

www.nanoskool.com

**NANO**  
nanoskool.com **SKOOL**  
Imagine. Design. Create.



# STEAM ENCYCLOPEDIA

SAMPLE  
BOOK  
NOT FOR SALE

SCIENCE  
TECHNOLOGY  
ENGINEERING

ARTS  
MATHEMATICS  
ROBOTICS

CODING  
STEAM EDUCATION  
AUTOMATION



# Learning Mission to

# South Korea 2023



## NanoSkool's Maiden Mission to South Korea Pioneering AI and STEAM in Education



NanoSkool's first Learning Mission to South Korea concluded successfully, bringing together Indian education leaders for an immersive journey through cutting-edge AI and STEAM-integrated education. Highlights included visits to AI-integrated schools, workshops led by Luxrobo at LG Sciencepark, and a deep dive into the future of adaptive AI classrooms inspired by South Korea's National Education Policy.

The mission was driven by visionary leaders like Senthil Kumaran, and emphasized the transformational role of AI in K-12 education. Sang Hun Oh's leadership was celebrated for modernizing South Korean education through computational thinking and innovative AI solutions, showcasing a global model for education technology.

**Inviting Educational Leaders to  
Learning Mission South Korea**



**+91 97786 65473**  
E-mail: [info@nanoskool.com](mailto:info@nanoskool.com)

**BOOK NOW**





EMPOWERING  
SCHOOL LEADERS &  
SCIENCE TEACHERS



# A Transformative Event at IISc Bangalore



On June 20, 2024, the Indian Institute of Science (IISc) Bangalore hosted the IGNITE - A STEAM Workshop, co-organized by NanoSkool and the Indian Development Foundation (IDF). The event aimed to promote Science, Technology, Engineering, Arts, and Mathematics (STEAM) in K-12 education. Notable speakers included Padmashri Dr. Y. S. Rajan, a key figure in India's space program, and Dr. A. P. Jayaraman, a nuclear scientist known for popularizing science through storytelling.

Dr. Rajan emphasized the need for integrating STEAM into curricula to prepare students for the technological future, while Dr. Jayaraman inspired educators with his storytelling-based approach to teaching science. Rajesh George, CEO of NanoSkool, introduced the STEAM Encyclopedia, a new robotics and coding curriculum integrated with Luxrobo's Modi kits for hands-on learning.

The event brought together educators and students to explore innovative approaches in education, emphasizing creativity, problem-solving, and real-world applications.



**nanoskool.com SKOOL STEAM Encyclopedia Launched**

Nanoskool and Luxrobo have launched new STEAM Robo-Coding textbooks for students in grades 1-10, focusing on creativity, problem-solving, teamwork, and logical thinking. The curriculum integrates Luxrobo's Modi robotics kits, allowing students to build and program robots through hands-on activities. This partnership ensures students gain both theoretical and practical STEAM knowledge, preparing them for a technology-driven future.



GRADE

5

PROJECT

1

## ORIENTEERING IN THE SILK ROAD ERA: ANCIENT NAVIGATION TECHNIQUES



### What You Will Learn

- How ancient traders and explorers navigated the vast Silk Road.
- The role of orientation and direction in historical navigation methods.
- The functionality of MODI's Network, Display, and IMU modules in creating a navigation tool.
- Programming concepts like "while" loops and their application in continuous monitoring tasks.
- The process of turning a conceptual idea into a working digital project using MODI.

### Learning Objectives

- Understand the basics of ancient navigation and its significance in historical trade routes.
- Learn to assemble and program a digital navigation tool using specific MODI modules.
- Develop skills in applying programming concepts, particularly in creating responsive loops.
- Enhance problem-solving and logical thinking through coding and debugging digital projects.
- Bridge historical knowledge with modern technology by reimagining ancient tools through digital simulation.





Traveling on the Silk Road was no easy feat. It spanned vast and diverse landscapes, from the deserts of Central Asia to the mountains of the Himalayas and the forests of Europe. To navigate this challenging terrain, travelers relied on a combination of natural observations and cultural knowledge. One of the ways travelers found their way was by observing the natural environment. They looked to the sun, moon, and stars to determine directions, using their positions in the sky to navigate eastward or westward. The rising and setting of the sun provided valuable cues for direction during the day, while the stars offered guidance during the night.



Additionally, travelers sought assistance from local inhabitants and fellow travelers along the route. They asked for directions, traded information about safe paths and water sources, and shared stories of their travels. Experienced caravan leaders, who were familiar with the route, often guided the way, ensuring the safety of the caravan and its cargo.

Through collaboration and the sharing of knowledge, travelers successfully traversed the Silk Road, overcoming the challenges of long distances, harsh climates, and diverse cultures. Along the journey, they not only traded goods but also exchanged languages, religions, technologies, and ideas, shaping the course of history and fostering cultural exchange between East and West.



## **Inquiry & Reflection**

### **Discussion question 1**

How do you think ancient traders knew which way to go without modern maps or GPS while traveling on the Silk Road?

### **Discussion question 2**

Why was being able to navigate well so important for making friends and trading between faraway places in ancient times?





## UNIT 2 BRINGING IDEAS TO LIFE

### Project Introduction

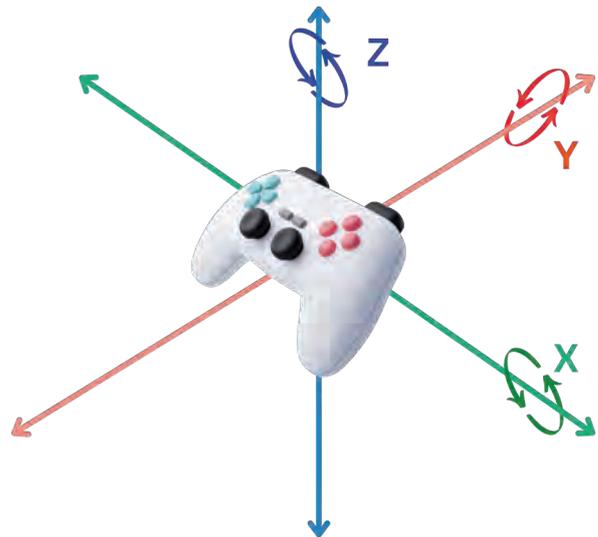
Imagine you're the captain of a ship in an adventurous video game, steering through treacherous waters by tilting your device. This tilting action, whether forward, backward, or side to side, is similar to the x, y, and z angles we learned about.

Suppose you want your ship to race ahead, you tilt your device forward. The game detects this tilt using the x angle, propelling your ship forward. Tilting it back slows your ship down, as the game interprets this as a negative x angle.

Now, imagine navigating your ship through narrow passages. Tilting your device to the left steers your ship left, thanks to the y angle. Tilting it right does the opposite, guiding your ship through the right path.

For sharp turns or dodging obstacles, you might twist your device. This action is picked up by the z angle. A twist to the left makes a swift left turn, and a twist to the right sends your ship darting right.

In our "MODI Compass System" project, we'll use an IMU module, similar to the technology in your game, to understand movement and direction. Just like in the game, the IMU module will help our creation sense which way it's facing, making our compass respond to the world around us, guiding us on our journey just like ancient travelers on the Silk Road!



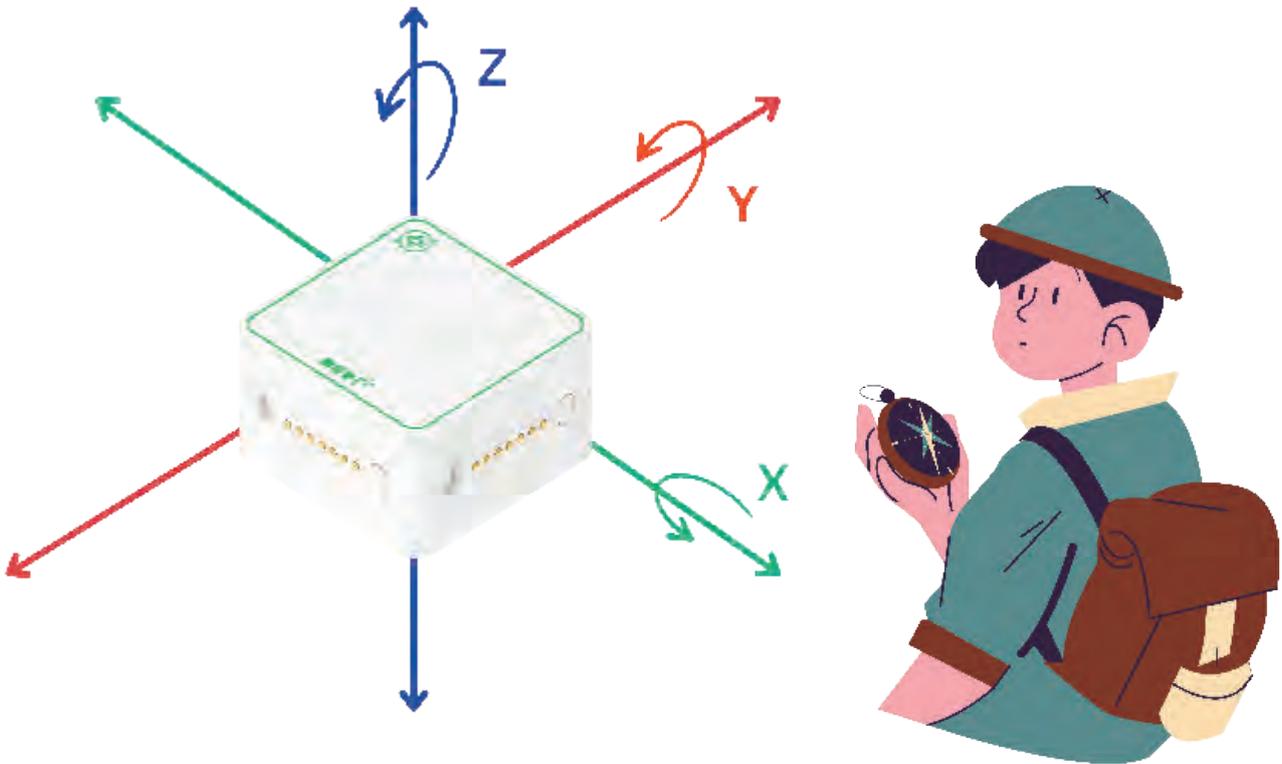
### TIPS

IMU modules are like the secret agents inside our devices, helping them understand how we're moving and making our interactions with technology feel more intuitive and fun!



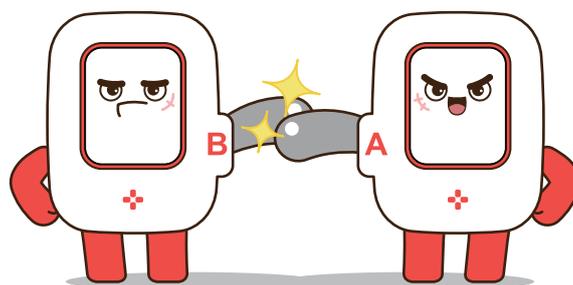
### MODI COMPASS SYSTEM (BASIC)

Our MODI Compass System keeps a close eye on how much the IMU module tilts using something called a "while" loop in its program. When the IMU tilts a lot (more than 10 degrees or less than -10 degrees), the screen shows a tired face, meaning it's not quite balanced.



But, if the IMU stays almost flat, with a tilt between -10 and 10 degrees, you'll see a happy, smiling face on the screen. This part of our project uses just the Z angle of the IMU to show us how a compass works, teaching us about staying balanced and finding our way, much like ancient explorers did.

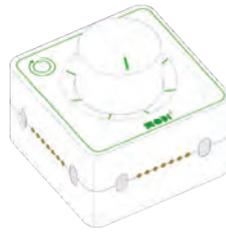
From now on, let's create a digital compass system with MODI modules that shows the correct direction to head!





## Required Modules & Brainstorming

Based on the story, which MODI modules do you need for this creation?



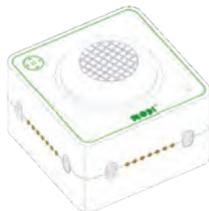
Dial



Button



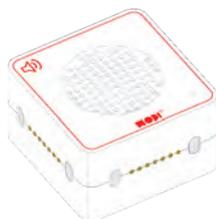
Environment



Joystick



ToF



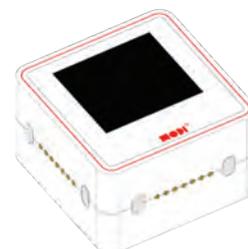
Speaker



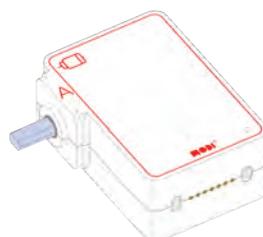
IMU



LED



Display

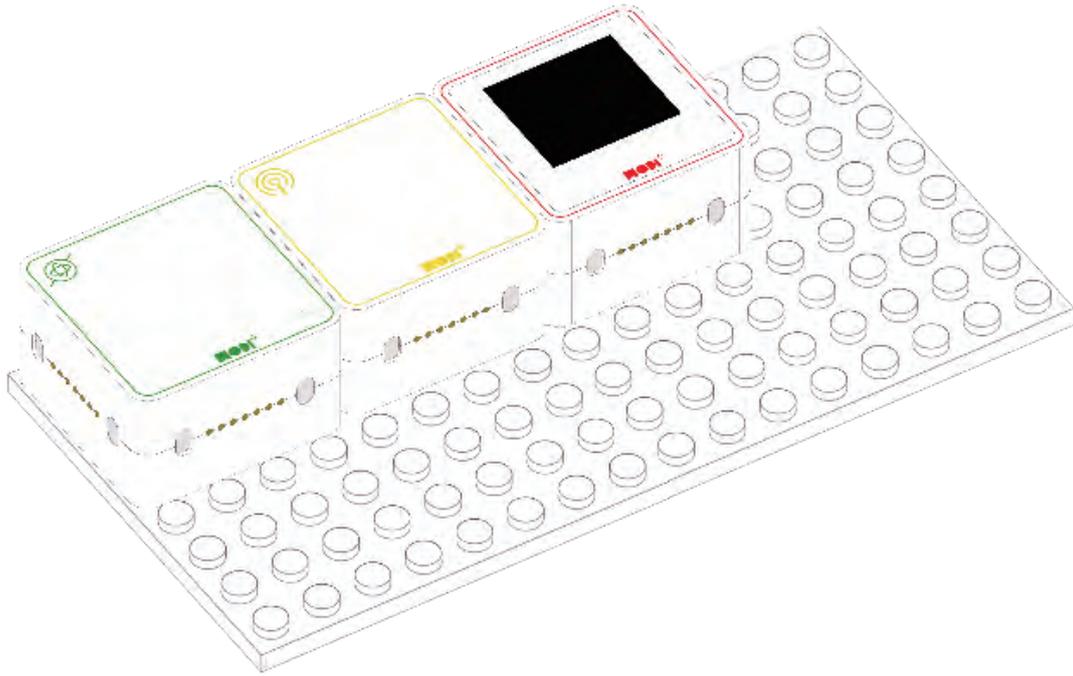


Motor A



Motor B

HOW DOES THE CREATION LOOK LIKE?





## Module Functionality

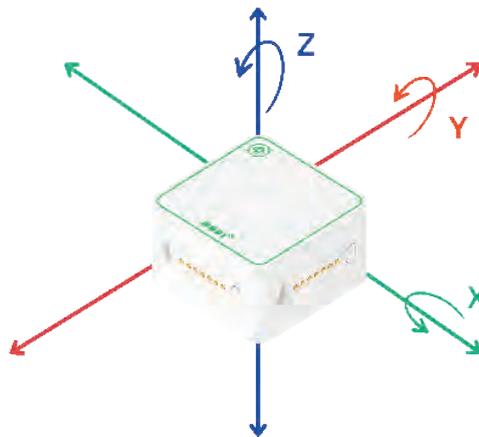
Let's check the details of the input module below for this project



- ✓ x angle(°)
- y angle(°)
- z angle(°)
- x acceleration(%)
- y acceleration (%)
- z acceleration(%)
- x angular velocity (%)
- y angular velocity (%)
- z angular velocity (%)
- vibration (%)

### IMU

- **X, Y, Z Angle:** Shows how much the module is tilted.
- **X, Y, Z Acceleration:** Tells how fast the module moves in any direction.
- **X, Y, Z angular velocity:** Shows how quickly the module spins around.
- **Vibration:** Detects any shaking or vibrating motion.



<p><b>1. X, Y, Z Angle</b></p>	<p><b>2.Vibration:</b></p>
<p>Rotation angle measurements around each axis.</p> <p>Range: -180° ~ 180°</p>	<p>To measure it, there are 3 important parameters — acceleration, velocity and displacement.</p> <p>Range: 0% ~ 100%</p>

**TIPS**  
IMU can receive 4 types of inpu data





## UNIT 3 DESIGNING SOLUTIONS

### Programming Concepts

#### WHILE STATEMENT

# ✓ WHILE



#### condition

While the timer is running,



#### task

Keep brushing your teeth.

