make a prompt that anyone can understand just by reading it!

X

The Prompt

Light up the LED display alternately in red and green.

- The display lights up, but each LED is an alternate color...
- The display stays lit instead of lighting up alternately...



- Use a clearer prompt...
- And add a condition!

O

The Prompt

Make the LED display **blink** alternately in red and green.
Make sure that every LED is the same color.

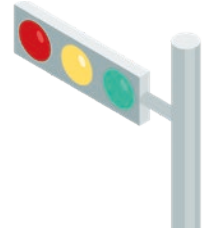
★ In a prompt, light up means the display or LED stays on and blink means they turn on for a short time!

Element ② Deeper Context

This element means giving information on how you're going to use the program you want to make. Telling the AI why it needs to make the program is what lets it give you the exact answer you need!

The Prompt

I want to simulate the changing colors of a traffic light.
Please make the LED flash alternately between red and green.



Roleplaying

If you want to give an AI context, try role playing by asking it to imagine it has a specific job. Assigning it a job makes it give answers which keep the context and history of that job in mind!

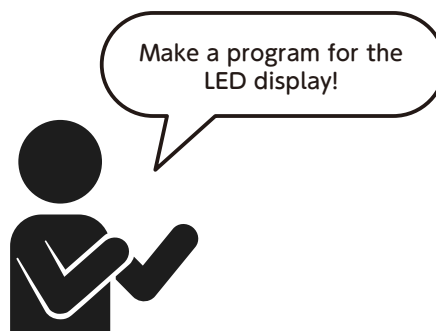
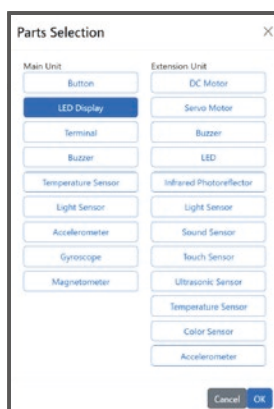
Tuning

You are a professional pixel artist who specializes in creating 5 x 5-pixel pixel art. Draw a picture of a flower.

Element ③ The Input

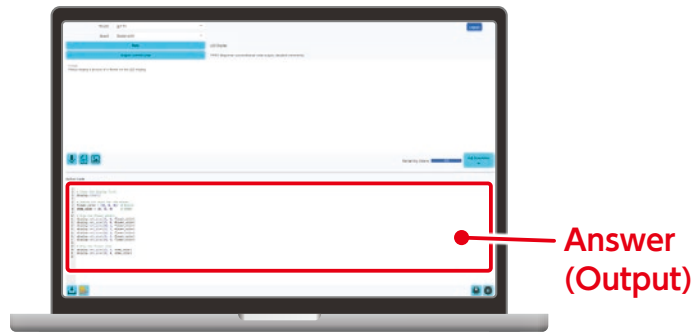
Asking AI to work with certain data or rules in mind is a key part of helping it make decisions. If you asked it to make a three-sentence summary of a news article, for example, that article would be the input!

When you're working with Artec Intelligence, this input is the parts you select.



Element ④ Output Format

This element specifies what kind of format or appearance you want the AI's answer (output) to have. The reason Artec Intelligence always answers in Python code is because the output format is specified behind the scenes.



! Hardware Prompts

Behind the scenes, Artec Intelligence automatically adds a common instruction to your prompt along the lines of, "Output the information needed to control the selected parts as Python code that runs on the Studuino:bit." This software calls this mechanism a **hardware prompt**.

◆ 3-2. Prompting Basics

From here, let's learn about **prompting**, the art of writing prompts that draw out the exact answer you want. Let's start with the following challenge.

Try it Out Making Faces on the LED Display

Let's use the LED display's lighting pattern to show a facial expression. Below are the results of the prompts that Anita and Ben each made.

	Happy Face	Sad Face
Anita		
Ben		

The colors are different, but the LEDs are lit in the same positions, so it's hard to tell the expressions apart...



The eyes and mouth are totally different shapes, so you can easily tell the expressions apart. Nice!



What kind of prompt did Anita and Ben each write? Try it out for yourself in Artec Intelligence as you think it through.

Let's take a look at the prompts Anita and Ben used to make their faces.

● Happy Face

Anita

Tuning

Please display a happy face.

Ben

Tuning

■ Face Examples

- For an angry face, the eyes are at (0,0), (1,1), (3,1), and (4,0), and the mouth is at (0,4), (0,3), (1,3), (2,3), (2,4), (3,3), (4,3), and (4,4).
- For a crying face, the eyes are at (0,0), (0,1), (1,0), (4,1), (3,0), and (4,0), and the mouth is at (0,4), (1,3), (2,3), (3,3), and (4,4).
- The eyes are green, and the mouth is red.

■ Instructions

Please display a happy face.

● Sad Face

Anita

Tuning

Please display a sad face.

Ben

Tuning

■ Face Examples

- For an angry face, the eyes are at (0,0), (1,1), (3,1), and (4,0), and the mouth is at (0,4), (0,3), (1,3), (2,3), (2,4), (3,3), (4,3), and (4,4).
- For a crying face, the eyes are at (0,0), (0,1), (1,0), (4,1), (3,0), and (4,0), and the mouth is at (0,4), (1,3), (2,3), (3,3), and (4,4).
- The eyes are green, and the mouth is red.

■ Instructions

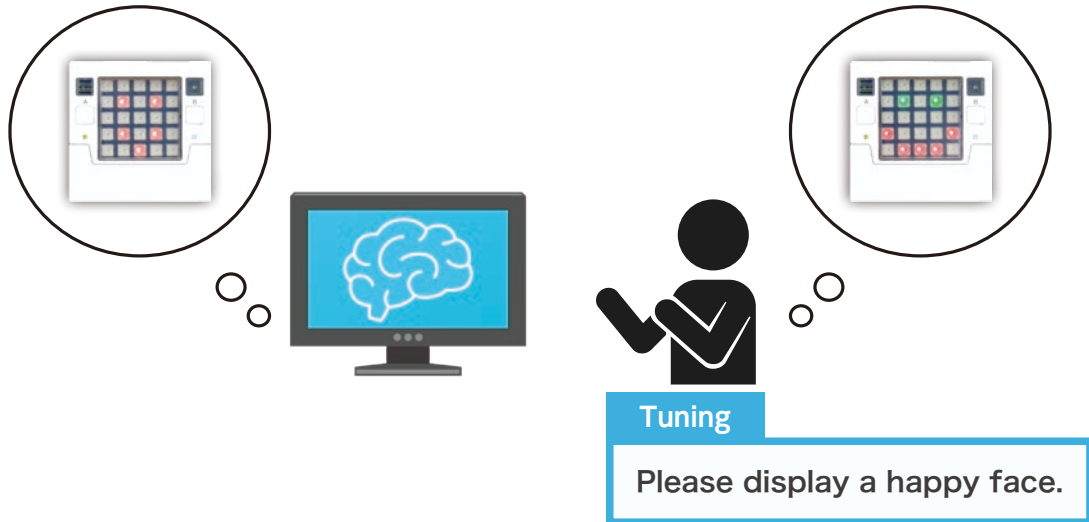
Please display a sad face.

Anita and Ben's prompts each match one of the following two prompting techniques.

- ★ The numbers like "(0,0)" used in Ben's prompt are "coordinates," a system for indicating LED positions. We'll go over how to read them in detail on page 28.

Zero-Shot Prompting

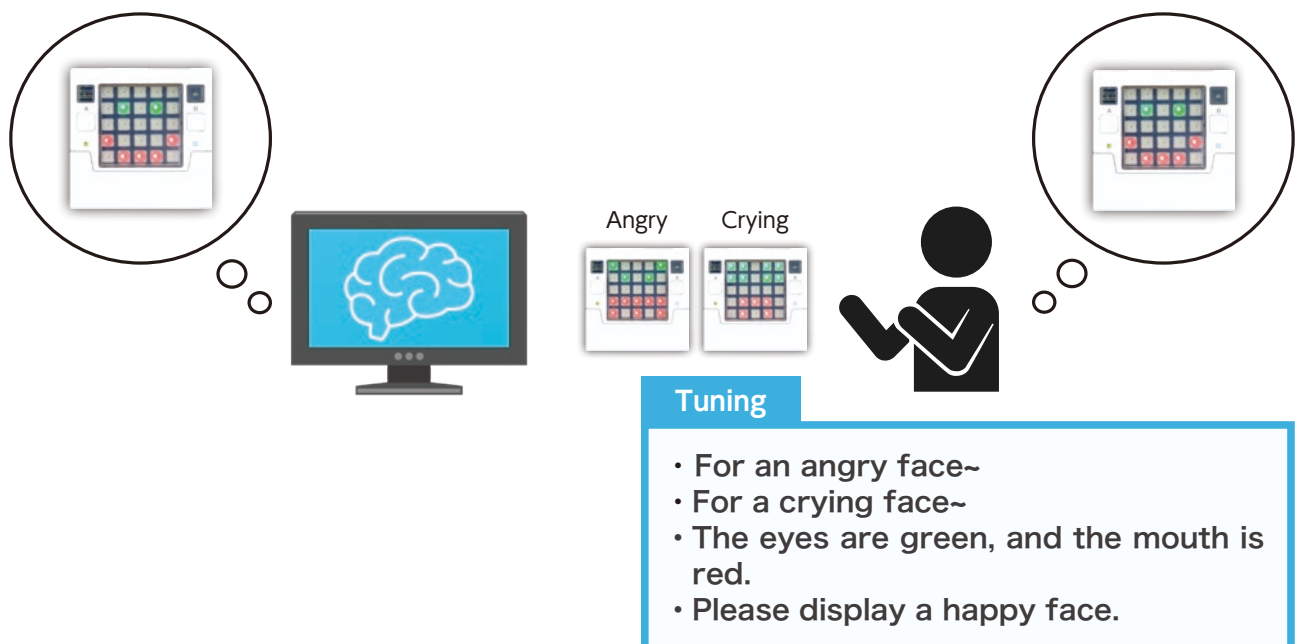
Giving an instruction straightaway without providing any examples, like Anita's prompt, is called **zero-shot prompting**. It's an easy technique to try, but it also makes it harder to get the exact answer you had in mind.