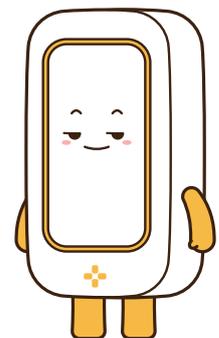
Imagine you're brushing your teeth with a timer set to make sure you brush long enough. A while loop in coding is like saying, "**Keep brushing your teeth while the timer is still running!**"

You continue brushing, and each moment, you check if the timer is still going. As long as the timer hasn't run out, you keep brushing. But once the timer stops, you stop brushing too. That's how a while loop works—it keeps doing an action (like brushing your teeth) as long as a certain condition (the timer running) is true.

#### TIPS

While loop keeps repeating an action as long as a condition is satisfied. In real life, a while loop can be compared to tasks such as:

"**Keep doing assignments while there are assignments left in your 'to-do' list.**"





**Example**

- 1 We are spooking a tired MODI, when we are near by the MODI, it's surprised. Find the condition and the action(task) in the code.

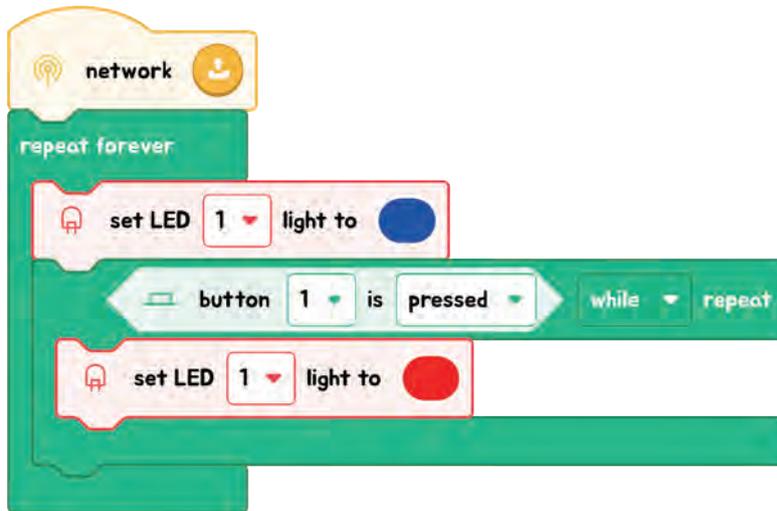
The image shows a Scratch script starting with a 'network' block, followed by a 'repeat forever' loop. Inside the loop, there are three blocks: 'show drawing' (Tired, on display 1), a 'while repeat' block (ToF 1, distance < 15 cm), and another 'show drawing' block (Surprised, on display 1). To the right is a MODI character with a surprised expression and a red hat.

Condition(s)

>>>

Task(s)

2 Draw a flowchart based on the code shared below:



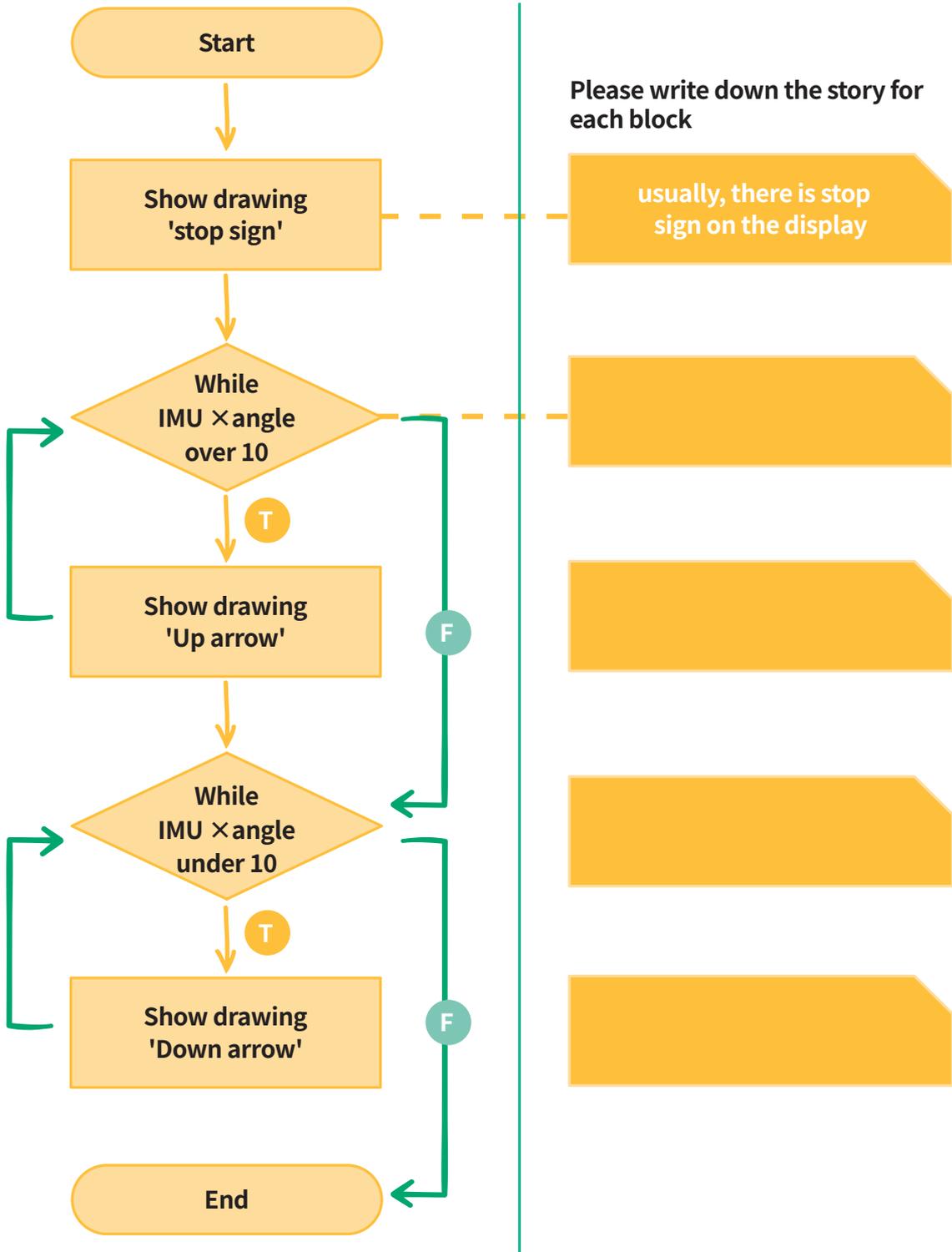
YOUR FLOWCHART



## Flowcharts & Algorithms

Let's read the flowchart and create the code.

### FLOWCHART





## UNIT 4 CODE QUEST

### Applying Knowledge (code & debugging)

#### CODE

```

network
repeat forever
  show drawing Stop Sign on display 1
  IMU 1 x angle > 10 while repeat
  show drawing Up Arrow on display 1
  IMU 1 x angle < -10 while repeat
  show drawing Down Arrow on display 1
  
```

#### DEBUGGING

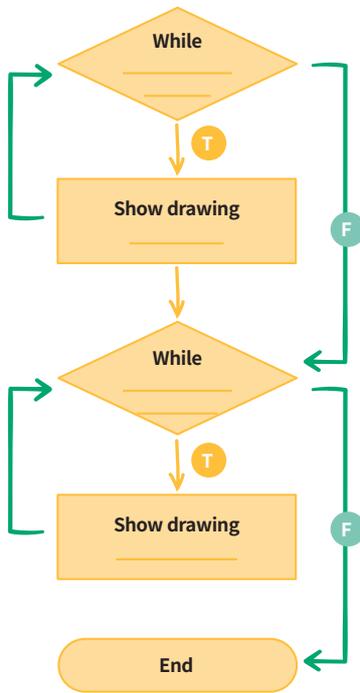
- 1 What if we use 'if statement' instead of 'while statement'?
- 2 Which block do we need to add 'while statement' for stop sign using 'and' operator?



## Missions

### Mission 1

1 Let's add right and left direction at the bottom of original code.



2 Based on the flowchart, make the code

**Mission 2****1** Let's make the LEDs light up depending on the direction

Direction	Color
Straight	ex) red
Back	
Right	
Left	

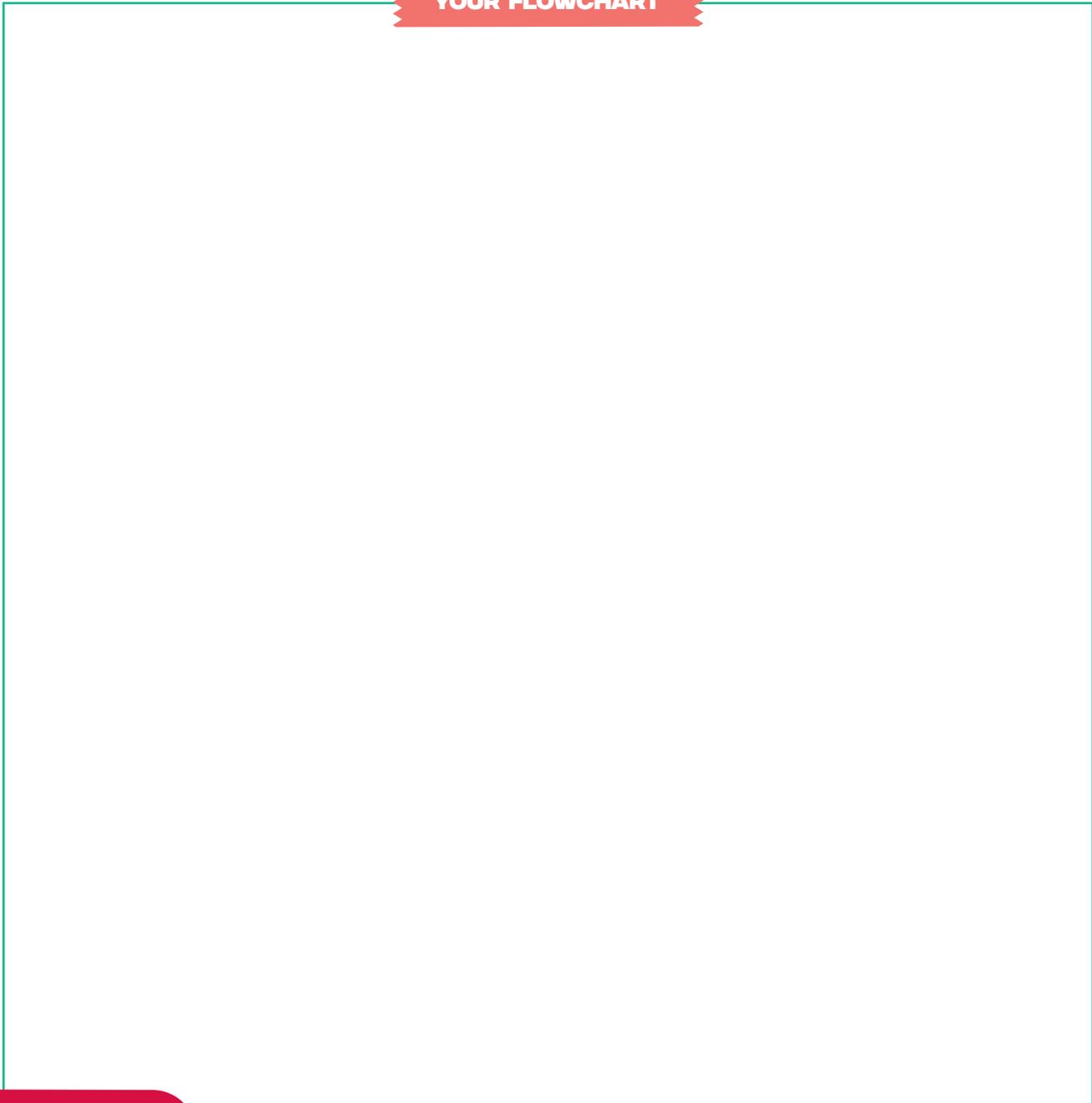


2 Based on the code, make the flowchart.

```

    show drawing Stop Sign on display 1
    set LED 1 light to red
    while IMU 1 x angle > 10 repeat
        show drawing Up Arrow on display 1
        set LED 1 light to green
    
```

**YOUR FLOWCHART**



# KNOWLEDGE CHECK

**1** Which direction did the Silk Road connect?

- A North and South
- B East and West
- C East and South
- D West and South

**CORRECT ANSWER**  B

**2** How did travelers find their way on the Silk Road?

- A By following signs
- B By using GPS
- C By asking for directions
- D By using natural landmarks and stars

**CORRECT ANSWER**  D

**3** What was the Silk Road?

- A A road made of silk
- B A historic trade route
- C A modern highway
- D A river in China

**CORRECT ANSWER**  B

**4** What is IMU can not detect?

- A Vibration
- B X angle
- C Y Angle velocity
- D Temperature

**CORRECT ANSWER**  D

**5** What is the purpose of a while statement in programming?

- A To check 2 or more conditions are met
- B To define a function in the program
- C To repeat code if a certain condition is true
- D To declare variables and assign values to them

**CORRECT ANSWER**  C

**6** Which of the following is NOT available with display module

- A Write a sentence
- B Show a picture
- C Show a number
- D Draw a picture

**CORRECT ANSWER**  D

**7** Which of the following is a function of the IMU??

- A Clicked
- B Vibration
- C Turn
- D Temperature

**CORRECT ANSWER**  B

**8** Which shape in a flowchart is typically used to represent an while statement?

- A Rectangle
- B Diamond
- C Circle
- D Parallelogram

**CORRECT ANSWER**  B



## Critical Thinking Challenge

---

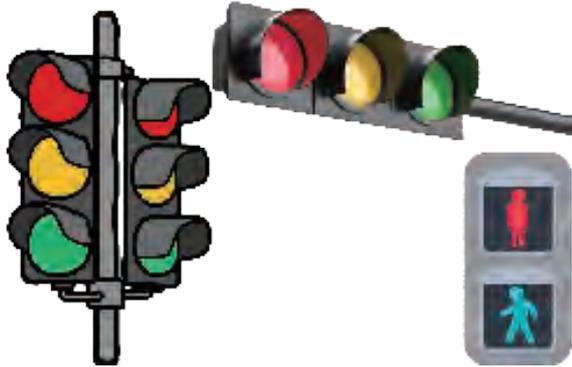
**1** What is difference between if statement and while statement?

**2** How Can we make navigation for the blind?



## UNIT 6 EXPANDING KNOWLEDGE ON THE TOPIC

### The History of Signposts: Guiding Humanity Through the Ages



Signposts have been an integral part of human history, dating back to ancient civilizations. They were primarily used to guide travelers and mark important locations.

Long ago, even before the times of ancient Rome and Greece, people used signposts. These weren't just any signs; they were made of stone and told travelers how far they had to go and which way to turn to reach different cities. Imagine walking through a city gate or into a village long ago and seeing a signpost, not just telling you where to go, but also keeping the place safe from danger.

As time went on and we invented more ways to get around, especially during the industrial revolution when lots of new machines and roads were made, signposts changed too. Now, they're super important for drivers to know the rules of the road and to stay safe. They also help tourists find fun places like parks and museums.

Today, no matter where we are, signposts help us. They show us where to go, help us in emergencies by pointing out exits and alarms, and make sure we're safe. Signposts have really become a big part of our everyday lives, changing and growing with us to make our world a better







GRADE

6

PROJECT

2

## EXPRESSING EMOTIONS WITH EMOJI



### What You Will Learn

- The origin and significance of emojis in digital communication.
- How to blend art and technology to create expressive emoji faces.
- The use of "wait" in programming to animate and time emoji expressions.
- The process of designing custom emojis that convey specific emotions.
- The role of emojis in enhancing digital art and storytelling.

### Learning Objectives

- Gain an appreciation for emojis as a form of digital art and communication.
- Develop skills in using display technology to create visual expressions.
- Understand and apply timing in programming to animate digital creations.
- Exercise creativity in designing personalized emojis with emotional depth.
- Explore how emojis can enrich digital narratives and artistic expressions.



## UNIT 1 EXPLORING THE CORE

### Topic Investigation

#### THE JOURNEY OF EMOJIS

#### THE ORIGINS OF EMOJI

Before emojis, we had emoticons like :-) and :( to show how we felt in messages. These simple symbols were a big deal in online chats back in the 1990s. In 1999, a creative guy from Japan named Shigetaka Kurita made the first emojis for something called "i-mode," which was a way to use the internet on your phone.

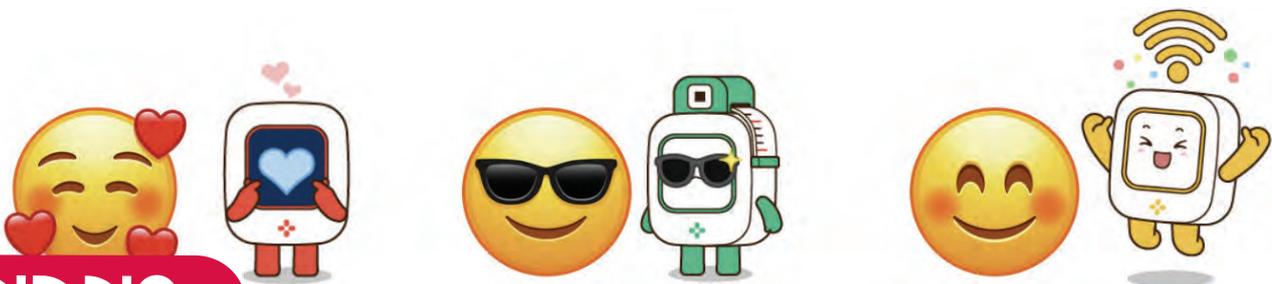


*Copyrights: NTT DOCOMO, INC.*

Kurita thought it would be cool to have little pictures instead of just words to show things like the weather or how you're feeling. He made tiny 12x12 pixel images, like little suns for sunny days or cars for traffic updates. These first emojis weren't just about faces but all kinds of symbols. This innovative visual language, now recognized by New York's Museum of Modern Art, underscored the blend of information and emotion in digital communication.