The prompt "Please draw a picture of a flower" from chapter 2, which generated all sorts of different flowers, is also an example of zero-shot prompting.

Few-Shot Prompting

Giving a few examples before the instruction, like Ben's prompt, is called **few-shot prompting**. Providing examples helps the AI understand a common pattern for what kind of tone and rules to follow in its answer. This gets you an answer closer to what you actually wanted than zero-shot prompting does.

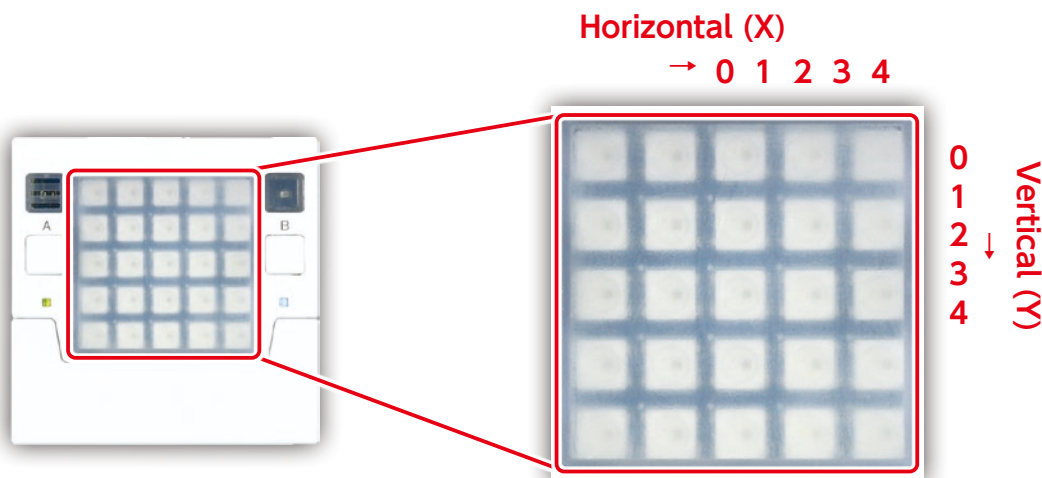


◆ 3-3. Prompting with Coordinates

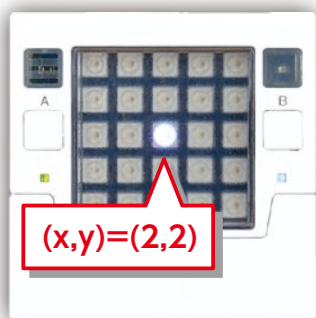
Few-shot prompting is useful when you want the AI to copy the pattern or rule you prefer. Even so, depending on how the AI judges things, you might not get exactly the answer you intended. That's because no matter how many examples you show it, the fine details of which LEDs to light up are ultimately left up to the AI's judgment. In cases like this, it helps to specify exactly which LEDs to light up using coordinates.

! LED Display Coordinates

The LED display is made up of 25 LEDs arranged in a 5-by-5 grid. The position of each LED can be specified using vertical (Y) and horizontal (X) coordinates.



For example, if you want to light up the LED in the center of the display, either of the following prompts will work.



Tuning

Please turn on the LED in the center of the LED display.

Tuning

Please light up the LEDs located two across and two down on the LED display.

Tuning

Please turn on the LED located at position (2,2) on the LED display.

◆ 3-4. Prompt the Right Way

On page 26, Ben used symbols like “■” and “•” to neatly organize his prompt, as shown below. Writing a prompt this way, so that a human can tell at a glance what's a heading and what's an example, is also very effective for helping generative AI understand where each section of information begins and ends.

Tuning

■ Face Examples

- For an angry face, the eyes are at (0,0), (1,1), (3,1), and (4,0), and the mouth is at (0,4), (0,3), (1,3), (2,3), (2,4), (3,3), (4,3), and (4,4).
- For a crying face, the eyes are at (0,0), (0,1), (1,0), (4,1), (3,0), and (4,0), and the mouth is at (0,4), (1,3), (2,3), (3,3), and (4,4).
- The eyes are green, and the mouth is red.

■ Instructions

Please display a happy face.

One common way to use symbols to clearly show the structure of a piece of text is called **Markdown**. A lot of the text on the Internet that large language models are trained on is written using this format. That's why organizing your prompt using Markdown helps generative AI understand exactly what you mean.

Common Symbols	What does it do?
#	• These are used to format titles. • Adding more symbols lets you organize information by size, like a large heading (#), medium heading (##), or small heading (###)!
-	• Use these to make a list. • You can also use “*” or “+”.

Writing Ben's prompt in Markdown looks like this.

Tuning

Face Examples

- For an angry face, the eyes are at (0,0), (1,1), (3,1), and (4,0), and the mouth is at (0,4), (0,3), (1,3), (2,3), (2,4), (3,3), (4,3), and (4,4).
- For a crying face, the eyes are at (0,0), (0,1), (1,0), (4,1), (3,0), and (4,0), and the mouth is at (0,4), (1,3), (2,3), (3,3), and (4,4).
- The eyes are green, and the mouth is red.

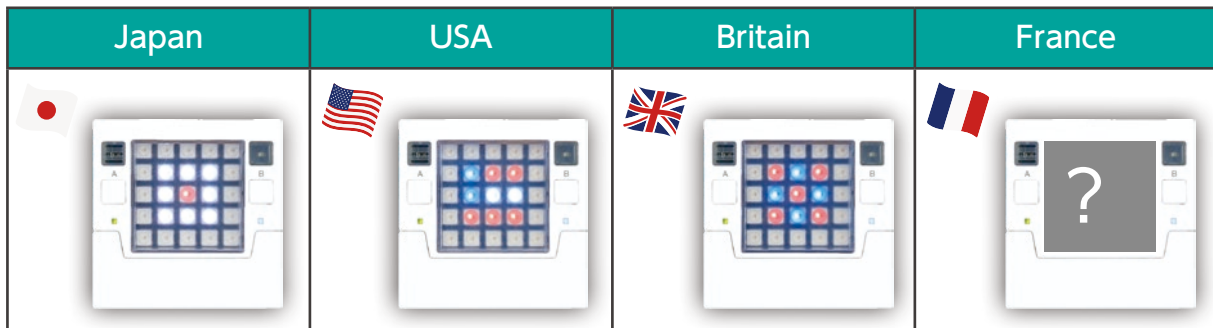
Instructions

Please display a happy face.

Let's use what we've learned so far to tackle the next challenge.

Try it Out Showing Flags on the LED Display

- ① Let's use the LED display's lighting pattern to show a national flag. Using the three flag examples below as a reference, let's get generative AI to display the flag of France.



- ② What kind of answer will you get if you use the prompt from ① to ask generative AI about something it doesn't know? As an example, let's try asking it the following two questions.
- a) The host country of the 2025 World Expo (correct if the Japanese flag is displayed)
 - b) The host country of the 2225 World Expo (not yet decided)

★ You can find example prompts on page 32.



Getting Accurate Answers

When you asked about the host country of the 2225 World Expo, even though there's no way to know the answer yet, chances are your AI displayed the flag of one of the countries you gave as an example, right? (We'll cover the reason for this in the column on the next page.)

To prepare for cases like this, it's a good idea to add an instruction to your prompt that stops the AI from giving a wrong answer, like the example below.

Tuning

Displaying National Flags

~~~

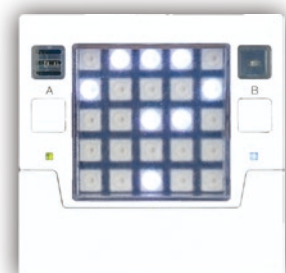
#### ## Instructions

Display the flag of the host country for the 2225 World's Fair.

**However, if you do not know the answer, display a "?".**

#### ## Displaying "?"

**Highlight the coordinates (1,0), (2,0), (3,0), (0,1), (4,1), (2,2), (3,2), and (2,4) in white.**



## How Generative AI Gets Things Wrong

In the last exercise, when you asked about the “host country of the 2025 World Expo,” which shouldn't be decided yet, the AI still confidently displayed the flag of some country, didn't it? As this shows, generative AI doesn't always give you the correct answer.

As you learned on the previous page, generative AI doesn't understand the meaning of words like a human does. It only creates sentences based on patterns from its training data. This means it can end up creating a “plausible-sounding sentence” even when the answer isn't in its data, giving you a wrong answer that sounds just as convincing as a correct one.

For example, what if it told you, “Pandas are members of the bear family, and their main food is meat. Adults eat 12 to 38 kg of meat a day.” How would you react? The sentence sounds natural and even includes specific numbers, but it's actually wrong. Pandas mainly eat bamboo.



Pandas are members of the bear family  
and their main food is meat.  
Adults eat 12 to 38 kg of meat a day.



This phenomenon where an AI outputs information that looks correct but isn't is called **hallucination**. Hallucination isn't a bug or a flaw. It's something that inevitably happens because of how AI works, creating sentences from patterns without understanding their meaning.

When using generative AI, it's important to design your prompts to avoid hallucination and to always check for yourself whether the answer it gives you is correct.

①

### Tuning

#### ## Displaying National Flags

Use pixels (1,1) through (3,3) on the LED display to display the national flags.

- The Japanese flag is represented with red at (2,2) and white everywhere else.
- The American flag is represented with blue at (1,1) and (1,2), white at (2,2) and (3,2) and red everywhere else.
- The British flag is represented with red at (1,1), (3,1), (2,2), (1,3), and (3,3), and blue everywhere else.

#### ## Instructions

Please display the French flag.

Example of the  
French Flag



② A)

### The Prompt

#### ## Displaying National Flags

Use pixels (1,1) through (3,3) on the LED display to display the national flags.

- The Japanese flag is represented with red at (2,2) and white everywhere else.
- The American flag is represented with blue at (1,1) and (1,2), white at (2,2) and (3,2) and red everywhere else.
- The British flag is represented with red at (1,1), (3,1), (2,2), (1,3), and (3,3), and blue everywhere else.

#### ## Instructions

Please display the flag of the host country for the 2025 World Expo.

B)

### The Prompt

#### ## Displaying National Flags

Use pixels (1,1) through (3,3) on the LED display to display the national flags.

- The Japanese flag is represented with red at (2,2) and white everywhere else.
- The American flag is represented with blue at (1,1) and (1,2), white at (2,2) and (3,2) and red everywhere else.
- The British flag is represented with red at (1,1), (3,1), (2,2), (1,3), and (3,3), and blue everywhere else.

#### ## Instructions

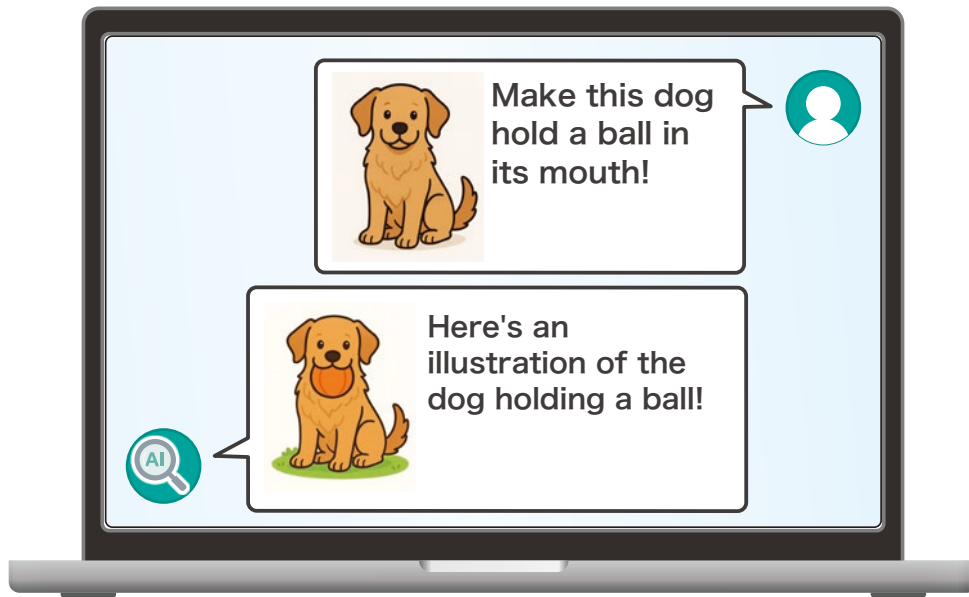
Please display the flag of the host country for the 2225 World Expo.



## [Chapter 4] Prompting with Pictures

### ◆ 4-1. Asking with Images

In chapter 1, you learned about multimodal generative AI, which can combine different kinds of information. Because this kind of AI treats data like images and audio the same way it treats words, you can freely combine them with text in your prompts.



You can also upload images to be used along with text prompts.

#### Try it Out Choosing a Color from an Image

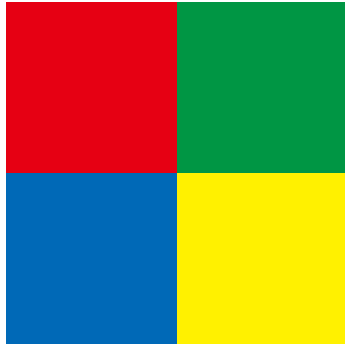
Follow the steps on the next page and try lighting up the LED using one of the colors from an image.



## ① Preparing an Image

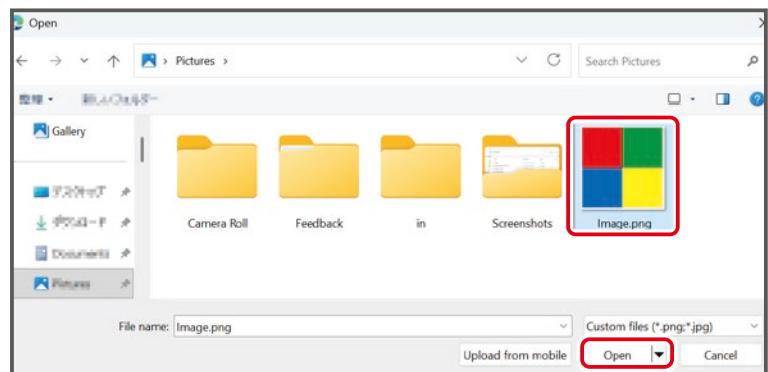
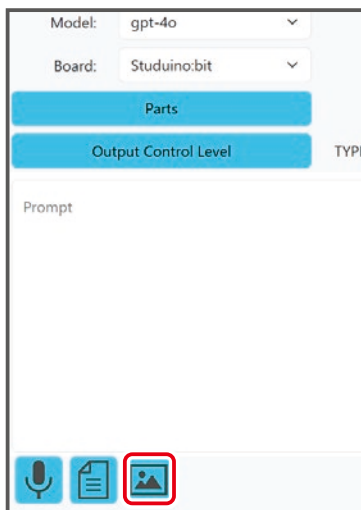
Take a photo of the four colors below, or draw them using paint software, and save the image to your device.

★ Artec Intelligence can only load png and jpeg files.



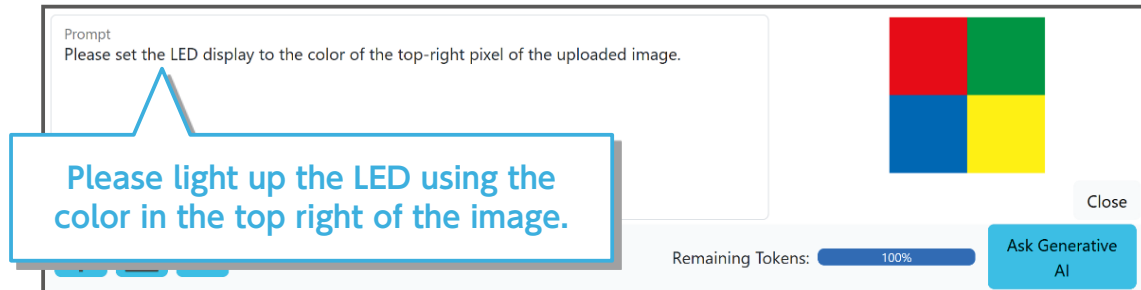
## ② Loading an Image

Click  in the bottom left of the Prompt box to load the image you prepared in ①.



### ③ Typing Prompts

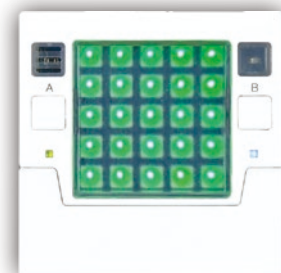
Once the image loads successfully, it'll appear to the right of the prompt. With the image loaded, type "Please light up the LED using the color in the top right of the image" into the prompt, and the AI will generate a program.



★ If you load the wrong image, click Close in the bottom right of the loaded image.

### ④ Generating the Program

Click Ask Generative AI to create the program. Try transferring your program to the ArtecRobo 2.0.



## Helpful Hints

Depending on your prompt, generative AI might not be able to generate code, and you may see a message like this in the Python code. If that happens, try making your prompt more detailed and asking again.