#### MORE THAN JUST FACES: EMOJIS AND EMOTIONS

Emojis aren't just about what they look like; they're about what they make us feel. A laughing emoji can make us smile, and a sad emoji can show sympathy. By choosing the right emoji, we can make our messages feel warmer and more personal.





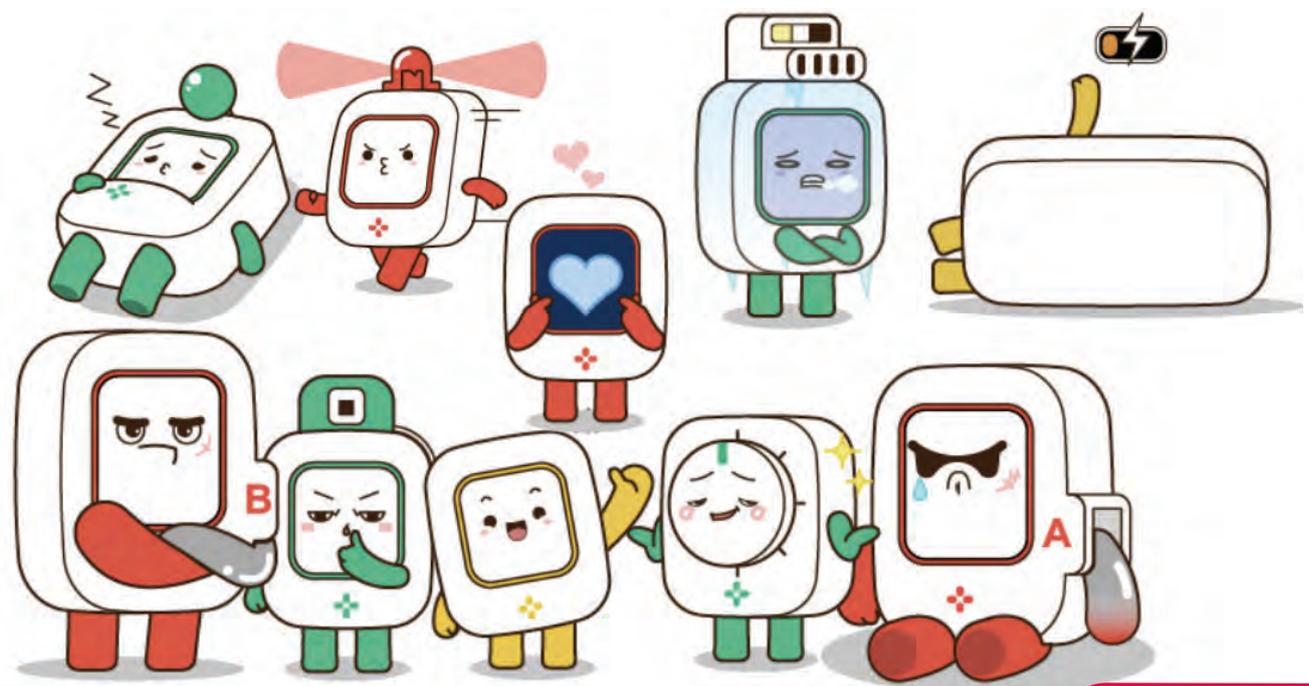
## EMOJIS TODAY: A NEW ART FORM

Today, creating with emojis is like a new kind of art. People even make whole pictures out of emojis! And with MODI, we're taking it a step further. We'll learn how to turn our own ideas into emojis and even make them change like magic to show different emotions.



## LOOKING AHEAD

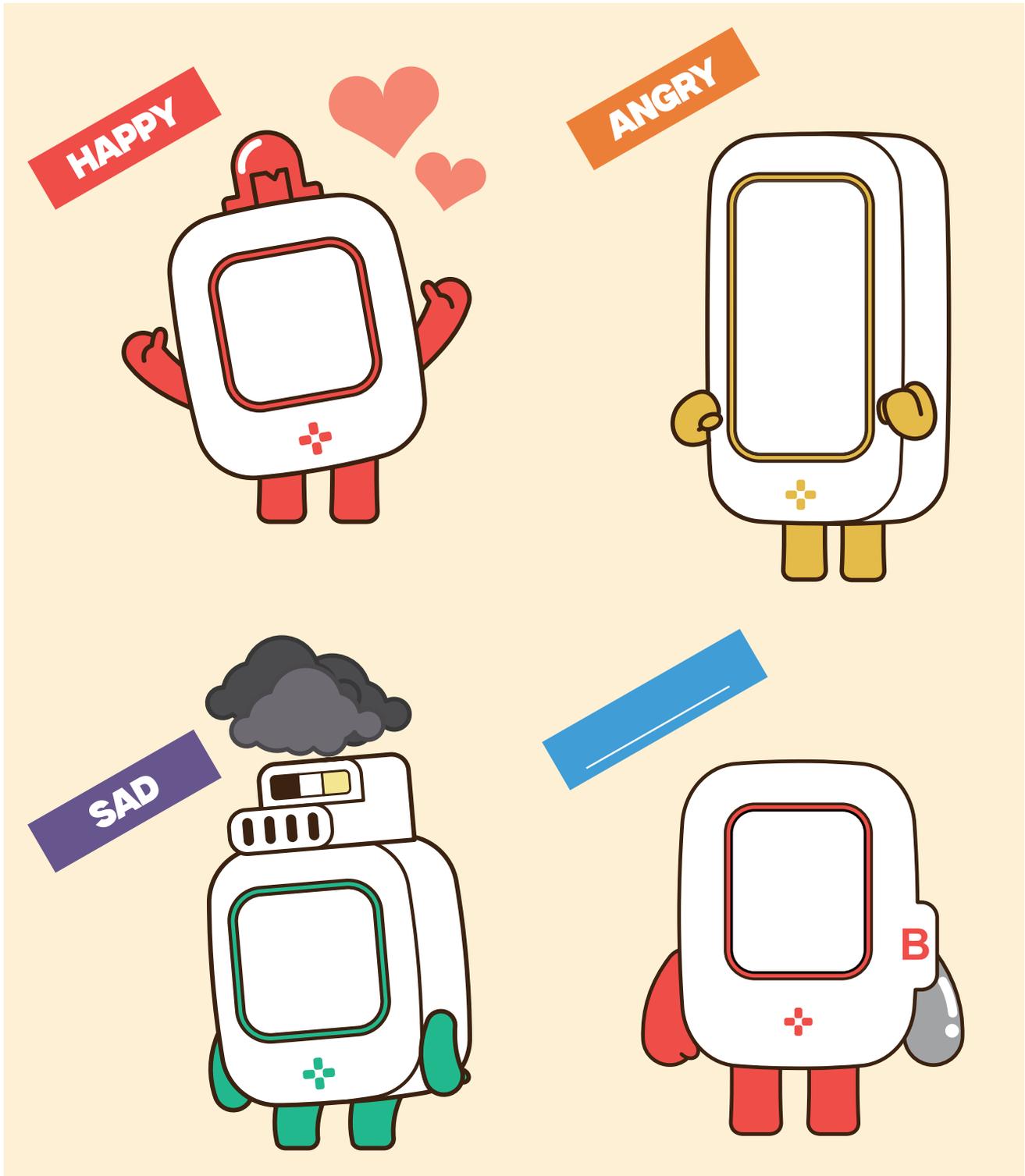
As we dive into making our own emojis, think about what emotions you want to share. What emoji would you invent to express something you feel that doesn't have an emoji yet? Don't worry! You have the MODI Master kit!



## Inquiry & Reflection

### Drawing Emojis

Draw a face for the Modi character. Give your character eyes, a nose, and a mouth for the given situation and express your mood. In the final section, create your own situation and facial expression.





## UNIT 2 BRINGING IDEAS TO LIFE

### Project Introduction

#### FACE EXPRESSION DEVICE FOR ALEX

In class, everyone's buzzing about the school science fair, and Cathy has a brilliant idea. She's noticed that their friend Alex is a bit shy and sometimes finds it hard to say how she's feeling, especially in big groups. Cathy thinks, "What if we could make something to help Alex express herself without having to say a word?"

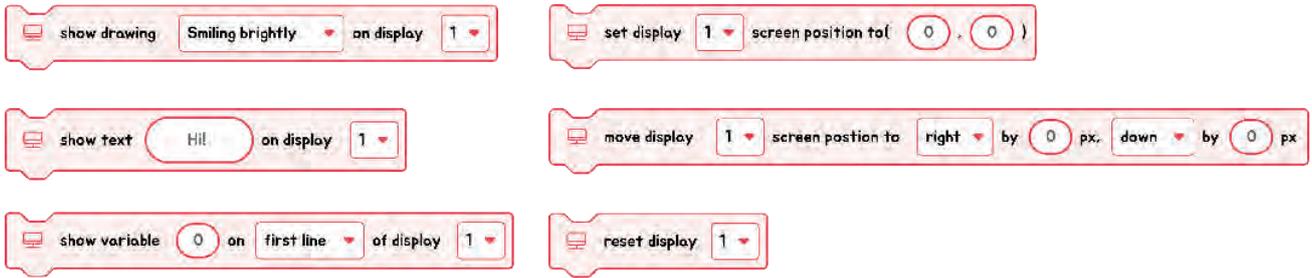


So, Cathy and her classmates decide to build a "Face Expression Device" using their MODI kit. They plan to design a gadget that can show different emojis based on how Alex is feeling. If Alex is happy, she can show a smiling emoji. If she's feeling a bit down or needs a break, he could choose a sad or a "tired" emoji.

Their goal is to give Alex a fun and easy way to share her mood with the class, making it easier for her to join in the conversation without feeling overwhelmed. Plus, they think it'll be a cool project that combines art, technology, and a bit of heart – exactly what the science fair is all about!



### ABOUT DISPLAY MODULE

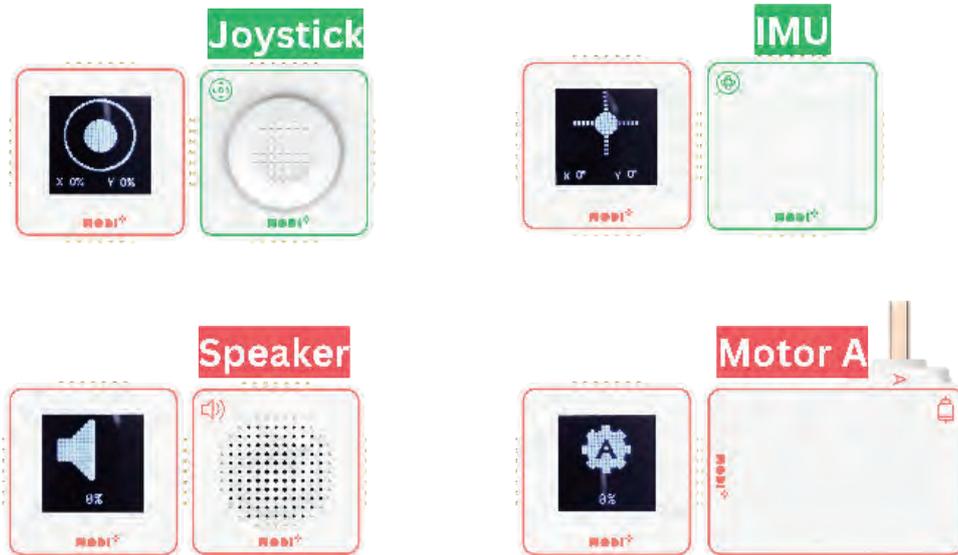


The Display module is like a tiny screen for your MODI projects. It uses 6 different blocks in the Code Editor to show pictures or information from other MODI modules nearby. So, when other modules talk to the Display, it can show you what they're saying or doing with images or data right on its screen!

### AUTOMATIC DATA DISPLAY WITH MODI MODULES

The Display module is like the heart of a conversation among MODI modules. Imagine it as a screen that lights up with information whenever other modules, like the Button or the Light Sensor, have something to "say." When any input or output module is connected to it, the Display module automatically picks up their signals and shows what they're doing on its screen.

### INTERACTION BETWEEN DISPLAY AND OTHER MODULES

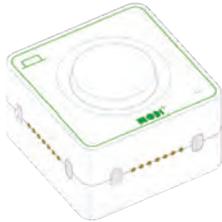


So, if you press a Button module or cover a Light Sensor, the Display will show these actions without needing any special programming. It's like having a translator that instantly shows you what's happening in the MODI module language!

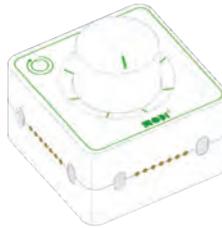


## Required Modules & Brainstorming

Based on the story, which MODI modules do you need for this creation?



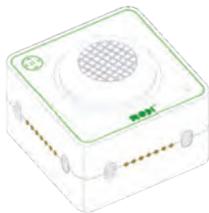
Button



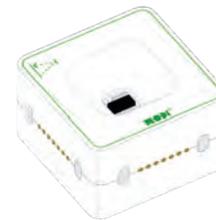
Dial



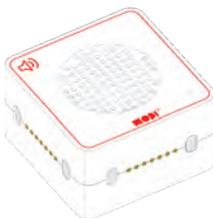
Environment



Joystick



ToF



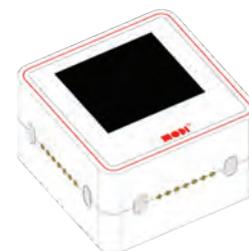
Speaker



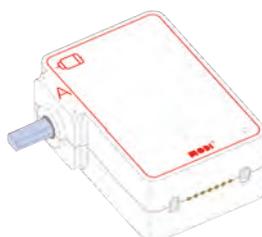
IMU



LED



Display

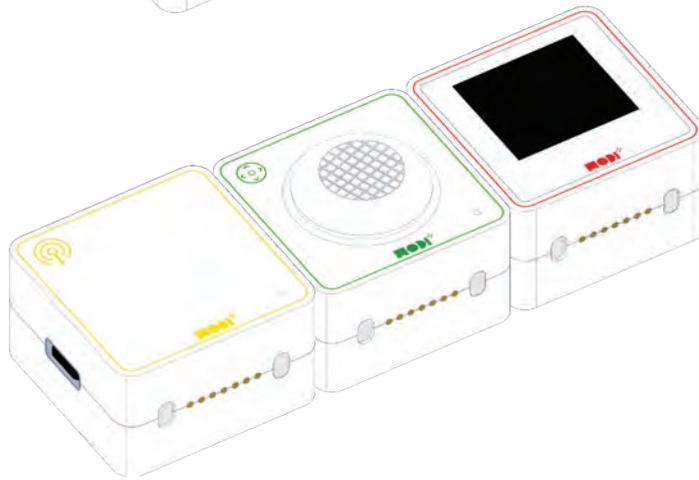


Motor A



Motor B

HOW DOES THE CREATION LOOK LIKE?





## Module Functionality

Let's look into the details of the Joystick module.



Including the neutral position 'origin,' the joystick has 5 different positions.



Pushing it upward and downward affects movement on the **y-axis**

- Range : -100 % (fully downward) ~ +100 % (fully upward)



Moving the joystick left and right controls movement along the **x-axis**



## UNIT 3 DESIGNING SOLUTIONS

### Programming Concepts

#### WHILE LOOP

A "while" loop continues to execute its block of code **as long as the condition is true**. It checks the condition before running the loop's code. If the condition is true, the loop runs. If it's false, the loop stops. For example, "while the sun is up, keep the LED off."

## ✓ WHILE



#### condition

While the sun is up,



#### task

Keep the LED off

#### Another Example:

"Think of using a 'while' loop like brushing your teeth while a timer is running. You start brushing, and 'while' the timer hasn't beeped (meaning time is not up yet), you keep brushing. As soon as the timer beeps, signaling the time is up, you stop brushing."



#### condition

While the timer is running,

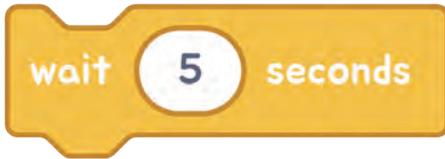


#### task

Keep brushing your teeth.

**WAIT**

The wait block is like a pause button for your code. It stops the code from moving on to the next step for a while. So, if a motor is spinning or an LED is shining, they'll keep doing their thing during the pause. The wait block just makes sure the code waits before jumping to the next action.