```
Python Code
1 No code was generated.
2
3 - Hint:
4 If the selected parts and instructions match, try changing the instruction to one that asks to create a program, like the example below.
5 Example: Please draw a picture of a flower -> Please create a program that draws a picture of a flower
```

Example)

Please light up the LED using the color in the top right of the image.

→ Please light up every LED using the color in the top right corner of the four colors arranged in a 2 x 2 grid.

## ◆ 4-2. Asking with Flowcharts

Using an image makes it easy to share a visual impression, but the instructions can end up more abstract than words.

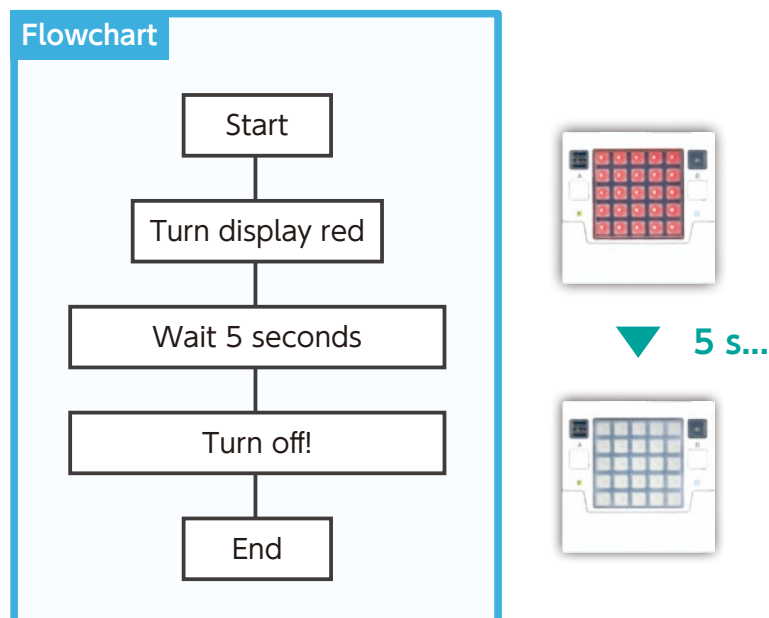
That's where a **flowchart**, which shows the flow of a process using shapes, comes in handy. Turning a program into a flowchart following set rules lets you convey the steps of a process more intuitively and specifically than words alone.

### ! What's a Flowchart?

It's a set of standard shapes used to represent the algorithm, or process and structure, behind a program. In a flowchart, each type of process is shown using a fixed symbol, and these are connected by lines.

|              |                                                                     |
|--------------|---------------------------------------------------------------------|
| Command      | Each individual process is shown as a rectangle.                    |
| Start<br>End | The "start" and "end" of a process are shown as rounded rectangles. |

For example, a program that lights up the LED in red for just 5 seconds would look like this as a flowchart. Symbols connected by lines are executed one at a time, from top to bottom. This method of processing is called **sequential processing**.



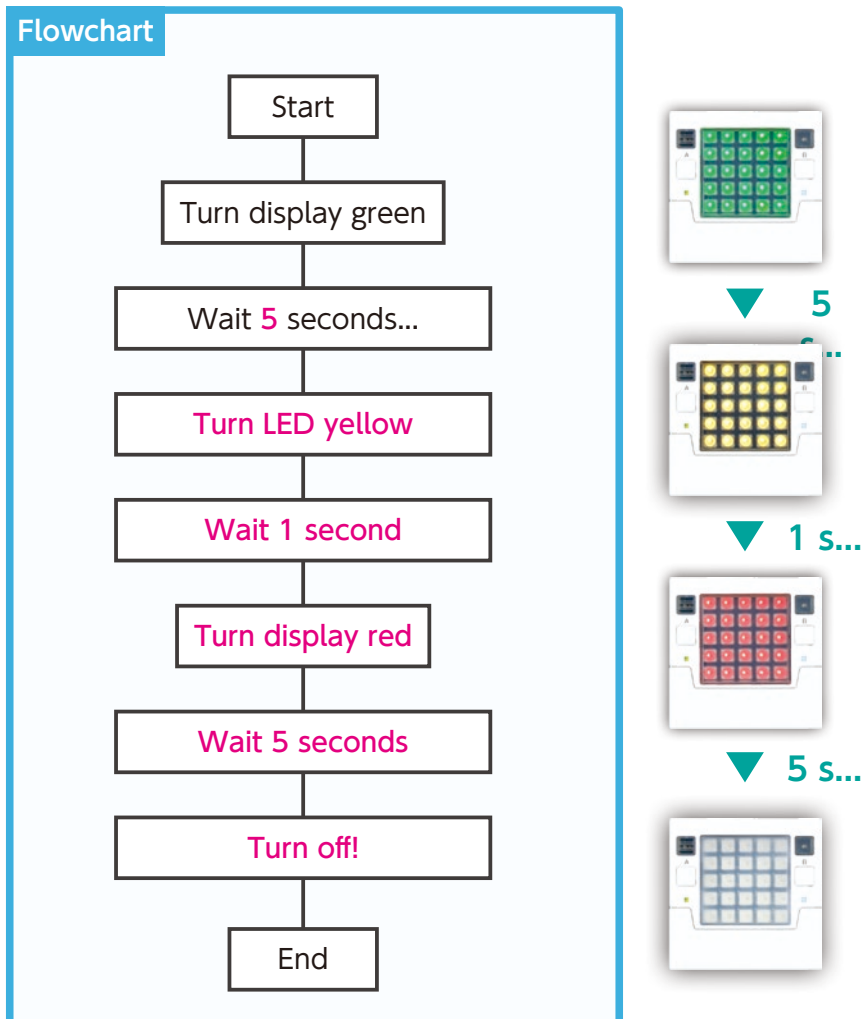
Take a photo of the flowchart, load it onto your device, and ask using the prompt on the right. The AI will create a program that follows the flowchart's steps.

#### The Prompt

Please write a program based on the flowchart.

## Try it Out Making a Program for a Traffic Signal

Fill in the flowchart below and try generating the traffic signal program. This traffic signal changes the LED's color in order from blue (green) to yellow to red, as shown in the diagram below.



Load this flowchart image, type in a prompt, and try asking generative AI.

### Helpful Hints

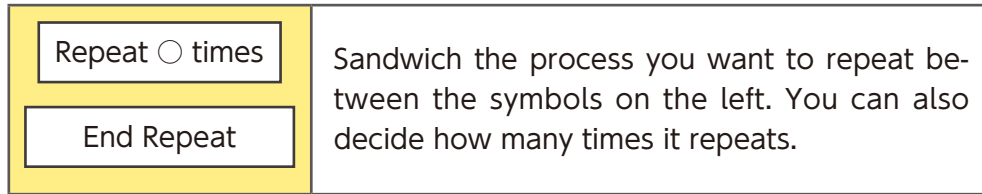
When generative AI reads an image, it can sometimes fail to understand it correctly if there's too much text. When you want to make a complex program, use the flowchart to explain the flow and add the detailed instructions in your prompt.

#### The Prompt

Please write a program based on the flowchart.  
## Conditions  
- Red: Turn on all LEDs in red  
- Yellow: Turn on all LEDs in yellow  
- Green: Turn on all LEDs in green

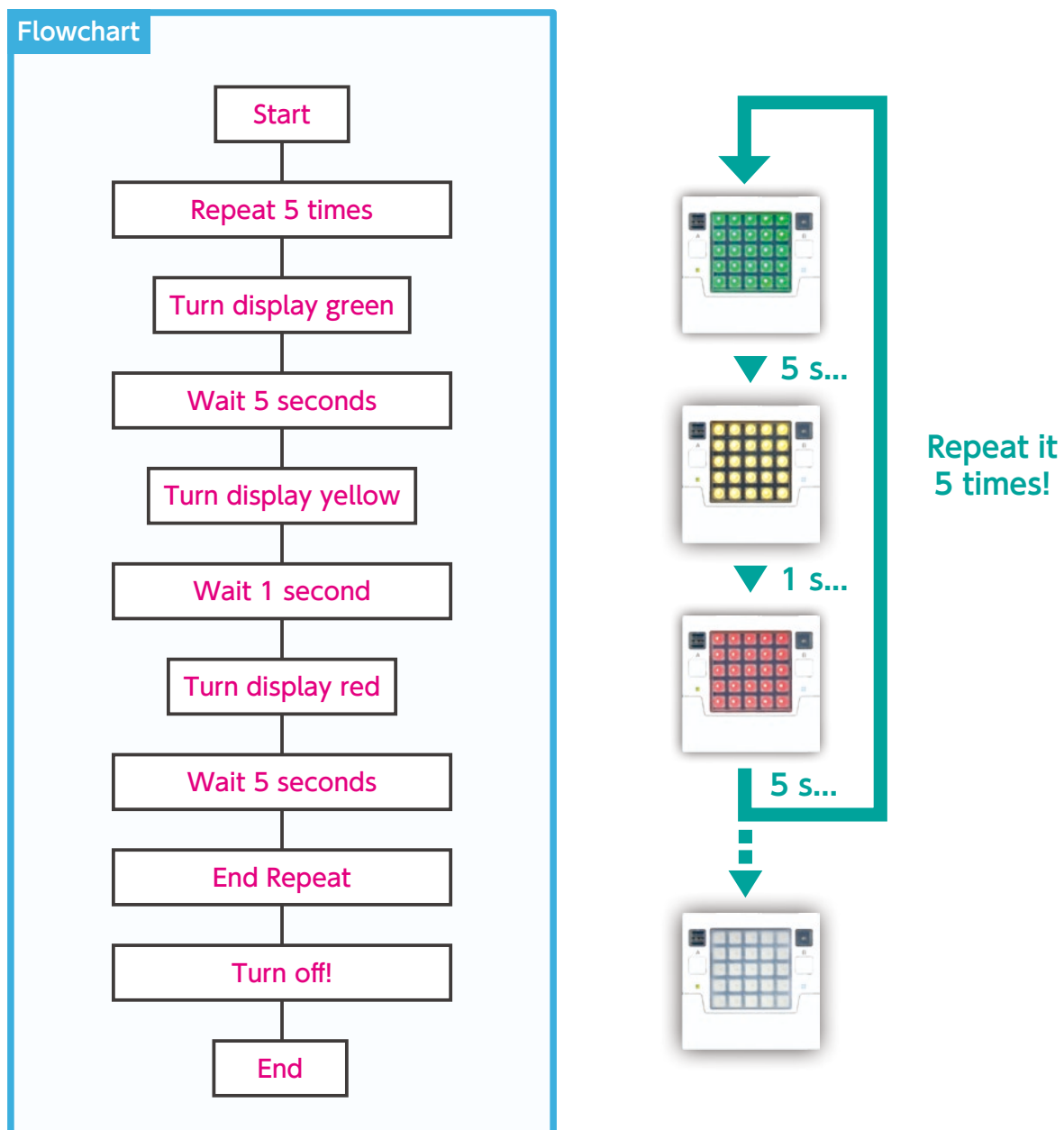
### ◆ 4-3. Programming Repeats

The traffic signal program you made on the previous page finishes after lighting up blue, yellow, and red just once each. Next, let's add a repeat symbol to the flowchart so it runs over and over again, just like a real traffic signal.



#### Try it Out Making the Traffic Signal Repeat

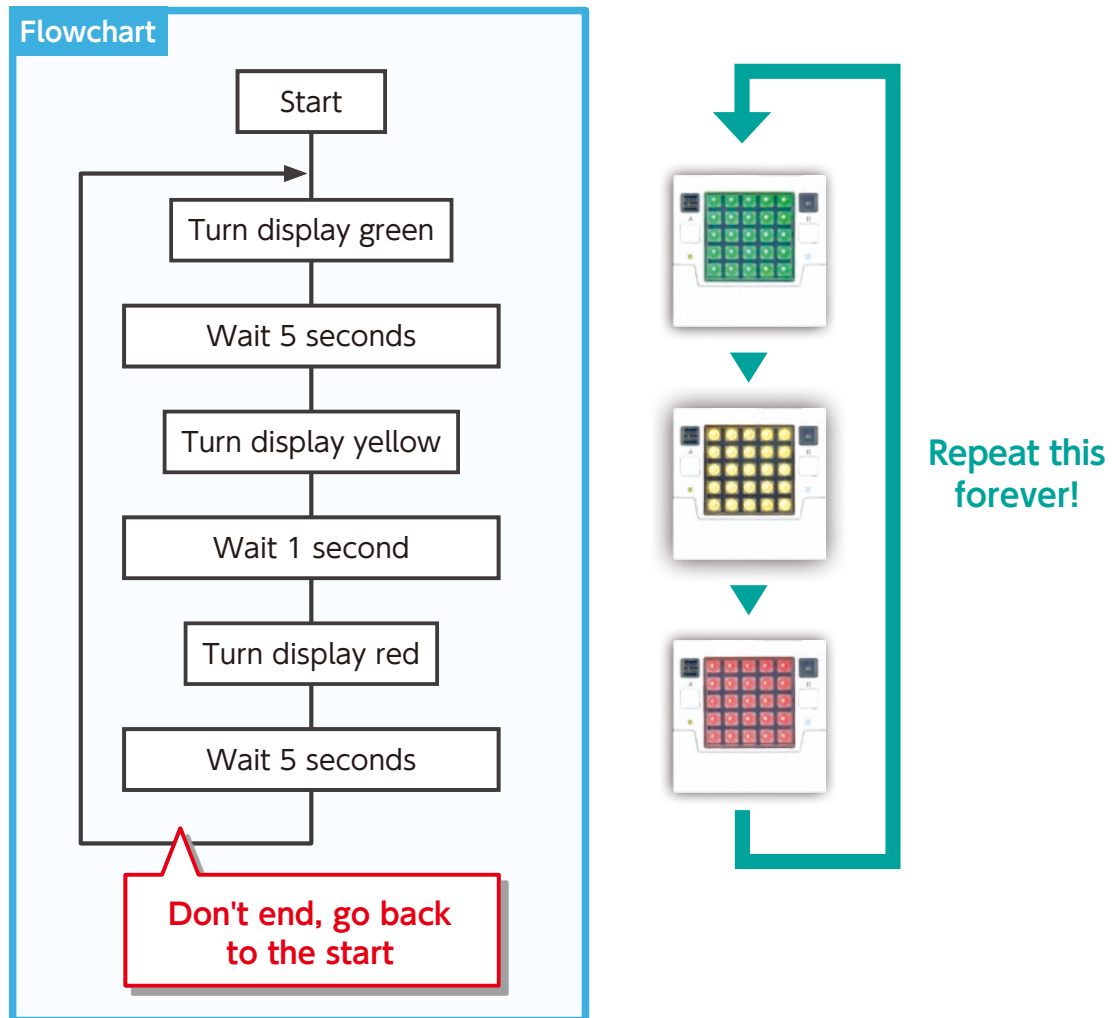
Fill in the flowchart below so the traffic signal process repeats 5 times, take a photo with your device, and ask generative AI. Use the same prompt as the previous page.



Using the repeat symbol to repeat the same process lets you keep the flowchart short and simple. This kind of process, where the same action repeats, is called **iterative processing**.

### How to Show Repeats (Iteration)

If you want a program to keep repeating forever instead of ending, you can show this in a flowchart using an arrow, like below.