**DID YOU KNOW?**

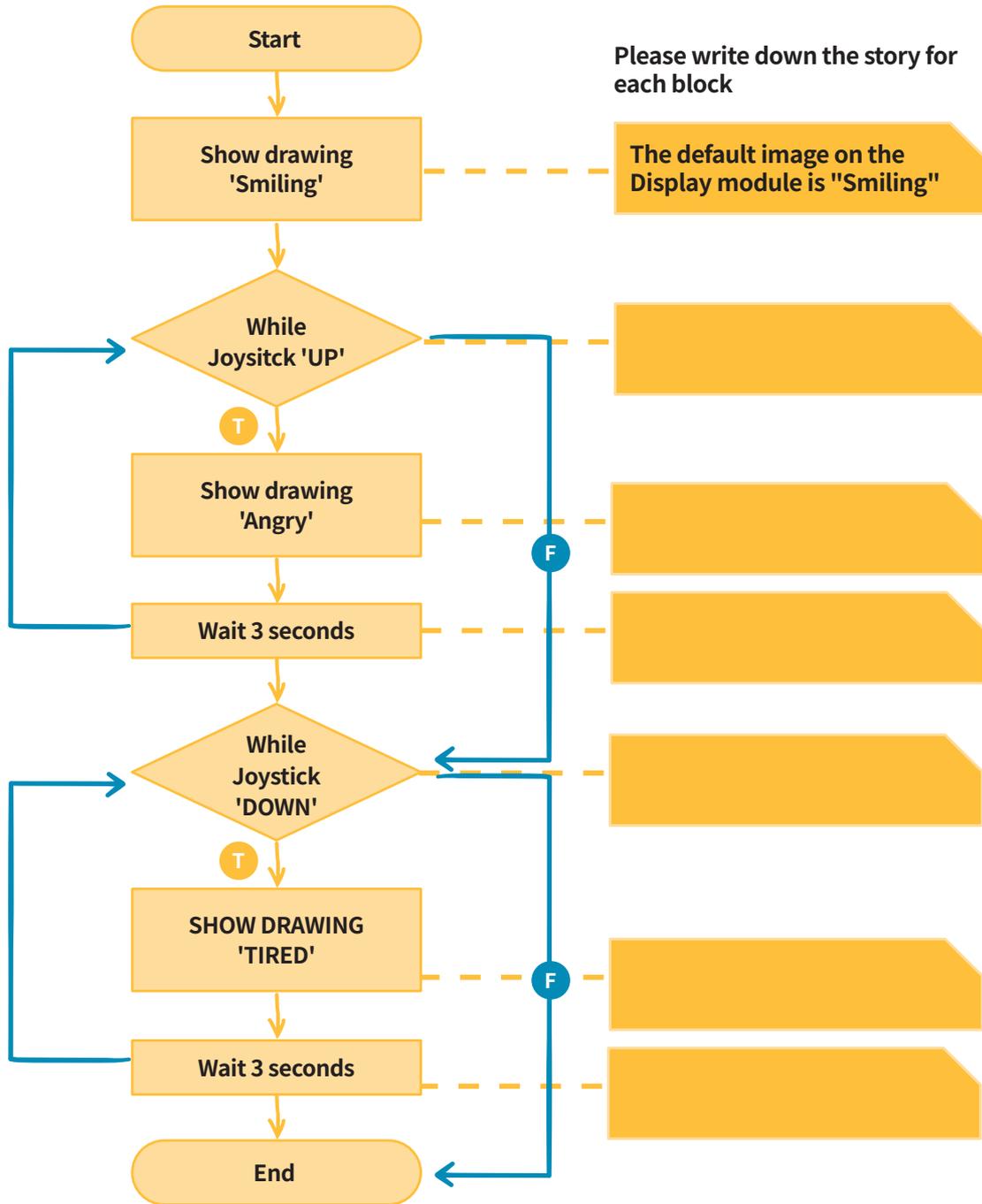
When using the wait block, it's important to remember: while the wait block is running, no other commands are executed. No buttons can be pressed, no dials can be turned, and all the code will stop



## Flowcharts & Algorithms

Let's read the flowchart and create the code.

### FLOWCHART





## UNIT 4 CODE QUEST

### Applying Knowledge (code & debugging)

#### CODE

```

network
repeat forever
  show drawing Smiling on display 1
  joystick 1 is up while repeat
  show drawing Angry on display 1
  wait 3 seconds
  joystick 1 is down while repeat
  show drawing Tired on display 1
  wait 3 seconds
  
```

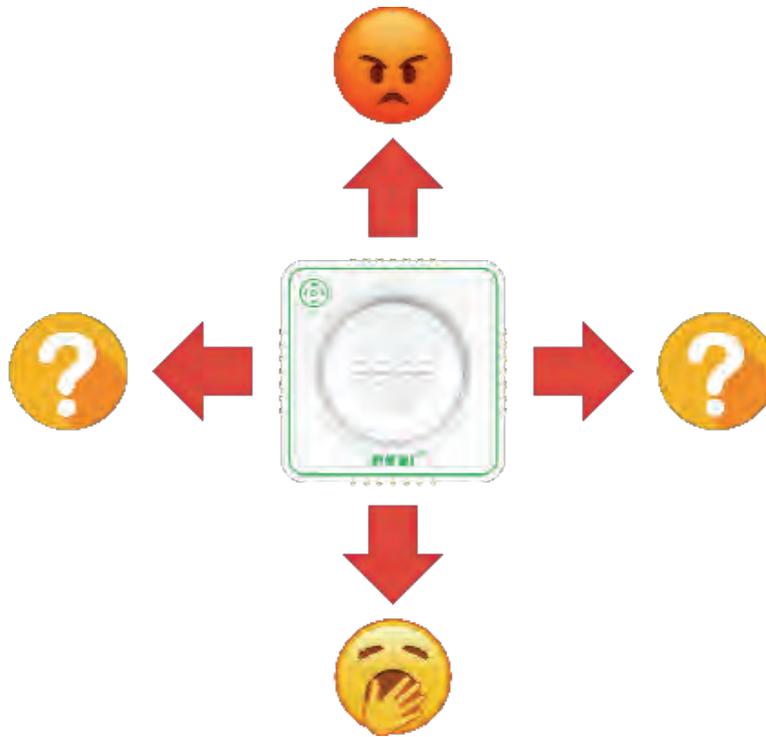
#### DEBUGGING

- 1 What if the wait block is above the show drawing block?
- 2 What if we changed Button's option to toggle?

## Missions

### Mission 1

- 1 Let's add two emotions linked to the left and right movements of the Joystick.



- 2 Write down the conditions and tasks in sentences for the code you wrote above.

Blank area for writing conditions and tasks.



**Mission 2**

**1** Below is a chart outlining face expressions corresponding to dial percentages. Complete the blanks with your own ideas.

	Over 50%	Under 50%
Up	Angry	
Down	Tired	
Right		
Left		

**2** Change the code below, as guided by the chart above.

```

while joystick 1 is down
  if dial 1 turn > 50 % then
    show drawing Tired on display 1
  else
    show drawing Surprised on display 1
  
```

# KNOWLEDGE CHECK

## 1 Why do we use emojis when we text?

- A To make texts longer
- B To show feelings and ideas without using many words
- C To make texts look bigger
- D To make texts look funny

**CORRECT ANSWER**  B

## 2 How did emojis first start?

- A They've always been on phones.
- B As simple smiley faces before becoming all sorts of pictures.
- C As a secret code only adults could understand.
- D They were drawn by astronauts in space.

**CORRECT ANSWER**  B

## 3 What special touch do emojis add to messages?

- A Lonley
- B Seriousness
- C A way to show how we're feeling
- D Extra length

**CORRECT ANSWER**  C

## 4 What does a "while" loop do in our programs?

- A It keeps doing something over and over as long as it's supposed to.
- B It skips parts of the program.
- C It makes the program wait.
- D It does something once and stops.

**CORRECT ANSWER**  A

## 5 When do we use "if" in our code?

- A To repeat an action many times.
- B To check if something is true and then decide what to do.
- C To give our program a rest.
- D To end the program.

**CORRECT ANSWER**  B

## 6 Which of the following is correct about wait block?

- A To do the next step right away.
- B To take a short break before doing the next step.
- C To turn off the program.
- D To make everything go faster.

**CORRECT ANSWER**  B

## 7 What happens in our code when we use a "while" loop with a condition that is always true?

- A The program immediately stops.
- B The "while" loop runs forever, or until the program is manually stopped.
- C The loop runs once and then stops.
- D The loop skips over and does nothing.

**CORRECT ANSWER**  B



## Critical Thinking Challenge

---

- 1** If you could invent an emoji to express a new emotion, what would it be and why? Describe how it looks, and how it would be used among people.

- 2** Reflect on a routine or habit in your daily life that's similar to a "while" loop in programming. Describe the routine, identify the "condition" that keeps it going, and explain how it eventually stops, just like a "while" loop would.



## UNIT 6 EXPANDING KNOWLEDGE ON THE TOPIC

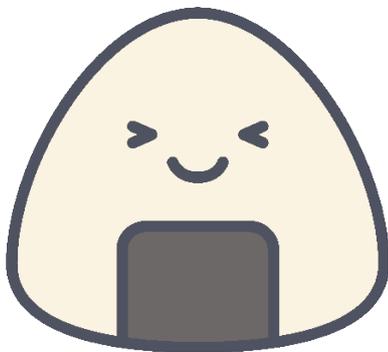
### Emojis Around the World: A Cultural Exploration

Did you know the same emoji can mean different things around the world? This fascinating twist adds color to our digital conversations, showing how diverse cultures shape our emoji usage.

#### Smiles and Gestures

A smiley 😊 might convey politeness in Japan, not just happiness.

And be careful with that thumbs-up 👍—it's not friendly everywhere!



#### Food for Thought

Emojis aren't just about emotions; they include a variety of objects, like food.

The rice ball 🍙 is a common snack in Japan and carries a sense of comfort and familiarity there. But outside of Japan, people might be less familiar with it, using it more for its visual appeal than its culinary significance.

#### Animal Symbolism

Animal emojis also carry diverse meanings. The owl 🦉, for example, is often associated with wisdom in Western cultures. However, in some other cultures, it might be seen as a bad omen. Similarly, the frog 🐸 might just be a cute amphibian to some, but in Japan, it's associated with good luck and safe travel because of the word "kaeru," which means both "frog" and "return."



#### Embracing Emoji Etiquette

As our world becomes more interconnected, understanding the cultural contexts of emojis can enhance our digital communication, making it more inclusive and empathetic. So, the next time you choose an emoji, consider its wider meanings. It's a small step towards embracing our global digital community, one emoji at a time. 👥



GRADE

7

PROJECT

3

# MUSICAL PROGRAMMING: COMPOSING WITH MODI MODULES



## What You Will Learn

- **Understand** the elements of music including notes, rhythms, and how they form a melody.
- **Learn** how sound can be digitally created and controlled through programming.
- **Discover** which MODI modules are used for sound production and how they interact within a system.
- **Learn** to write simple code sequences that translate musical ideas into functioning outputs.
- **See** how different MODI modules can be combined to build a complex electronic project like a music box.

## Learning Objectives

- **Use** your understanding of musical notes to program a music box that plays a specific tune.
- **Acquire** hands-on skills in assembling and configuring MODI modules for specific functionalities.
- **Improve** your ability to use control structures such as loops and conditionals to manage the timing and sequence of musical notes.
- **Develop** the competence to systematically test and debug your code to ensure the music box operates as intended.
- **Encourage** innovative thinking by challenging you to add unique features or additional songs to the music box project.



## UNIT 1 EXPLORING THE CORE

### Topic Investigation



#### WHAT ARE MUSICAL NOTES?

Musical notes are the foundation of any musical piece. They represent sounds at specific pitches, which when played in sequence, form melodies and harmonies that we enjoy in songs and performances. In written music, these notes are depicted using symbols placed on a staff—a series of five parallel lines. Each position on these lines or in the spaces between them corresponds to a particular musical pitch. For example, notes like A, B, C, D, E, F, and G are written on different lines or spaces, and each can be modified with sharps or flats to adjust their pitch.

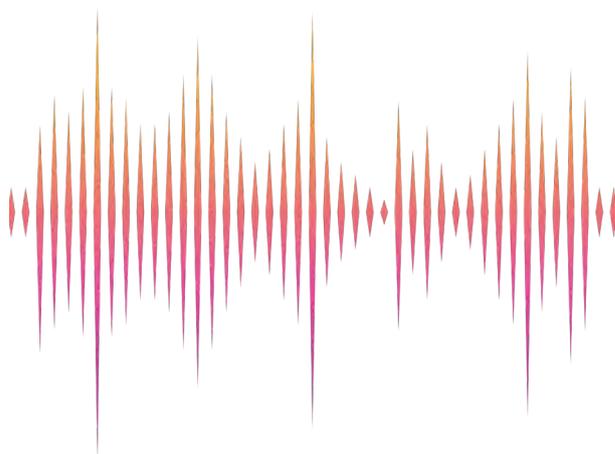
#### HOW DO WE HEAR MUSIC?

Music reaches us through sound waves created by vibrations from musical instruments, voices, or digital outputs like speakers. When an object vibrates, it causes the air around it to move, creating waves that propagate through the environment. These sound waves enter our ears and are translated by our auditory system into sounds that our brains perceive as music. The nature of these vibrations—how fast or slow they are—determines the pitch of the sound we hear.



#### FREQUENCY AND PITCH

The concept of frequency is central to understanding musical pitch. Frequency refers to the number of vibrations per second and is measured in Hertz (Hz). Each musical note corresponds to a particular frequency. For example, when we say a note is "middle C," we are referring to a sound wave that vibrates at 261.6 Hz.





## DETAIL ON PITCH

Pitch is our perception of how high or low a sound is and is closely related to the frequency of the sound wave. When discussing musical scales, particularly the standard Western music scale, pitch plays a crucial role in differentiating each note:



## MUSICAL SCALE AND NOTES

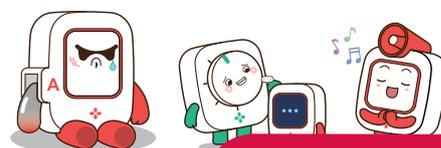
A musical scale is a series of notes that are ordered by pitch. In the Western music system, a standard scale consists of seven notes: Do (C), Re (D), Mi (E), Fa (F), Sol (G), La (A), and Si (B). Each note on this scale is associated with a specific pitch; for instance, middle C (Do) has a lower pitch than D (Re).

## FREQUENCY OF NOTES

Each successive note in a scale typically has a higher frequency than the one before it, which means each note sounds progressively higher. For example, middle C (Do) might be set at a frequency of 261.6 Hz, while D (Re) is at 293.7 Hz, and this pattern continues up the scale.

## OCTAVES AND DOUBLING FREQUENCY

Each successive note in a scale typically has a higher frequency than the one before it, which means each note sounds progressively higher. For example, middle C (Do) might be set at a frequency of 261.6 Hz, while D (Re) is at 293.7 Hz, and this pattern continues up the scale.



## Inquiry & Reflection

---

### Discussion question 1

**Content-Related Question:** Why do different instruments playing the same note sound unique? Consider the role of frequency and other factors.

### Discussion question 2

**Everyday Life-Related Question:** How does changing the speed of a song affect its pitch and your reaction to the music?

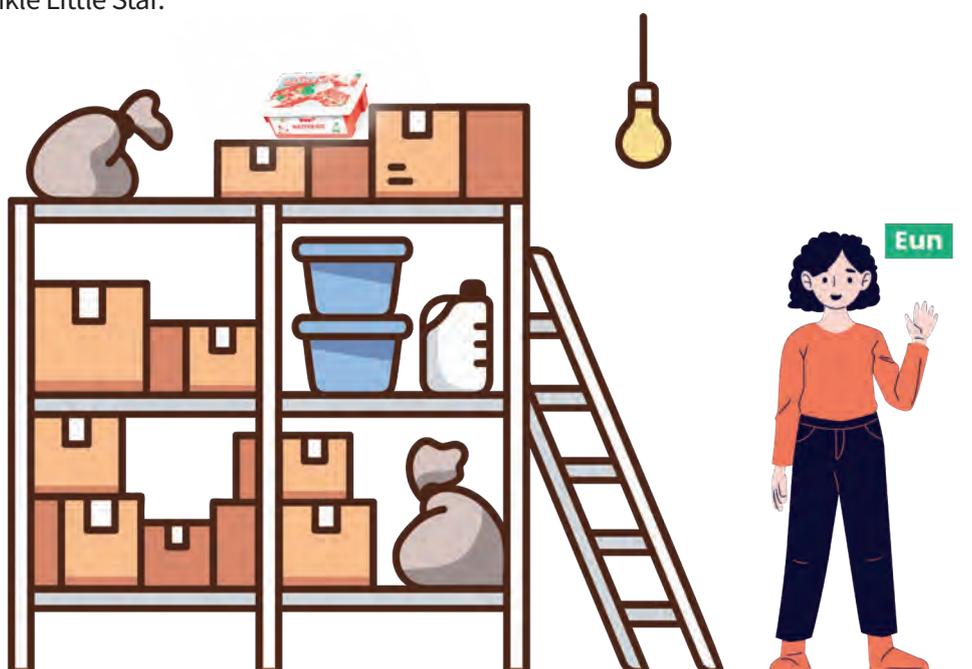


## UNIT 2 BRINGING IDEAS TO LIFE

### Project Introduction

#### TWINKLE TUNES: EUN'S MUSICAL DISCOVERY

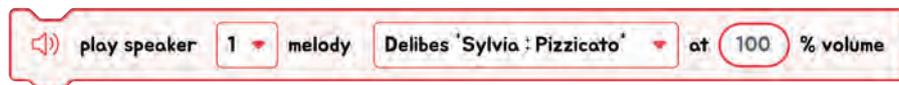
Meet Eun, a creative seventh grader who loves all things music and tech. One sunny afternoon, while digging through her family's old storage boxes in the attic, Eun found an intriguing box filled with various electronic parts. Among these were colorful modules, wires, and some buttons. The sight of these components sparked an idea in Eun's mind: why not build something that combines her love for music with her interest in technology? Excited by the challenge, Eun imagined creating a music box that could play her favorite lullaby, "Twinkle Twinkle Little Star."



Eager to share her idea and get some help, Eun called her friends, Cathy and Philipp. They loved the idea and agreed to help bring the music box to life. Together, the trio spent several afternoons after school working on the project. Eun and her friends designed a music box equipped with a button that, when pressed, plays "Twinkle Twinkle Little Star." The music box combines a storage mechanism for the tune and an output system for sound delivery, all integrated into a user-friendly and aesthetically pleasing device. This setup ensures each button press reliably activates the beautifully programmed melody, showcasing an effective blend of electronic hardware with creative software solutions.



## ABOUT SPEAKER MODULE

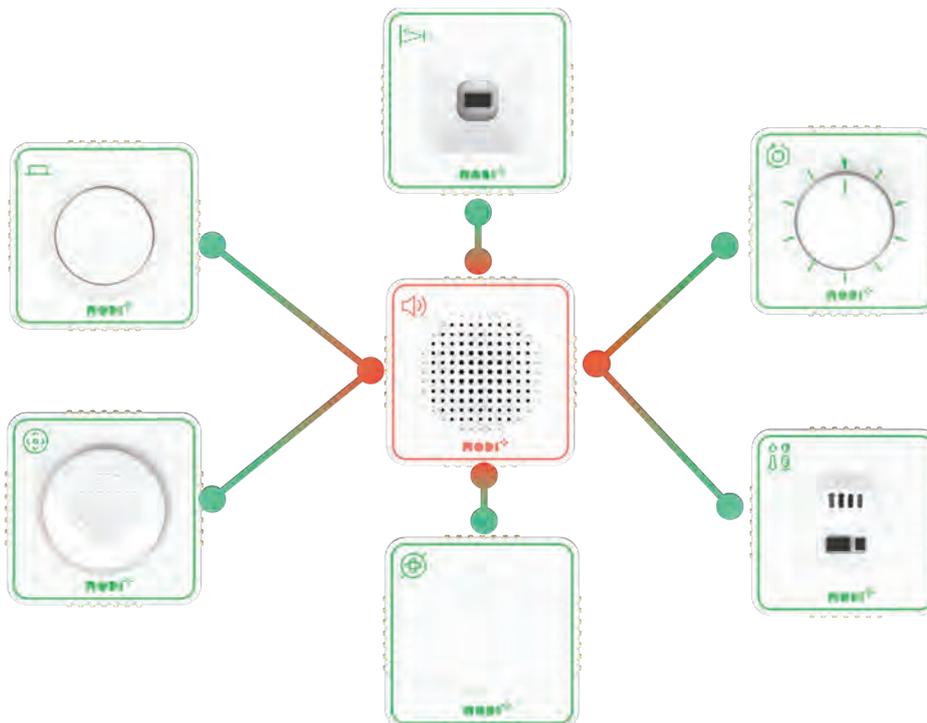


In the Code Editor, the Speaker module features four blocks in the block area. These blocks enable you to customize the sound of the Speaker module by setting its frequencies, selecting musical notes, and playing pre-programmed music.

## INTERACTIONS WITH OTHER MODULES

As an output module, the Speaker can interact with all input modules, producing a unique response to each. Explore these interactions by connecting the Speaker to different input modules and observe how it responds!

## INTERACTION BETWEEN LED AND OTHER MODULES



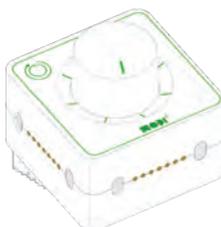


## Required Modules & Brainstorming

Based on the story, which MODI modules do you need for this creation?



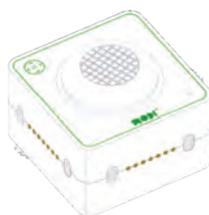
Button



Dial



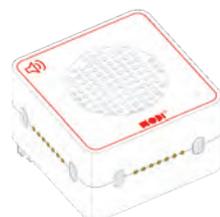
Environment



Joystick



ToF



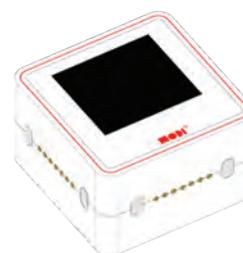
Speaker



IMU



LED



Display

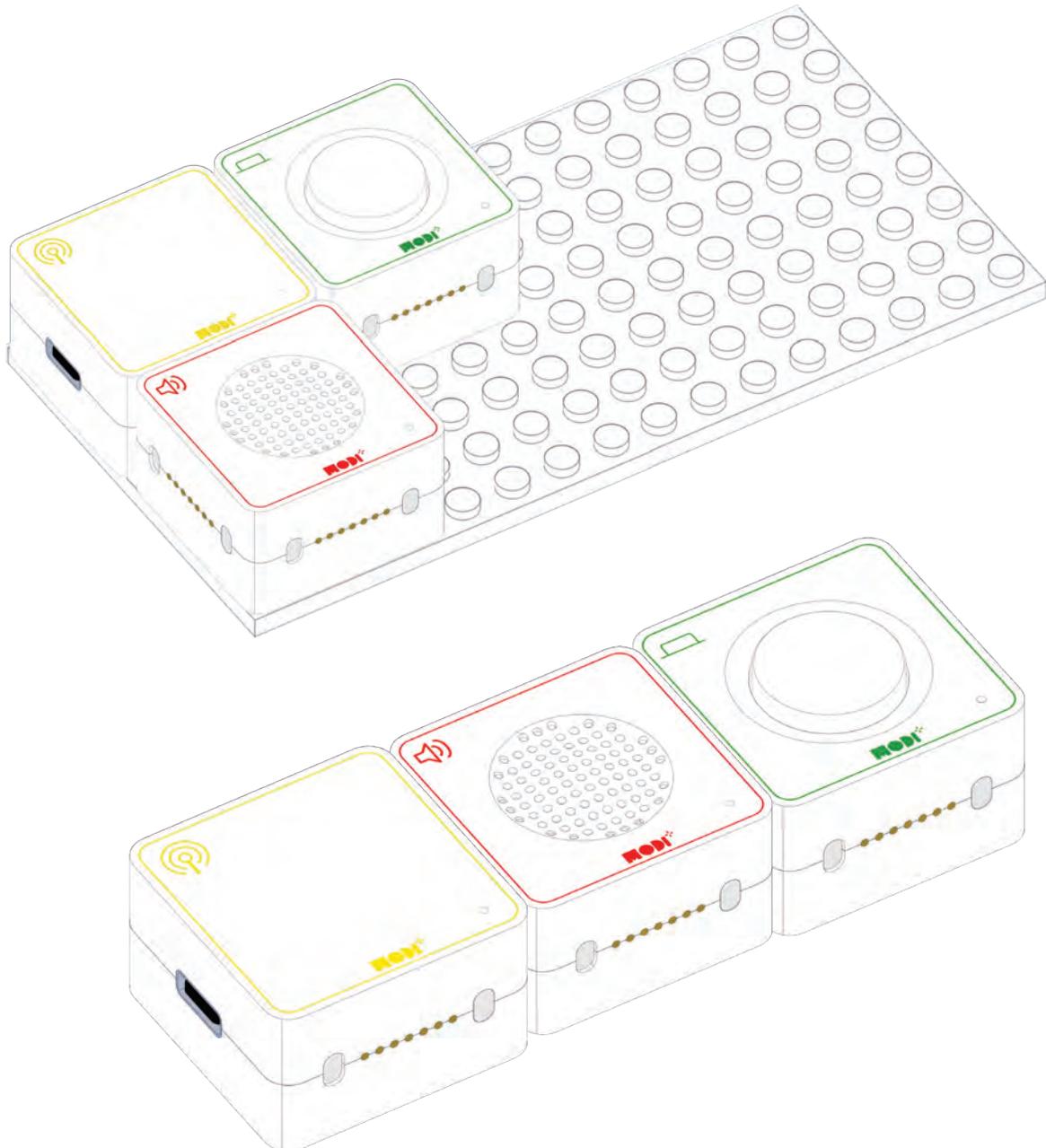


Motor A

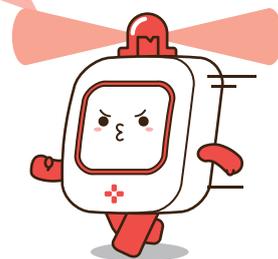


Motor B

## HOW DOES THE CREATION LOOK LIKE?

**THINK ABOUT IT!**

The physical architecture you assembled does not matter. The important thing is whether you chose the right modules and whether they are working





## Module Functionality

Let's look into the details of the ToF module.

### 1. Click



Detects if Button is clicked or not.  
Range: True(Click once) : 100 / False : 0

### 2. Double click



Detects if Button is double - clicked or not.  
Range: True(Click twice) : 100 / False : 0

### 3. Press



Show the press status of Button.  
Range: True(Pressed) : 100 / False : 0

### 4. Toggle



Changes on / off when Button is clicked.  
Range: True(On) : 100 / False : 0

### TIPS

You can observe how input data changes in the module map on the Code Editor when you experiment with the four different click types.



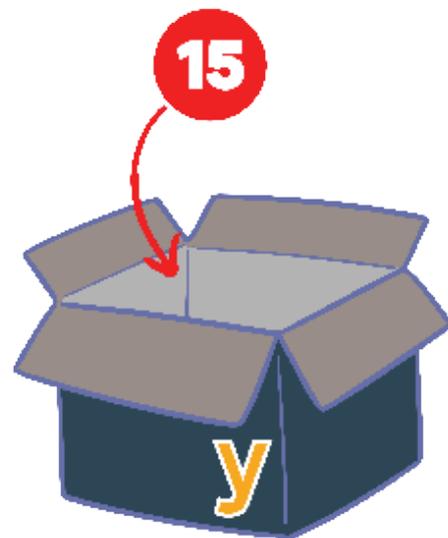
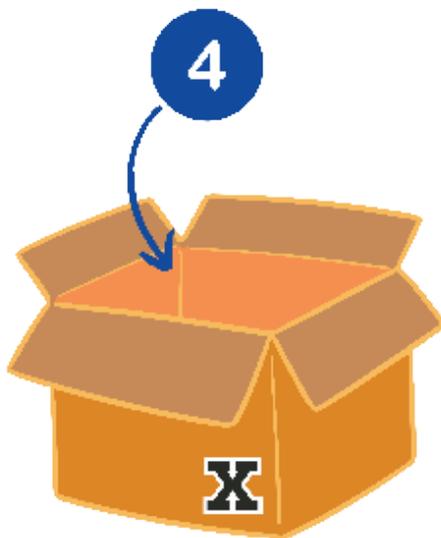


## UNIT 3 DESIGNING SOLUTIONS

### Programming Concepts

In mathematics, variables such as **x** and **y** are used to represent unknown values that can change. For instance, in the equation  $x + 2 = 5$ , **x** is a placeholder for a value we need to discover. Once we solve the equation, we find that **x** equals 3.

# ✓ VARIABLES



Similarly, in programming, a variable is a name assigned to a storage area in the computer's memory where data can be stored, modified, and retrieved later. Just like in math, the value of a programming variable can change as the program runs. For example, if we set a variable **x** in a program with  $x = 5$ , we can later change it to  $x = 10$  based on some conditions or calculations performed during the execution of the program.

### EXPLANATION WITH CODE EXAMPLE

## INCREMENT AND DECREMENT OPERATORS

Increment and decrement operators are used to adjust a variable's value by a fixed amount, typically 1, which simplifies coding in repetitive tasks. These operators are especially useful in loops for iterating over sequences.



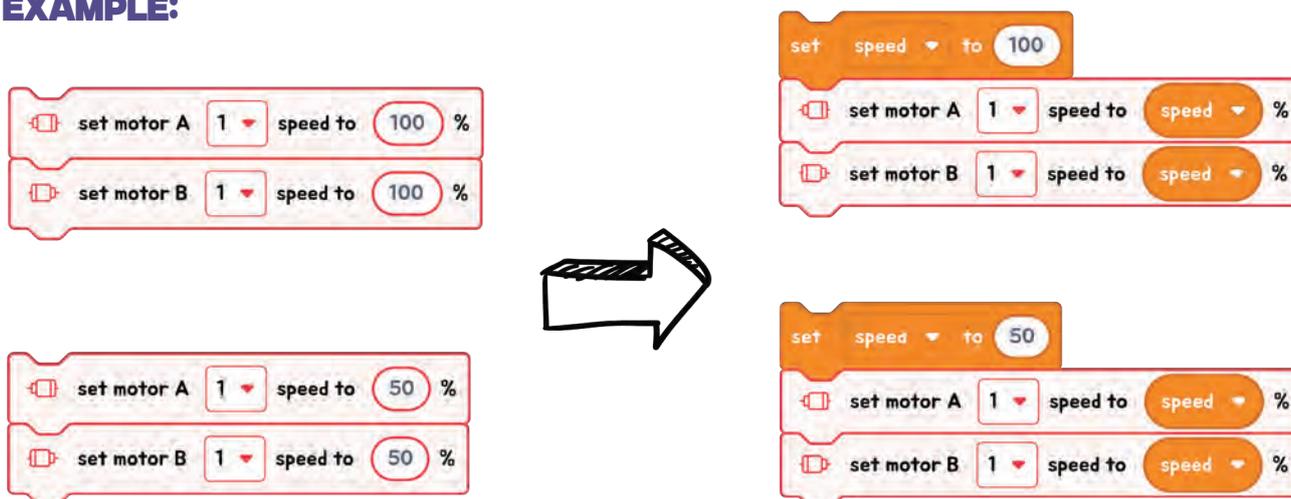
- The block on the left, known as the increment operator, increases the variable's value by 1.
- The block on the right, known as the decrement operator, decreases the variable's value by 1.



## HOW ARE VARIABLES USED?

Variables make it easier to update and manage your code. Instead of repeatedly changing the same piece of information everywhere it appears in your program, you can use a variable. This means you only need to update the value in one place, and every part of your program that uses this variable will automatically use the new value.

### EXAMPLE:



Imagine you have a video game where a car is traveling at a speed of 100 units. If you want to slow the car down to 50 units, using a variable for speed is very helpful. Instead of finding every place in your code where the speed is set to 100 and changing it to 50, you simply change the variable 'speed' from 100 to 50 once. Now, every part of your game that uses this 'speed' variable will automatically update to the new speed.

### THINK ABOUT IT!

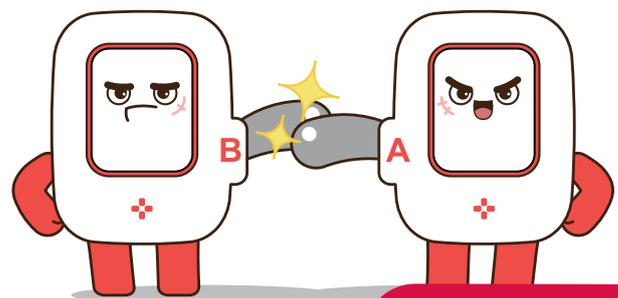
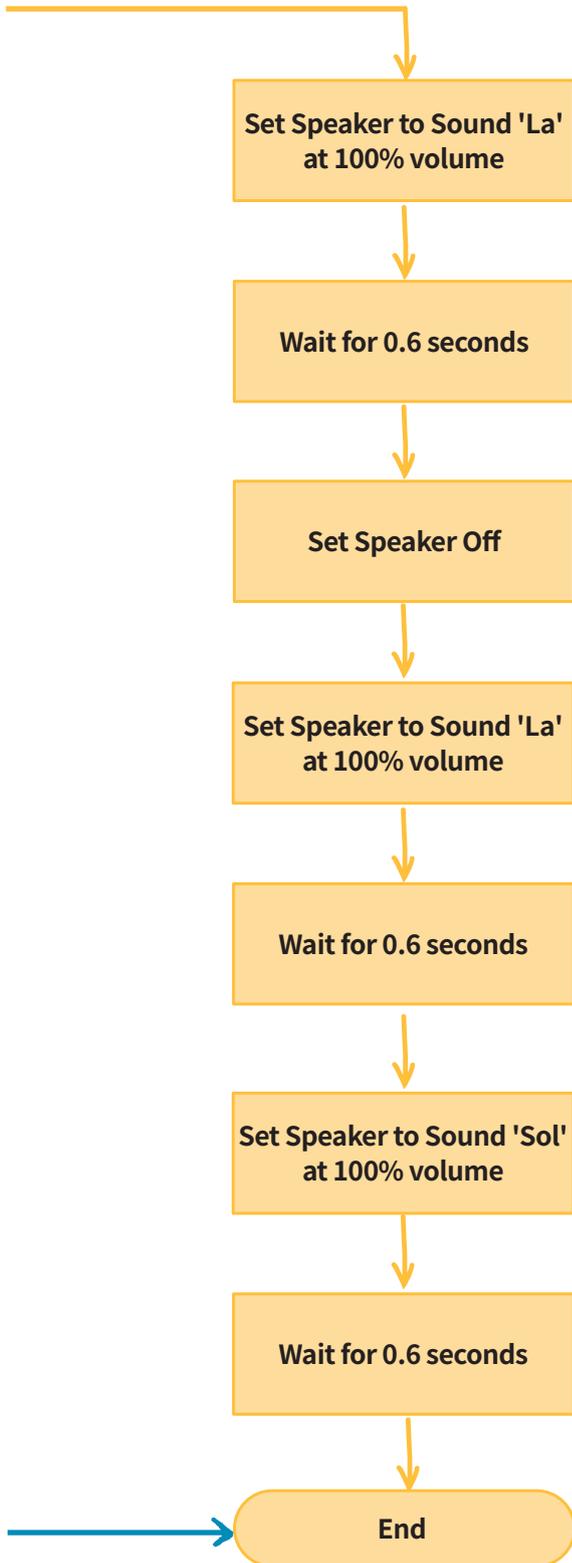
Imagine needing to change the speed in a game code in many places—maybe 10 or even 100 times. Without variables, you'd have to update each one by hand, which is slow and can lead to mistakes. Using variables, you just change the speed once, and it updates everywhere automatically. This saves time and reduces errors, making it easier to manage your code as it gets more complex.