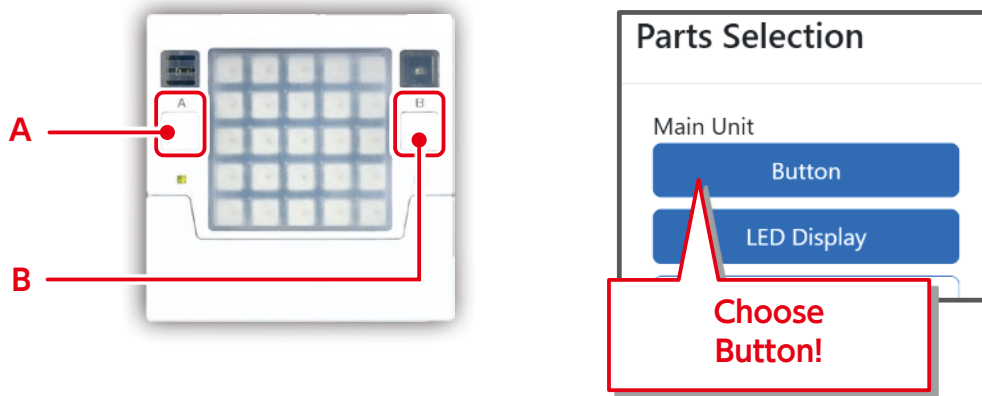


## ◆ 4-4. Programming Conditions

Not all traffic signals switch automatically. Some are push-button signals that change when someone presses a button. This time, let's make a signal program that changes its behavior when you press a button on the Core Unit.

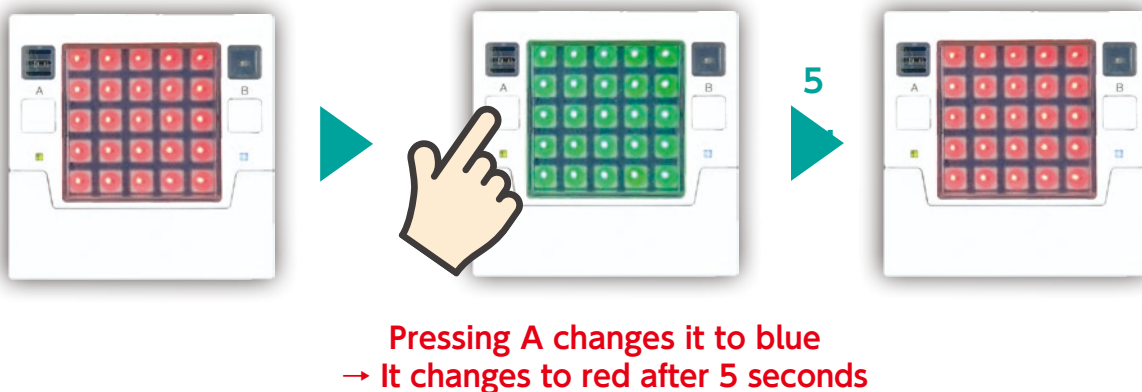
### ! What's a Button?

The ArtecRobo 2.0 has a button on each side of the LED display. Selecting Button under Parts in Artec Intelligence lets you make a program that does different things depending on whether the button is pressed.

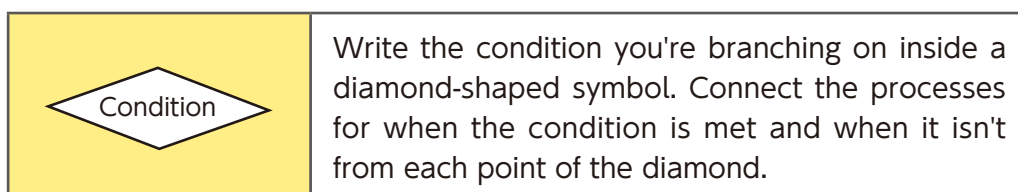


### Try it Out Making a Push-Button Signal

Let's make a program where pressing Button A changes the signal from red to blue.

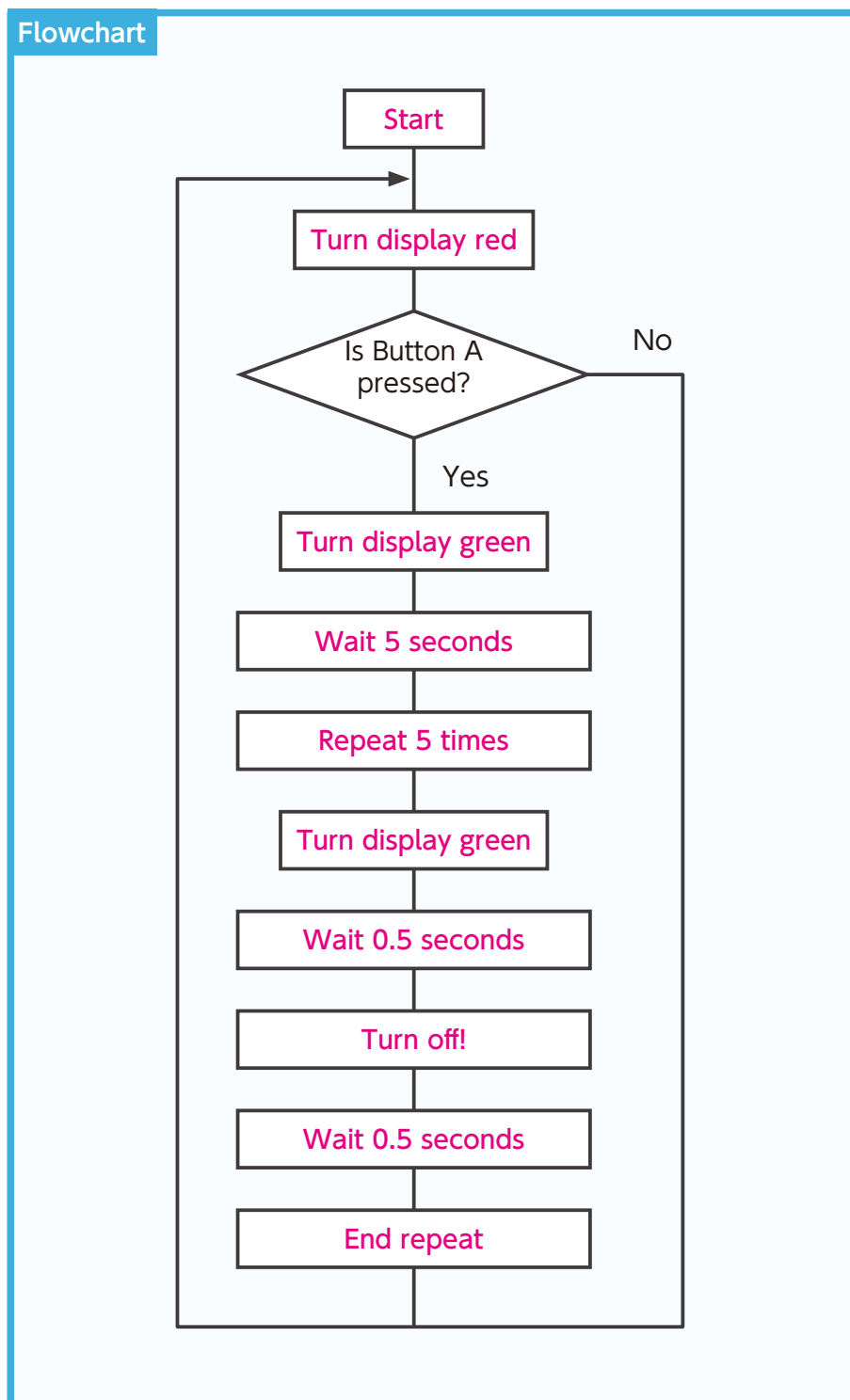


This time, let's add the condition "Button A is pressed" to the flowchart. Use the following symbol to add a condition.





Fill in the flowchart below, take a photo, and try generating the program.



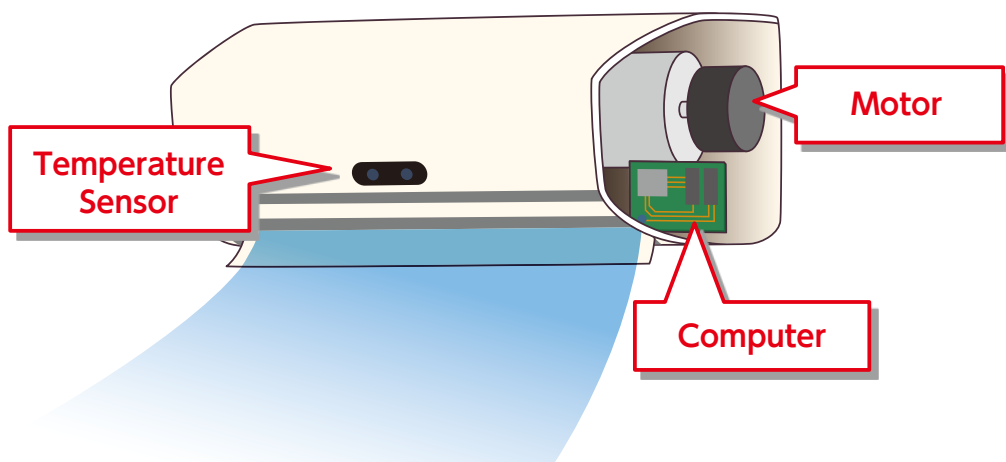
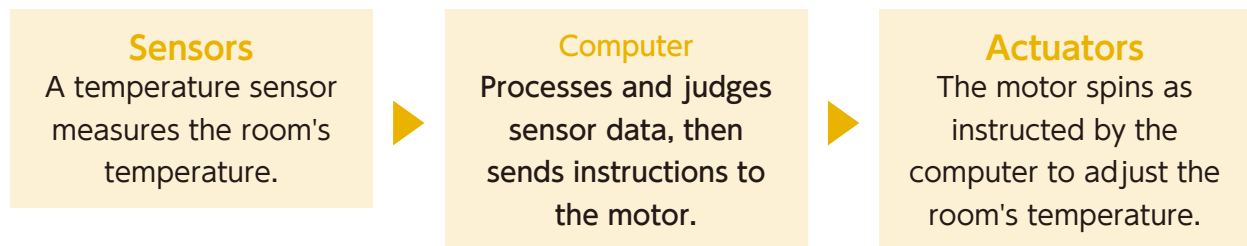
Doing different things depending on whether a condition is met is called **conditional branching**. Combining this with the sequential processing and iterative processing you've learned so far lets you build complex programs.

# [Chapter 5] Programming with Sensors

## ◆ 5-1. Electronic Parts

The electronics around us are made up of three elements: sensors, computers, and actuators. Electronics can automatically repeat a set action or flexibly respond to outside information because they use sensors and computers.

### Example How an Air Conditioner Works



### Try it Out Investigating a Familiar Product

Find a familiar product that uses sensors and computers, and investigate how it works.

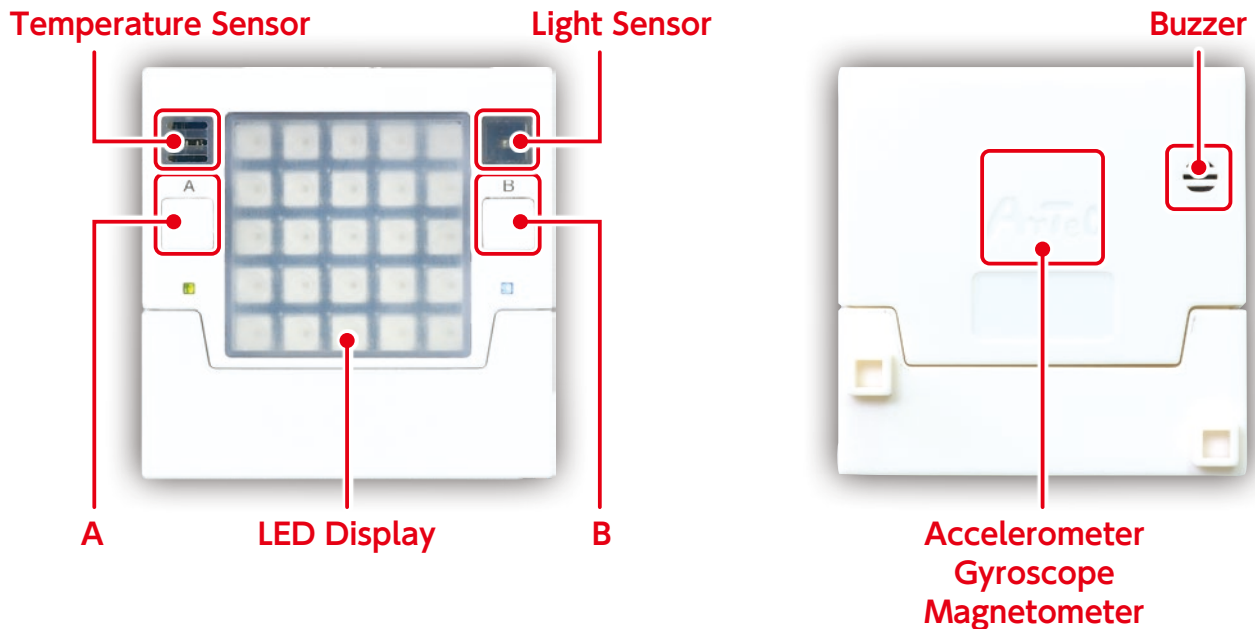
| Product Name:    | What the Sensor Measures           | How the Computer Controls It                                                               |
|------------------|------------------------------------|--------------------------------------------------------------------------------------------|
| Air Conditioning | Temperature                        | Adjusts the airflow and temperature to reach the set temperature.                          |
| Butlerbot        | Whether There's a Wall or Obstacle | When it detects a wall or obstacle, it changes the motor's direction to avoid a collision. |
| Security Light   | A Person's Movement                | When it detects a moving person, it lights up to warn them.                                |

## ◆ 5-2. ArtecRobo 2.0 Parts

The ArtecRobo 2.0 has all sorts of built-in parts beyond the LED display and buttons you've already used. You can find out what each part does starting on page 50.

Front

Back



Sorting these parts into sensors, computers, and actuators looks like this.

| Sensors                                                                                      | Computer               | Actuators             |
|----------------------------------------------------------------------------------------------|------------------------|-----------------------|
| B Button<br>Temperature Sensor<br>Light Sensor<br>Accelerometer<br>Gyroscope<br>Magnetometer | Studuino:bit Core Unit | LED Display<br>Buzzer |

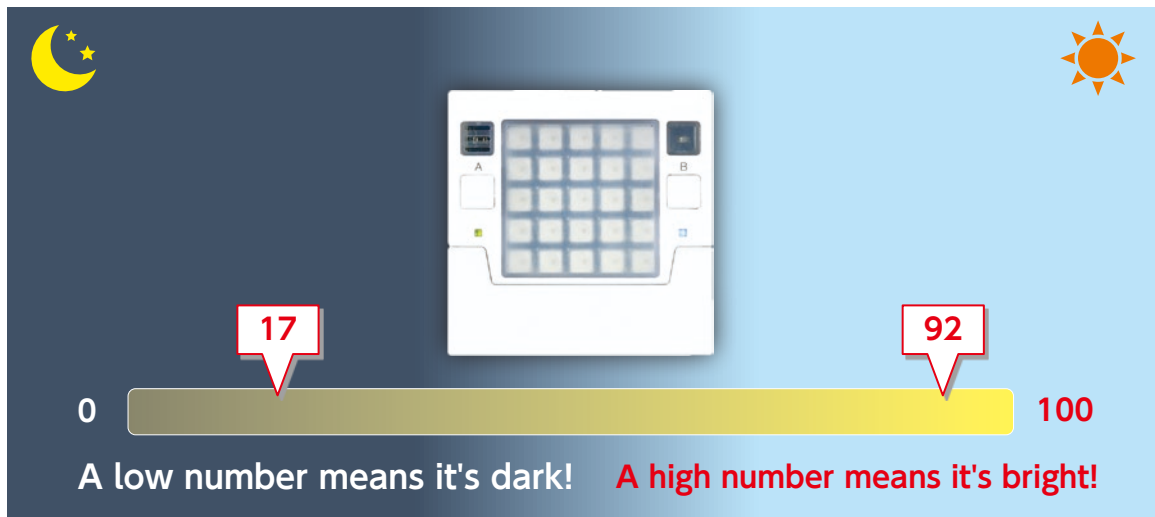
Setting these sensors and actuators under Parts to Use in Artec Intelligence lets you make programs that measure your surroundings or move using them. Starting on the next page, let's freely use these parts to make even more complex programs.

### ◆ 5-3. The Light Sensor

Here, let's use the light sensor to make a program that changes the LED display depending on how bright it is.

#### ! What's a Light Sensor?

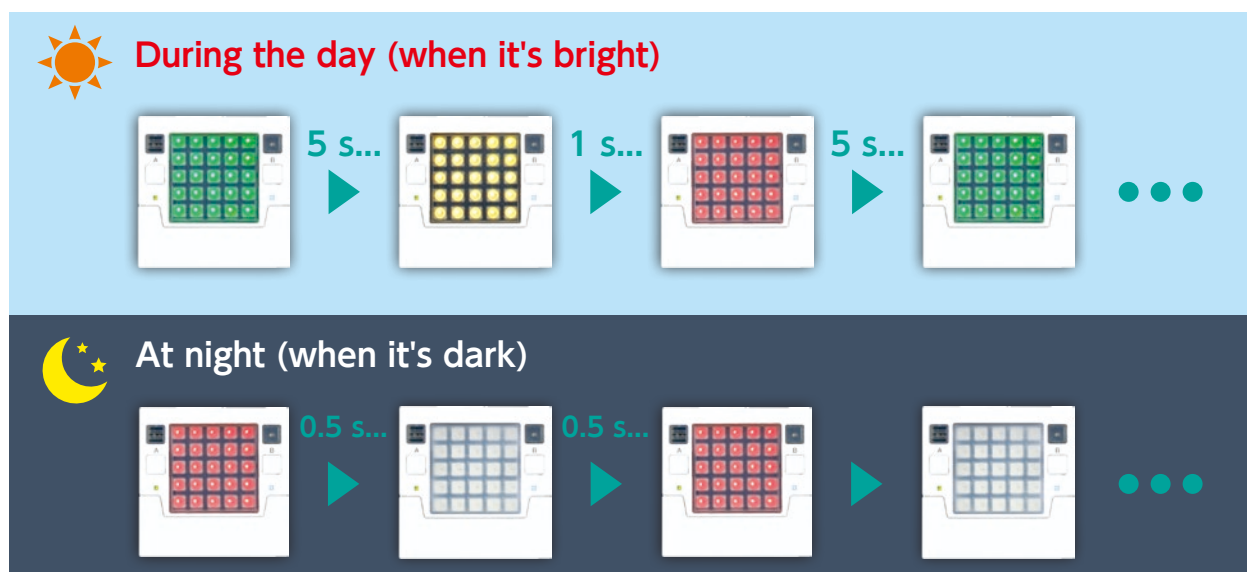
A light sensor measures how bright it is around it. The ArtecRobo 2.0 has a light sensor built in above the B button, which reads the surrounding brightness as a number from 0 to 100.



#### Try it Out Making a Nighttime Traffic Signal

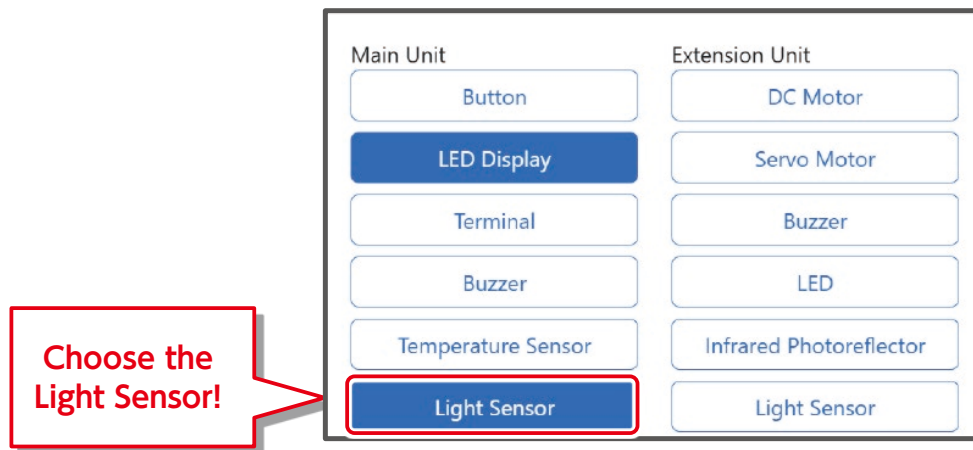
Let's use the light sensor to generate a program for a nighttime traffic signal. During the day (when it's bright), a nighttime traffic signal behaves like a normal signal. At night (when it's dark), the red light blinks every 0.5 seconds.

★ You can use just a text prompt, or fill in the flowchart on the right page instead.

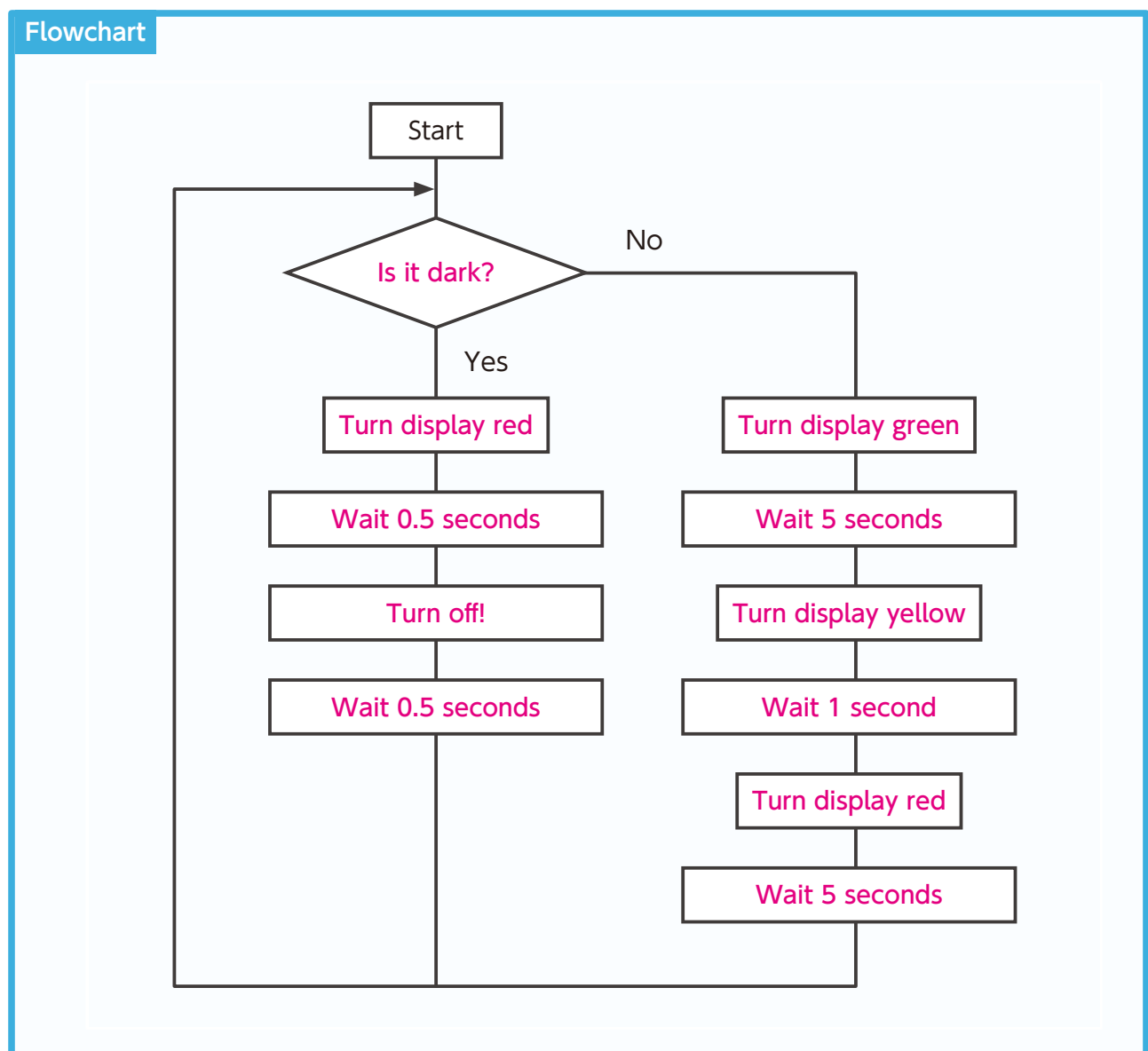


★ You can find example prompts on page 47.

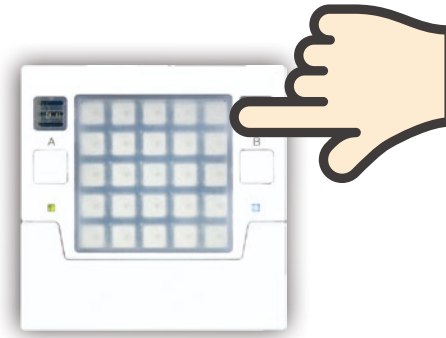
★ Add the light sensor under Parts to Use first.



★ Fill in the flowchart below, take a photo, and use it.



Cover the light sensor with your hand to change the brightness and check whether the behavior changes. If the behavior doesn't change, try adjusting the reference brightness in your prompt.



## Helpful Hints

The value used as the boundary for judging whether it's bright or dark is called a **threshold**. You need to adjust the threshold depending on your surroundings, but if you don't specify anything in your prompt, generative AI will just set an arbitrary value like 50. If it doesn't respond the way you expect, try including a specific threshold value in your prompt.

### The Prompt

**## Additional Conditions**  
**- Set the light sensor threshold to 30.**

★ That said, if you set the threshold too low, covering the sensor with your hand might not bring the value below it, so it won't behave as you expect.

[How to Check the Light Sensor's Value]

For example, you can use a prompt like the one below to make a program that displays the light sensor's value on the LED display for a set amount of time. You can use the value it shows to help decide on a threshold.

### The Prompt

**Display the light sensor value on the LED display when Button A is pressed.**  
**The display duration is 3 seconds.**

★ Add the Main Unit's Button under Parts to Use first.

#### The Prompt

Please write a program based on the flowchart.

## Conditions

- Red: Turn on all LEDs in red
- Yellow: Turn on all LEDs in yellow
- Green: Turn on all LEDs in green

(When using text only)

#### The Prompt

Please create a program for a traffic light that constantly measures the ambient brightness and changes its behavior depending on whether it is dark or light.

# Behavior in bright conditions

Continuously cycle through red, yellow, and green lights at intervals of 5 seconds, 1 second, and 5 seconds, respectively.

# Behavior in dark conditions

Flash the red light every 0.5 seconds.

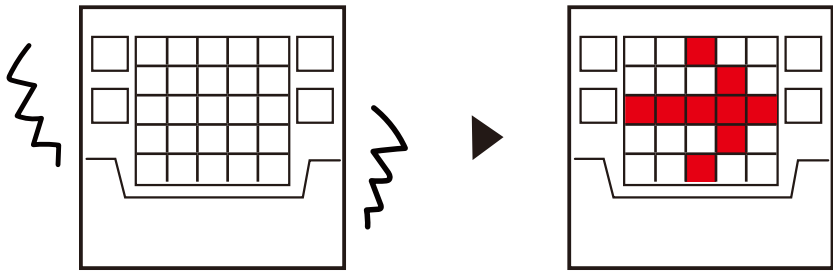
# Traffic light program

- red: Turn on all LEDs in red.
- Yellow Turn on all LEDs in yellow.
- green: Turn on all LEDs in green.

## ◆ 5-4. Solving Problems with Programs

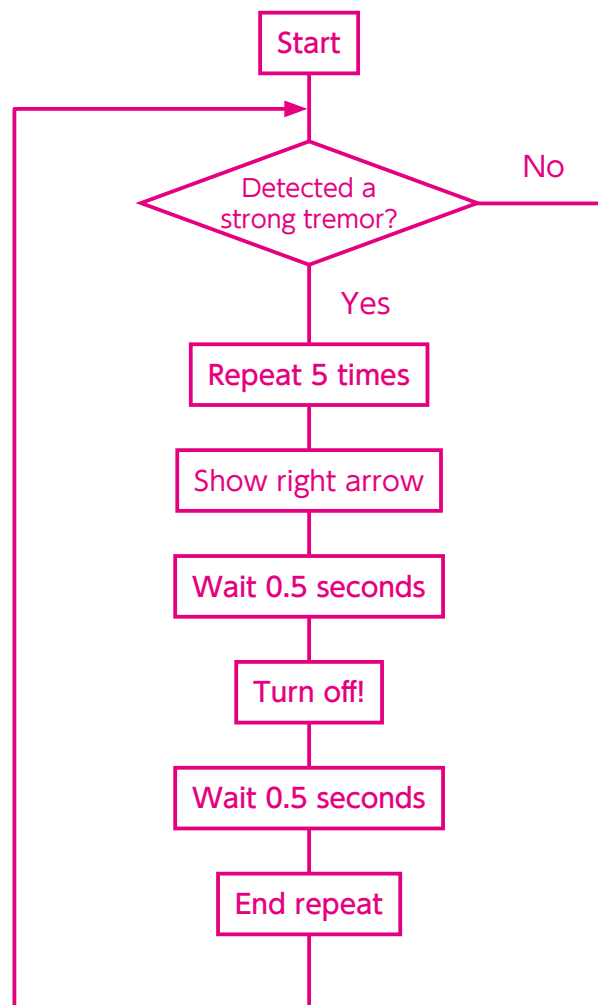
From here, let's use the ArtecRobo 2.0's sensors and actuators to think of ideas that solve problems around us. To bring your idea to life, use the prompts and flowcharts you've learned so far to tell Artec Intelligence what you want and generate a program.

★ For sensors and actuators not yet introduced, and how to write prompts for them, refer to the content starting on page 50.