**TIPS**

The flowchart from the previous page continues on this page.





## UNIT 4 CODE QUEST

### Applying Knowledge (code & debugging)

#### CODE

```
network
repeat forever
  turn off speaker 1
  if button 1 is clicked then
    set speaker 1 sound to Do at 100 % volume
    wait 0.6 seconds
    turn off speaker 1
    set speaker 1 sound to Do at 100 % volume
    wait 0.6 seconds
    set speaker 1 sound to Sol at 100 % volume
    wait 0.6 seconds
    turn off speaker 1
    set speaker 1 sound to Sol at 100 % volume
    wait 0.6 seconds
    set speaker 1 sound to La at 100 % volume
    wait 0.6 seconds
    turn off speaker 1
    set speaker 1 sound to La at 100 % volume
    wait 0.6 seconds
    set speaker 1 sound to Sol at 100 % volume
    wait 1.2 seconds
```

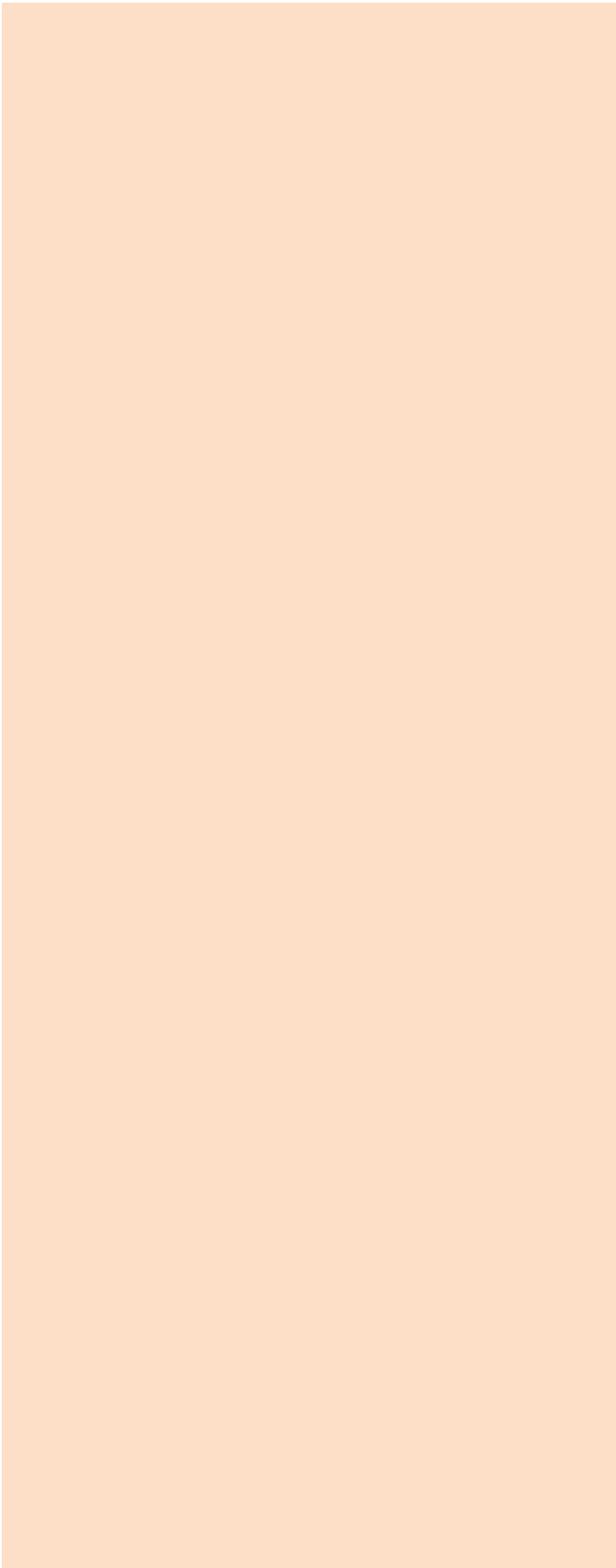
The code is a Scratch script starting with a 'network' block. It enters a 'repeat forever' loop. Inside the loop, it first turns off speaker 1. Then, it checks if button 1 is clicked. If clicked, it plays a sequence of notes: 'Do' (100% volume), 'Sol' (100% volume), 'La' (100% volume), and 'Sol' (100% volume). Each note is played for 0.6 seconds, followed by a 0.6-second wait. The sequence ends with a final 1.2-second wait.



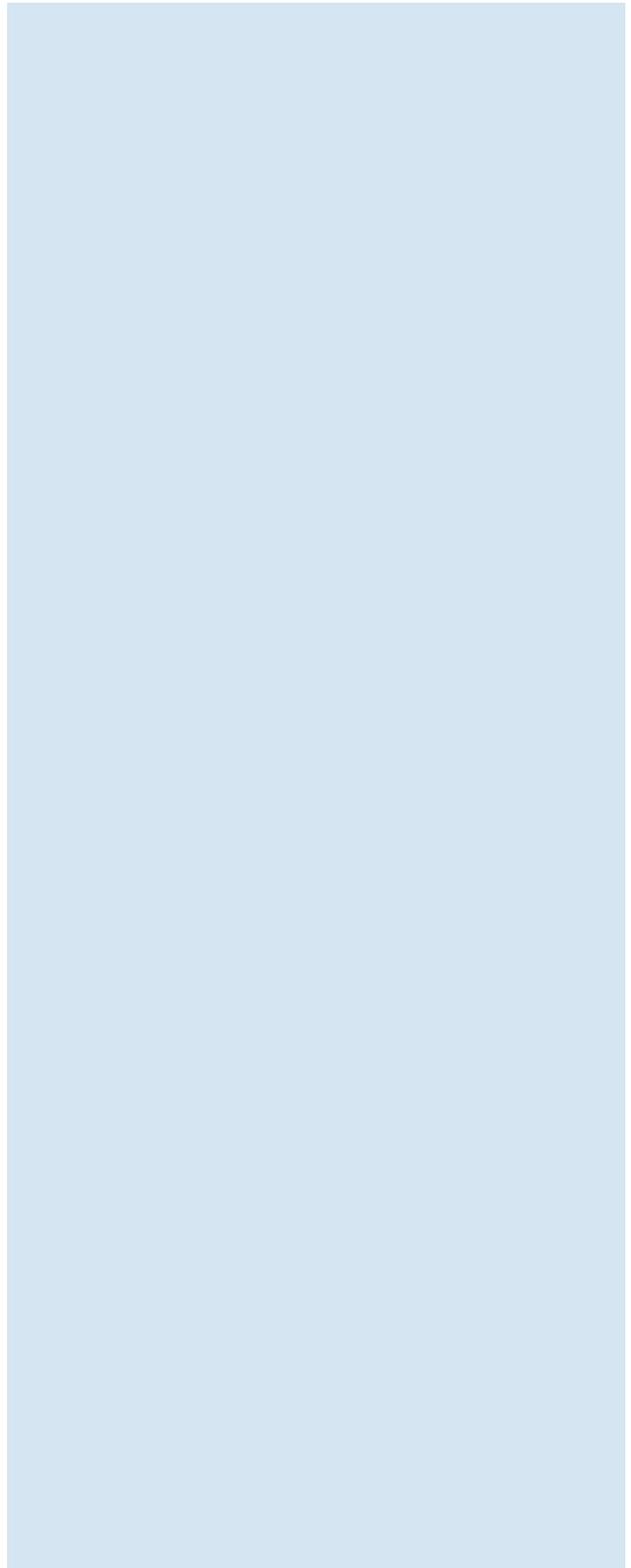
## DEBUGGING

There are two "Speaker Off" blocks in the sequence: one at the start and another in the middle of the song.

**1** What if the "Speaker Off" block at the start is missing?



**2** What if the "Speaker Off" block in the middle of the song is missing?



## Missions

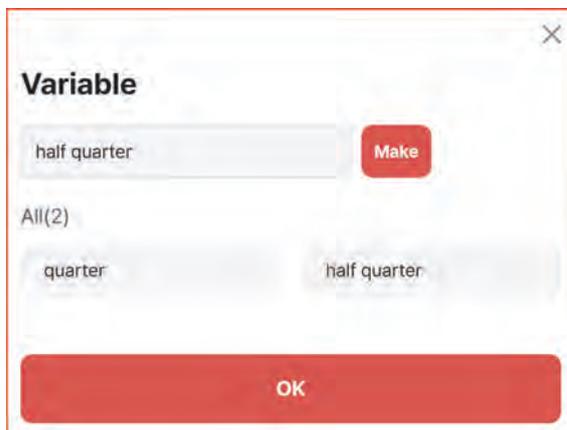
### Mission 1

#### 1 Let's replace repeating numbers with variables.

There are two different types of wait blocks in the code: ones with a 0.6-second wait and the other with a 1.2-second wait. Let's replace these two numbers with variables.

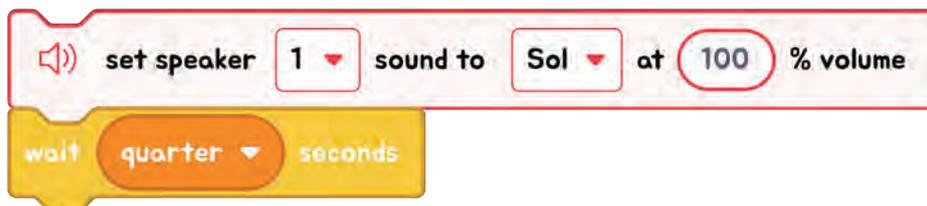
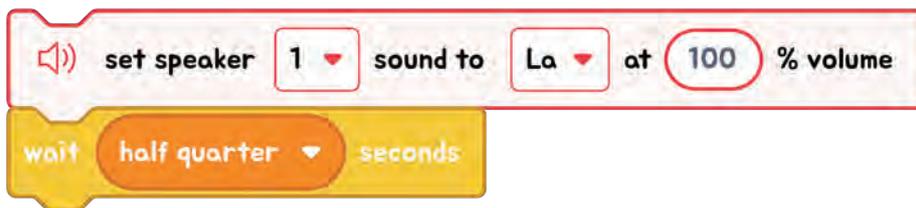
#### Step 1

Create two variables: one for the quarter note and another for the half note. Then, assign the appropriate duration to each variable.



#### Step 2

Replace the value "0.6" in the wait blocks with the variable "quarter" and the value "1.2" with the variable "half."





**2** Based on the blocks you've just created, write the corresponding code.

A large, empty yellow rectangular area intended for writing code.

**TIPS**

It's not that difficult. Once you have created the variables, simply replace the original values in the wait blocks with these variables.



## Mission 2

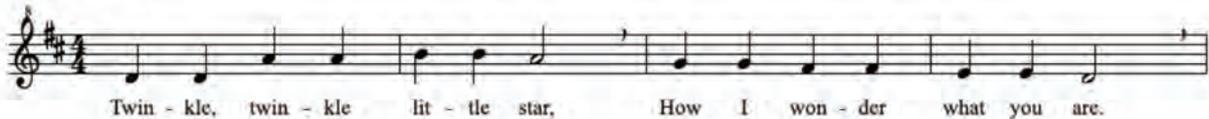
- 1 **Let's Finish the Song:** The music sheet contains a lot of repetition. To streamline our code, let's make use of loops to replicate these repeated sections efficiently.

## Twinkle, Twinkle, Little Star

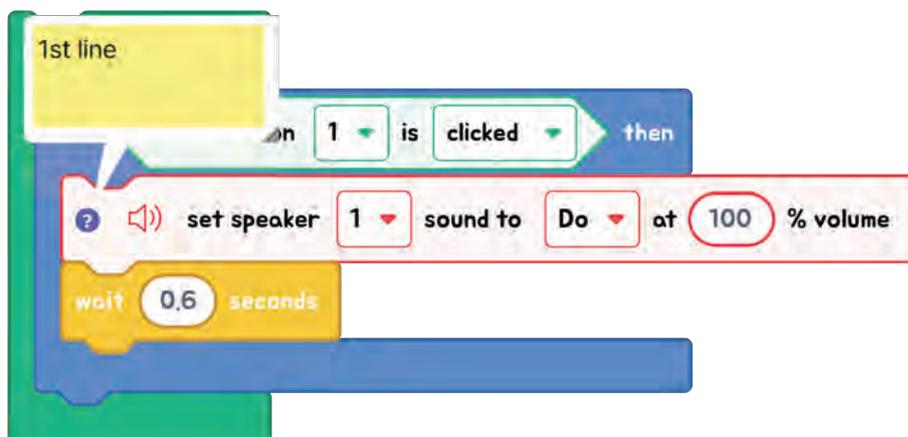
DEF#GAB

Traditional

Andante



- 2 **Change the Code:** If the code becomes too long and confusing, simplify it by using the comment function. When you press the 'Add Comment' button, a question mark will appear next to the block, where you can enter a comment to clarify what the block does.



# KNOWLEDGE CHECK



**1** What determines the pitch of a musical note?

- A The type of instrument played
- B The loudness of the note
- C The frequency of the sound wave
- D The length of the note

**CORRECT ANSWER**  C

**2** What is an octave in musical terms?

- A A series of three musical notes
- B A set of musical scales
- C A sequence of eight notes where the first and last notes have the same name but double the frequency
- D A type of musical composition

**CORRECT ANSWER**  C

**3** If a song is played faster than its normal speed, what happens to the pitch of the notes?

- A It increases
- B It decreases
- C It stays the same
- D It becomes inaudible

**CORRECT ANSWER**  C

**4** What is the purpose of using variables in programming a music box?

- A To store values like note durations that can be used multiple times
- B To increase the speed of the program
- C To create new sounds not existing in nature
- D To connect the music box to the internet

**CORRECT ANSWER**  A

**5** What does the 'loop' do in your music box code?

- A It stops the music from playing
- B It repeats a section of the code a specified number of times
- C It increases the volume of the music
- D It changes the pitch of the notes

**CORRECT ANSWER**  B

**6** Why would you add comments to your code?

- A To make the code run faster
- B To change the sequence of operations
- C To explain parts of the code, making it easier to understand
- D To play different songs automatically

**CORRECT ANSWER**  C

## Critical Thinking Challenge

---

- 1 Real-Life Impact of Frequency Adjustments:** Think about listening to your favorite song on different devices. How might changes in sound quality or pitch affect your enjoyment of the song?

- 2 Real-World Application of Programming Concepts:** How can loops and variables be used to address a real-world problem? Provide an example.



## UNIT 6 EXPANDING KNOWLEDGE ON THE TOPIC

### Exploring the Fascinating World of Frequencies

#### THE SCIENCE OF SOUND FREQUENCIES

Sound is produced by vibrations that create waves traveling through a medium such as air. The frequency of these waves, measured in hertz (Hz), determines the pitch of the sound we hear. Higher frequencies produce higher pitches, while lower frequencies result in lower pitches. This fundamental concept helps us understand how musical instruments can create a symphony of sounds just by varying the frequencies they produce.

#### FREQUENCIES BEYOND MUSIC

Beyond the realm of music, frequencies have numerous applications. In medicine, for instance, different frequencies are used in diagnostics, such as ultrasound imaging, which employs high-frequency sound waves to create images of organs inside the body. Similarly, in telecommunications, different frequencies facilitate the transmission of data over the airwaves, enabling everything from TV broadcasts to mobile phone communications.