|                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Idea Name            | Nighttime Emergency Evacuation Light                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Defining the Problem | When Does the Problem Happen?<br><br>When a major earthquake strikes at night and the power goes out.                                                                                                                                                                                                                                                                                                                                                     |
|                      | What inconvenience or danger is there if things stay the same?<br><ul style="list-style-type: none"> <li>• If the power goes out at night, the inside of the house and evacuation routes like hallways and stairs go completely dark. It's dangerous because you can't tell where to escape.</li> <li>• People might panic and not being able to see your feet means people might get hurt bumping into furniture or stepping on broken glass,</li> </ul> |
| How Does it Work?    | <p>Make a light that automatically turns on when it detects strong shaking from an earthquake, and shows a safe direction to evacuate with an arrow.</p>                                                                                                                                                                                                              |
| Parts                | Accelerometer, LED Display                                                                                                                                                                                                                                                                                                                                                                                                                                |



## Flowchart



## The Prompt

Please display a right-pointing arrow on the LED display when an earthquake occurs.

## Buzzer

It can play electronic tones at different pitches (notes). You can also specify note names or the title of a simple song in your prompt to have it play a tune.

Choose the Buzzer!

### Parts Selection

Main Unit

Button

LED Display

Terminal

Buzzer

## Example



### The Prompt

Please play the notes "Do, Re, Mi" in order.

### The Prompt

Please play "Twinkle, Twinkle, Little Star" on the buzzer.

## Temperature Sensor

This sensor measures the temperature in the area! It can measure temperature to two decimal places.

★ Using parts for a long time may cause the heat from the parts to raise the temperature detected by the sensor!

Choose the Temperature Sensor!

### Parts Selection

Main Unit

Button

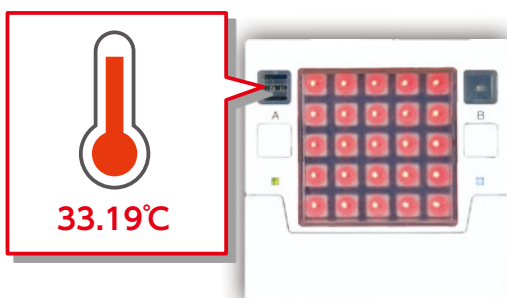
LED Display

Terminal

Buzzer

Temperature Sensor

## Example Temperature Alert



### The Prompt

Please light up green when the temperature is below 25°C, yellow when it is below 30°C, and red when it is 30°C or higher.

## Accelerometer

“Acceleration” is a number that shows how much speed changes over a set period of time. It can measure changes when the Main Unit moves or tilts.

Choose the  
Accelerometer!

### Parts Selection

Main Unit

Button

LED Display

Terminal

Buzzer

Temperature Sensor

Light Sensor

Accelerometer

### Example Earthquake Alert Light



### The Prompt

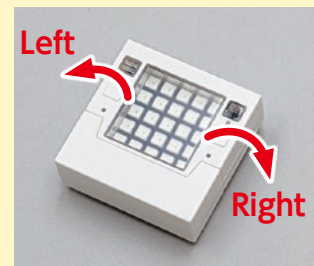
If the main unit is shaken, please have it light up in yellow.  
When it is not being shaken, please have it light up in white.

## Helpful Hints

When making a program that uses the accelerometer, you can easily ask by specifying left or right in your prompt.

### The Prompt

If the main unit is tilted to the right, the light will turn green.  
If the main unit is tilted to the left, the light will turn white.



## Gyroscope

The angle something rotates in one second is called “angular velocity.” By measuring angular velocity, the gyroscope can detect the Main Unit's rotational movement, which the accelerometer can't measure.

Choose the Gyroscope!

|                    |
|--------------------|
| Main Unit          |
| Button             |
| LED Display        |
| Terminal           |
| Buzzer             |
| Temperature Sensor |
| Light Sensor       |
| Accelerometer      |
| <b>Gyroscope</b>   |
| Magnetometer       |

## Example Rotation Detector



### The Prompt

When the Studuino:bit is rotating, please make the LED light up red.

## Magnetometer

It's a sensor that measures magnetism. It can measure changes in the surrounding magnetic field.

Choose the Magnetometer!

|                     |
|---------------------|
| Main Unit           |
| Button              |
| LED Display         |
| Accelerometer       |
| Gyroscope           |
| <b>Magnetometer</b> |

## Example Magnet Detector

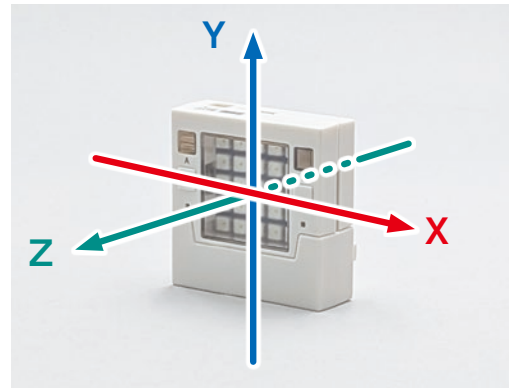


### The Prompt

If there is a magnet nearby, please sound the buzzer.  
If there is no magnet nearby, please stop the buzzer.

## ! The X, Y, and Z Axes

The ArtecRobo 2.0's accelerometer, gyroscope, and magnetometer each measure along three axes: X, Y, and Z. Combining the values from all three directions lets it calculate things like which way it's tilting or moving.



## [Chapter 6] Discussion Time

Based on what you learned in the basics textbook, let's have a discussion on the following topic.

[Topic] Why Learn to Program When You Can Prompt?

With Artec Intelligence, all you had to do was give an instruction (a prompt) in your everyday language, and generative AI would automatically write a Python program for you.

If generative AI keeps improving in the future and can write exactly the program we ask for, perfectly every time... Will humans still need to study how programs work, or read and write them at all?

First, look back on what you've learned so far and organize your own opinion. Then, share your opinions with each other and put together your group's opinion. Finally, compare each team's presentation and refine your own answer to this topic even further.

My Opinion (Before Discussion)





### For Facilitators

Please split students into groups of 3 to 5 for this discussion.

#### ● Expected Student Opinions

- If AI can write everything, humans don't need to be able to read or write code.
- You can't fix it when the AI gets something wrong, so it's better to be able to read and write it yourself.

#### ● Points to Help Students Realize

- The debugging perspective  
AI makes mistakes. Without the ability to understand what's wrong, you can't give the AI a prompt to fix it.

- The flowchart perspective

The way you build a process's steps and conditions is exactly the same whether you're writing a program or giving a prompt to AI.

- How the human role is changing  
Going forward, humans' work will shift from "typing out code" to "thinking about the goal" and "evaluating and correcting the AI's answers."

#### ● Example Conclusion for the Lesson

In the future, it may no longer be necessary to memorize code and type it by hand. But it's still humans who give the AI the right instructions, check for hallucinations, and take responsibility. That's exactly why learning how programs work and how to build logic will be the most important skill going forward.

## Other Groups' Opinions





Unauthorized reproduction  
and distribution prohibited



**Artec Intelligence**  
**Artec Intelligence**  
**[The AI Basics]**  
**Facilitator's Edition**