#### EVERYDAY LIFE AND FREQUENCIES

Every day, without realizing it, we interact with a variety of devices that rely on frequencies. From the microwave that cooks your food using specific frequencies to interact with water molecules, to the Wi-Fi router that uses radio frequencies to send data across the air, frequencies are integral to modern technology and convenience.

#### SUMMARY

As we continue to expand our knowledge about frequencies, we gain more insights into how this fundamental aspect of physics influences not only the music we enjoy but also the technologies we rely on every day. This deeper understanding enhances our appreciation of the world around us and empowers us to use science in creative and innovative ways. In your next projects, think about how you can manipulate frequencies not just in music but in any creative tech application you pursue.

**GRADE**  
**7****PROJECT**  
**4**

## HARNESSING RENEWABLE ENERGY



### What You Will Learn

- **Types of Renewable Energy:** Understand the various forms of renewable energy such as solar, wind, and hydroelectric power.
- **Benefits of Renewable Energy:** Learn why renewable energy is crucial for reducing carbon emissions and promoting sustainable growth.
- **Solar Energy Basics:** Discover how solar panels convert sunlight into electricity.
- **Hands-On Construction Skills:** Gain practical experience in building and assembling a solar-powered device.
- **Real-World Applications:** See examples of how renewable energy is used around the world to power communities sustainably.

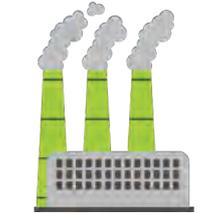
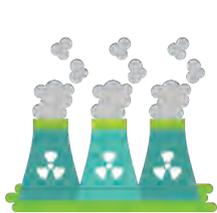
### Learning Objectives

- **Identify Renewable Energy Sources:** Recognize and describe different renewable energy sources and their roles in energy production.
- **Explain Solar Power Mechanisms:** Articulate how solar power is generated and how it can be utilized in engineering projects.
- **Develop Programming Skills:** Enhance programming skills by applying logical conditions and operations to make functional devices.
- **Analyze Environmental Impact:** Assess the environmental and social impact of using renewable technologies in everyday life.
- **Engage in Critical Thinking:** Engage in discussions about the challenges and opportunities that renewable energy presents for future generations.



## UNIT 1 EXPLORING THE CORE

### Topic Investigation



### EXPLORING RENEWABLE ENERGY

Welcome to the first unit of our journey into renewable energy, a key component in fighting climate change and creating a sustainable future. In this unit, we'll discover the different types of renewable energy sources and understand why they are essential for our planet.

### TYPES OF RENEWABLE ENERGY

Renewable energy comes from natural sources that continuously replenish themselves. The main types we'll cover include:



#### SOLAR POWER

Energy from the sun. This is harnessed using solar panels that convert sunlight directly into electricity.

#### WIND ENERGY

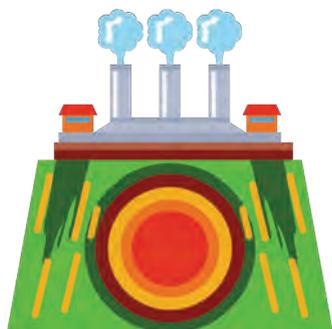
Energy generated by the wind. Wind turbines capture kinetic energy from the wind and convert it into electricity.





### HYDROELECTRIC POWER

Energy from water. This type of energy uses water flow, typically from rivers or dams, to generate power.

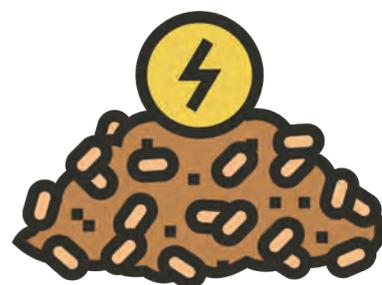


### GEOHERMAL ENERGY

Energy from the heat beneath the earth's surface. This can be harnessed to produce heat or electricity.

### BIOMASS ENERGY

Energy from organic materials. Plants, agricultural crops, and waste materials can be used to create bioenergy.



### WHY RENEWABLE ENERGY?

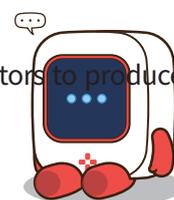
Renewable energy is crucial for reducing our carbon footprint. Unlike fossil fuels, renewable sources produce little to no greenhouse gases or pollutants. Here's why embracing renewable energy is essential:

- **Environmental Impact:** Reduces the effects of global warming and air pollution.
- **Sustainability:** Ensures a long-lasting energy supply without depleting natural resources.
- **Economic Benefits:** Creates jobs in new and emerging green technologies.

### THE SCIENCE BEHIND RENEWABLE ENERGY

Each type of renewable energy relies on specific scientific principles to convert natural phenomena into usable power. For example:

- **SOLAR PANELS** use photovoltaic cells to convert sunlight into electricity.
- **WIND TURBINES** use aerodynamic forces to turn rotor blades that drive generators to produce electricity.



## **Inquiry & Reflection**

### **Discussion question 1**

Which type of renewable energy do you think is most effective for our local community? Why?

### **Discussion question 2**

How could increasing the use of renewable energy change our daily lives and local environment?





## UNIT 2 BRINGING IDEAS TO LIFE

### Project Introduction

#### TWINKLE TUNES: EUN'S MUSICAL DISCOVERY



One sunny afternoon, seventh grader Hugh excitedly proposed a project to his friends, Cathy and Dawn: "Let's build a solar windmill!" Inspired by their recent lessons on renewable energy, the trio was eager to see these principles in action. Gathering in Hugh's backyard, they pooled their resources—small motors and solar panel from Cathy's old kit.

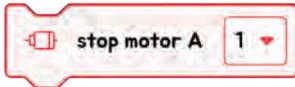
As they laid out the materials, Hugh suggested, "What if our windmill could use both wind and solar power?" His idea was to make the windmill blades rotate with wind energy while using solar power to light it up. Cathy and Dawn loved the hybrid concept, which perfectly demonstrated the synergy between different renewable energies. Filled with enthusiasm, they began sketching designs, ready to bring their innovative solar windmill to life.



Their completed solar windmill not only functioned as a brilliant school project but also sparked interest within their community. Hugh, Cathy, and Dawn showcased their creation at the local community center, explaining how renewable energy could promote environmental sustainability. They inspired neighbors to consider how small-scale renewable energy projects could be implemented in their homes.



### ABOUT MOTOR A MODULE

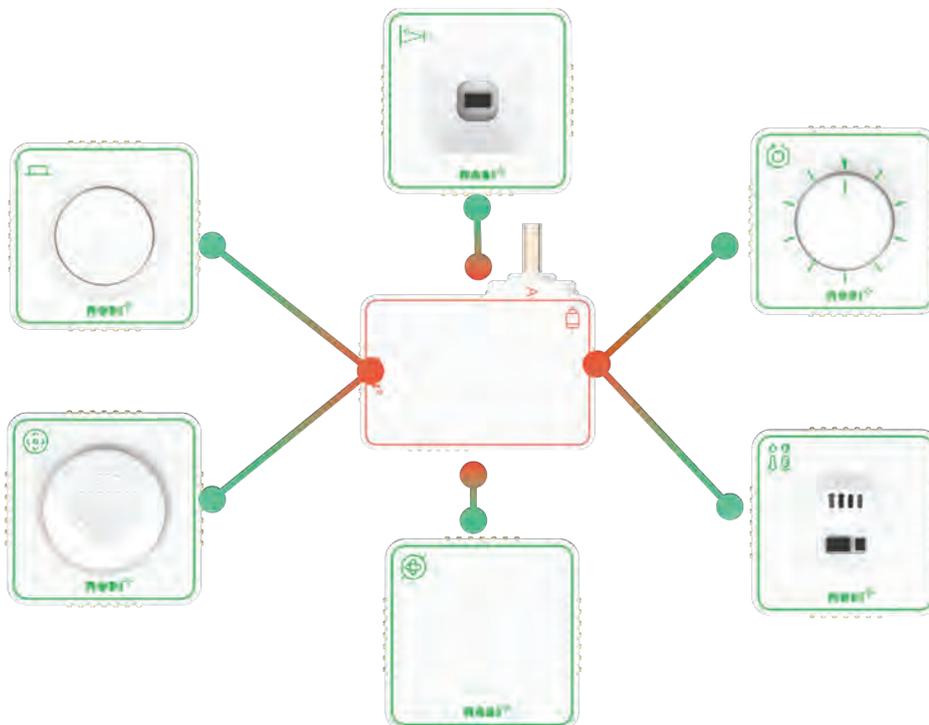


Motor A features five different blocks in the block area. These blocks allow you to customize your creations by setting the motor's angle, spinning direction, and speed.

### INTERACTIONS WITH OTHER MODULES

As an output module, Motor A can interact with all input modules, showing a unique response to each one. Explore these interactions by connecting Motor A to different input modules!

### INTERACTION BETWEEN LED AND OTHER MODULES



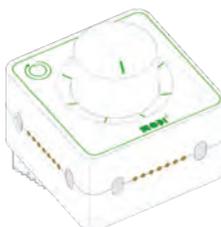


## Required Modules & Brainstorming

Based on the story, which MODI modules do you need for this creation?



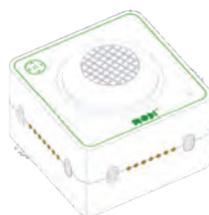
Button



Dial



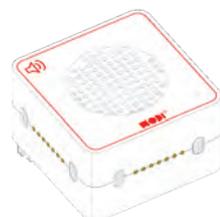
Environment



Joystick



ToF



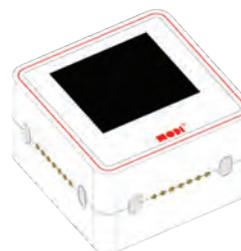
Speaker



IMU



LED



Display

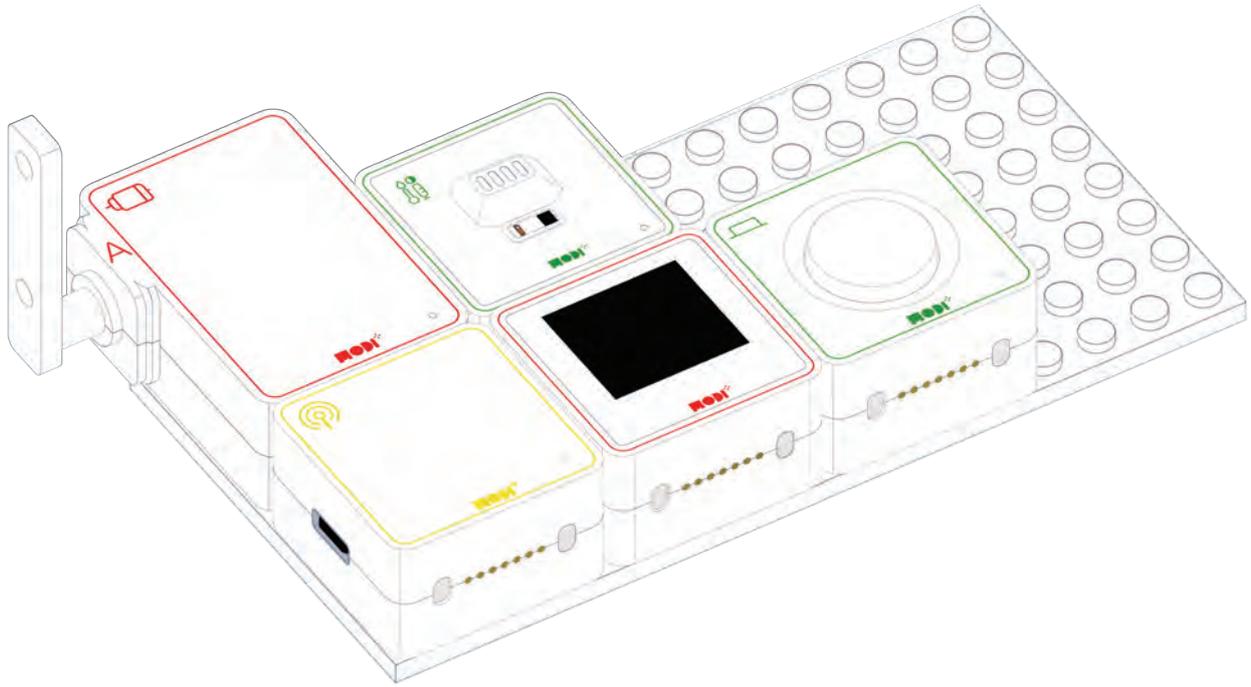


Motor A



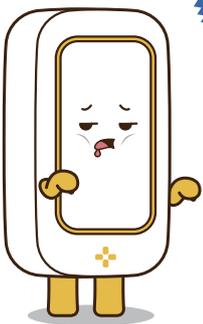
Motor B

HOW DOES THE CREATION LOOK LIKE?



**TIPS**

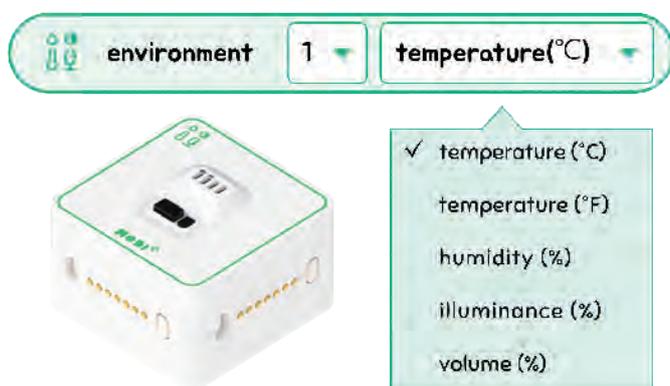
If you make wings and attach them to the I-horn, it will perfectly resemble a windmill.





## Module Functionality

Let's check the details of the Environment module below for this project.



### ENVIRONMENT

- **Temperature:** Measures how hot or cold these surroundings are.
- **Humidity:** Checks the amount of moisture in the air.
- **Illuminance:** Gauges the brightness of the light around it.
- **Volume:** Detects the level of sound in the environment.

The Environment Module is a critical tool for monitoring conditions that impact renewable energy sources. It tracks three crucial environmental factors:

#### 1. TEMPERATURE MEASUREMENT

This function measures the ambient temperature, which can affect the efficiency of solar panels and other energy systems. Maintaining an optimal temperature range is crucial for maximizing energy production.

#### 2. HUMIDITY MEASUREMENT

Humidity levels can influence the performance of renewable energy equipment. This part of the module checks the moisture in the air, which is essential for optimizing conditions for energy generation.

#### 3. LIGHT MEASUREMENT

Light intensity directly impacts the output of solar energy systems. This device ensures there is sufficient sunlight for solar panels to operate effectively, crucial for areas relying on solar power.

#### 4. NOISE MEASUREMENT

While typically less directly related to renewable energy, understanding noise levels can be important for the placement and operation of equipment like wind turbines, which may be sensitive to vibration or need to adhere to local noise regulations.





## UNIT 3 DESIGNING SOLUTIONS

### ● Programming Concepts

Imagine you and your friend want to play outside, but both of you have homework to finish. You agree that you can only go out if both of you complete your homework. This agreement is like using the "AND" operator in programming.



If you say, "I will finish my homework AND you will finish yours," it means both conditions must be true for you to go outside. If either of you doesn't finish your homework, you both stay inside.

## ✓ AND



### Condition 1

I finish my homework



### Condition 2

You finish your homework



### Task

We play outside



In programming, when you use the "AND" operator in an if statement, it ensures that both conditions must be true for the code block to execute. So, if your if statement includes two conditions separated by "AND," both must be satisfied for the code within the if statement to run.

Condition 1

and ▼

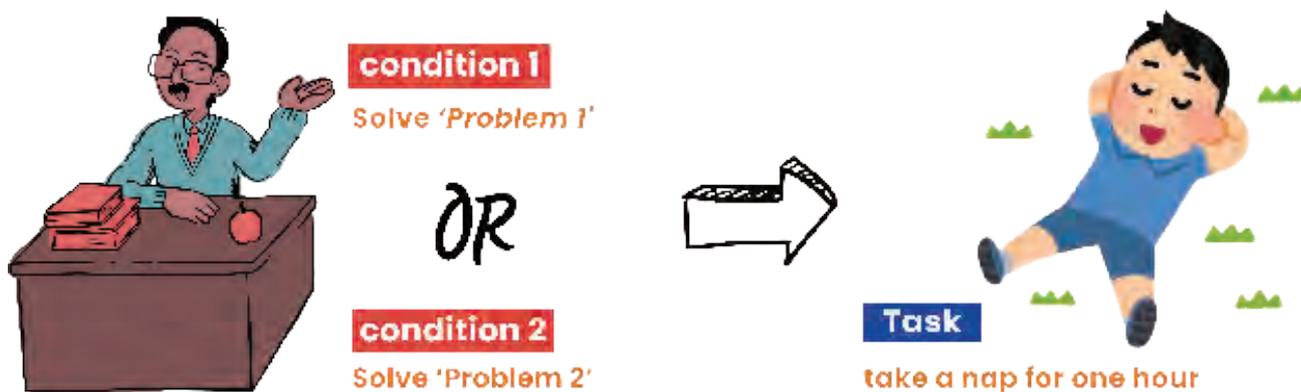
Condition 2



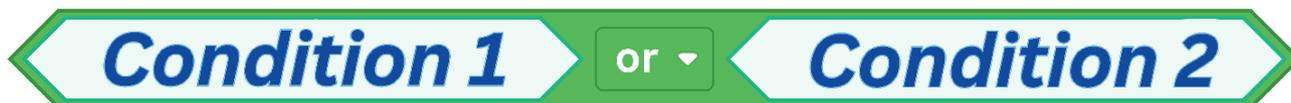
## OR

Think of the "or" operator in programming as providing options, similar to how choices might be given during school.

# ✓ OR



For example, imagine the principal announces during a break, "If anyone can solve either of these math problems (Problem 1 or Problem 2), then all students will get one hour of nap time today." This scenario does not require solving both problems—solving just one will grant the reward.



Similarly, in programming, using the "or" operator means if any of the conditions connected by "or" is true, the associated action will be taken.

### TIPS

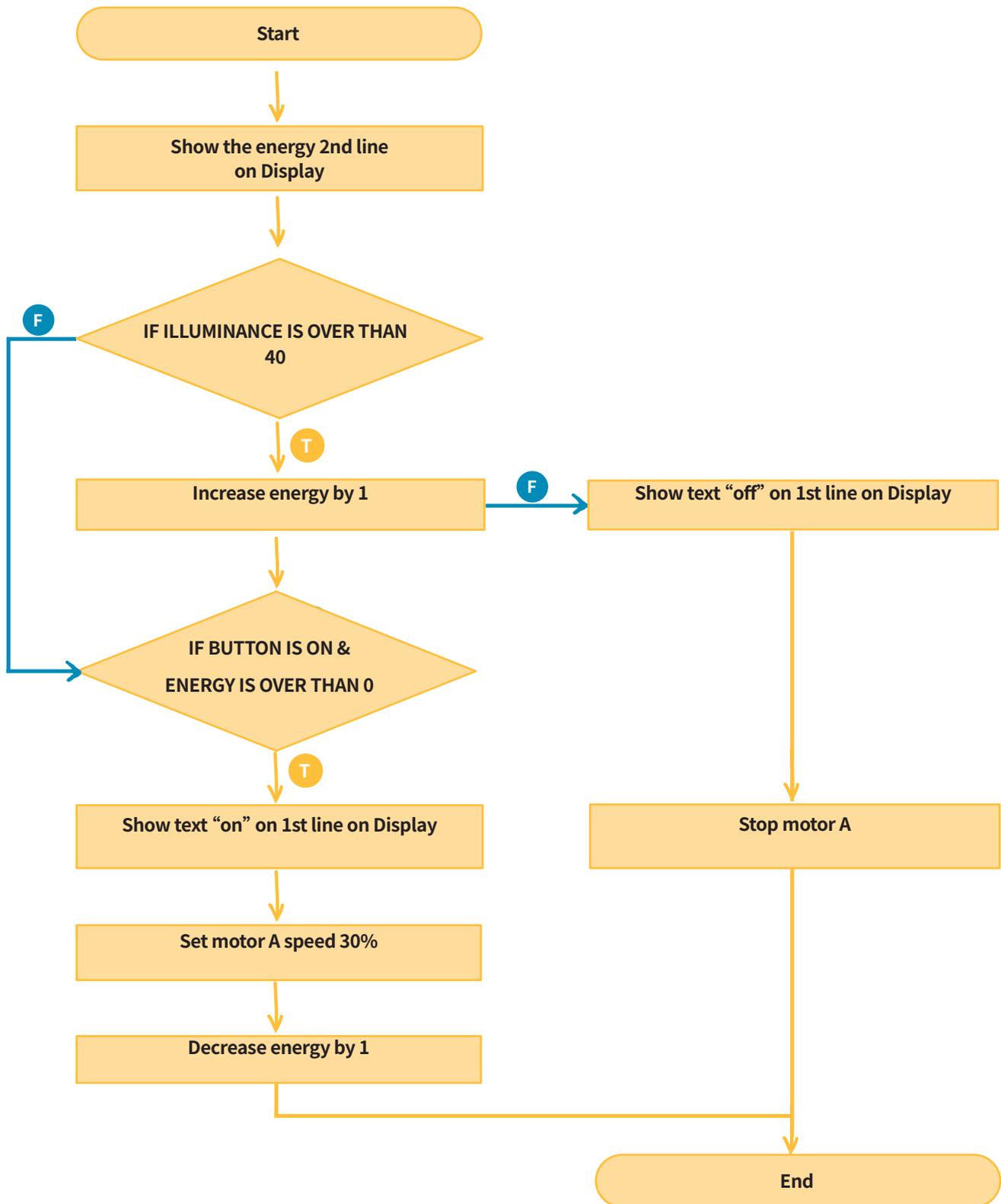
When using the "or" operator, it's crucial to manage the logical flow of your program effectively. **If any connected condition is true, the overall condition is evaluated as true.**



## Flowcharts & Algorithms

Let's read the flowchart and create the code.

### FLOWCHART





## UNIT 4 CODE QUEST

### Applying Knowledge (code & debugging)

#### CODE

```
network
repeat forever
  show variable energy on second line of display 1
  if environment 1 illuminance > 40 % then
    energy 1 addition
  if button 1 is toggle on and energy > 0 then
    show text on on display 1
    energy -1 addition
    set motor A 1 speed to 30 %
  else
    show text off on display 1
    stop motor A 1
```

#### DEBUGGING

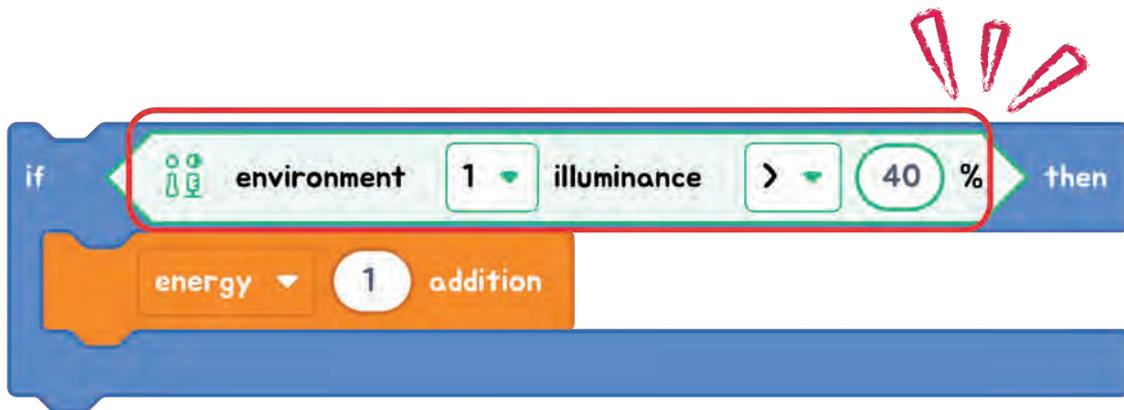
1 What if the condition "Energy > 0" is missing?

2 What if we changed the "and" operator to "or"?

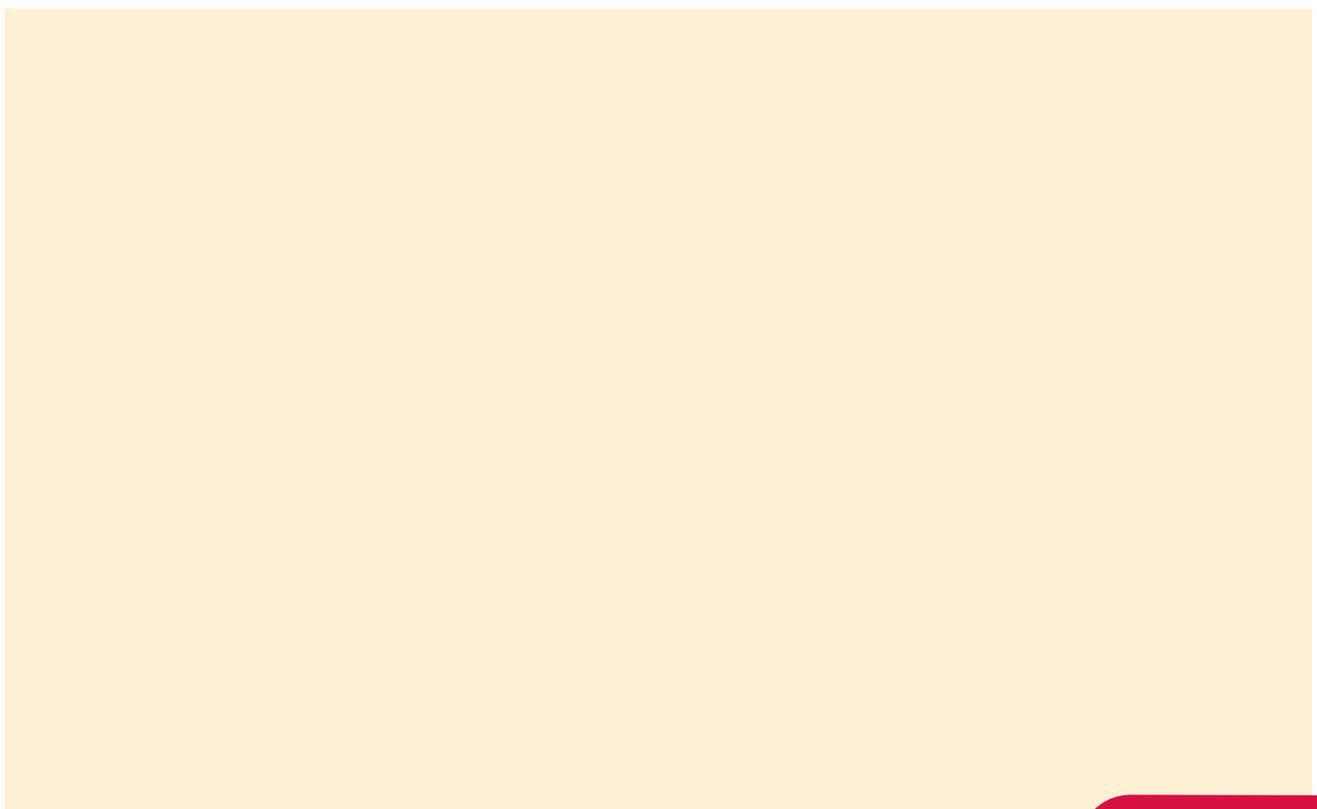
## Missions

### Mission 1

- 1 Change the condition to use a different energy source instead of illuminance.



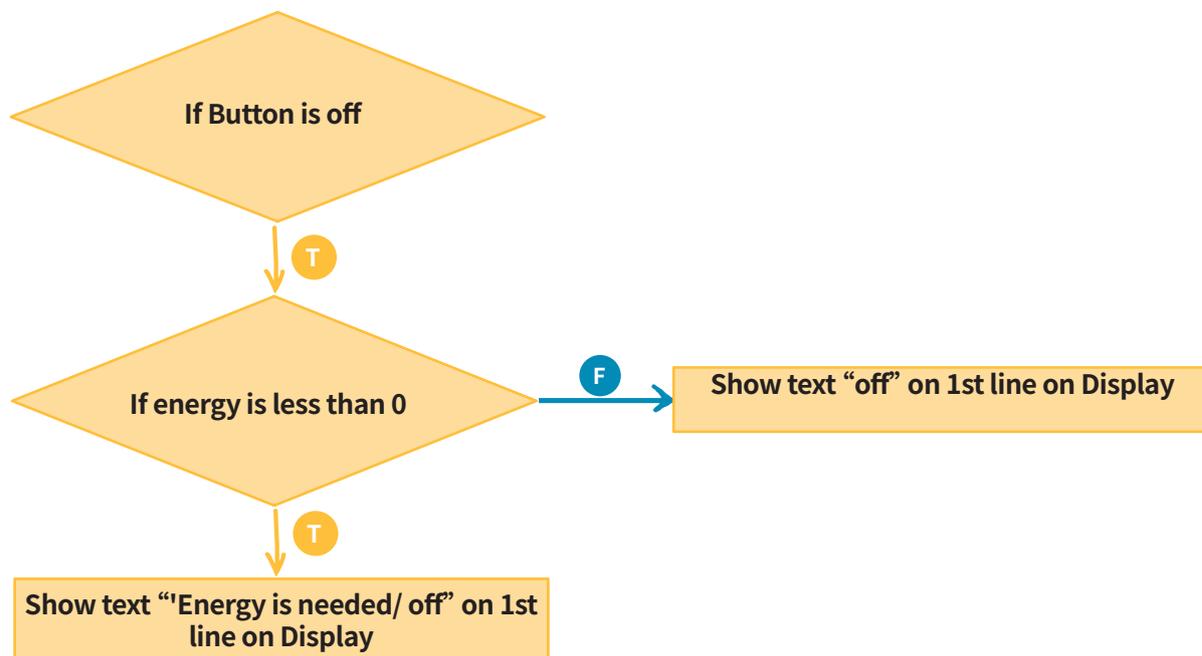
- 2 Based on the flowchart, make the code





## Mission 2

- 1 If the energy level is 0, display the message 'Energy is needed' along with 'Off'. However, if the energy level is sufficient but the motor is not working, display only the 'Off' message.



- 2 Based on the flowchart, make the code

# KNOWLEDGE CHECK

**1** What is the primary function of a solar windmill in renewable energy systems?

- A To produce chemical energy
- B To convert solar energy into electrical energy
- C To reduce solar radiation
- D To increase humidity levels

**CORRECT ANSWER**  B

**2** Which energy source is commonly used in renewable energy projects like solar windmills?

- A Coal
- B Solar
- C Natural Gas
- D Petroleum

**CORRECT ANSWER**  B

**3** What impact does using renewable energy have on the environment?

- A Increases air pollution
- B Reduces carbon emissions
- C Causes water pollution
- D All of the above

**CORRECT ANSWER**  B

**4** Why is humidity measurement important in environmental monitoring for renewable energy sources?

- A It affects the comfort of workers
- B It influences the efficiency of solar panels
- C It changes the color of the equipment
- D It increases energy consumption

**CORRECT ANSWER**  B

**5** What is the purpose of using an 'if else' statement in programming a solar windmill?

- A To choose between two energy sources
- B To control the display messages based on energy levels
- C To increase the speed of the windmill
- D To connect to the internet

**CORRECT ANSWER**  B

**6** What happens if you change an 'and' operator to an 'or' operator in a condition checking both energy sufficiency and motor status?

- A The system becomes less secure
- B It will display 'Off' more frequently
- C The system will display messages only when both conditions are met
- D The condition becomes true if either or both situations are true

**CORRECT ANSWER**  D

**7** Why would you use variables in programming the windmill?

- A To store temporary data like energy levels
- B To make the program longer
- C To create graphics for the display
- D To connect with other windmills

**CORRECT ANSWER**  A



## Critical Thinking Challenge

---

- 1** Discuss how the integration of solar and wind power in a single renewable energy project like a solar windmill could benefit your local community economically and environmentally.

- 2** Reflect on how changing one programming variable in the solar windmill could significantly alter its functionality. Provide an example of such a change and its potential impacts.



## UNIT 6 EXPANDING KNOWLEDGE ON THE TOPIC

### The Future of Natural Energy Sources



As we delve deeper into the realm of renewable energy, an interesting area of exploration is the sustainability and longevity of natural energy sources. Earth's natural energy reserves, such as sunlight, wind, and geothermal heat, are considered inexhaustible on a human time scale. Unlike fossil fuels, which are finite and deplete with usage, renewable resources regenerate and can supply continuous energy. Solar energy, for example, is harnessed from the sun's radiation, which the sun will emit for another estimated 5 billion years. This makes solar power a particularly sustainable option for the future of energy.

However, while the primary sources of renewable energy are virtually limitless, the infrastructure needed to capture and convert these energies, such as solar panels and wind turbines, requires significant resources and maintenance. The materials used to build these structures, such as rare earth metals and composites, are often scarce and pose their own sustainability challenges. As we look towards a future where renewable energy may become our primary energy source, research



Grade - 1



Grade - 2



Grade - 3



Grade - 4



Grade - 5



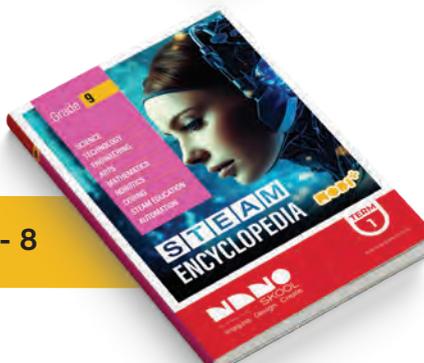
Grade - 6



Grade - 7



Grade - 8



# National Robotics Literacy Mission for Viksit Bharat 2047 NanoSkool STEAM Workshops in K12 Schools across India



# FUTURE CLASSROOMS

## STEAM

# ROBO-CODING SUITE FROM South Korea



### OUR PARTNER SCHOOLS

 <b>NARAYANA GROUP OF SCHOOLS</b> WEST BENGAL	 <b>AMARA JYOTHI PUBLIC SCHOOL</b> BANGALORE	 <b>VALISTUS INTERNATIONAL SCHOOL</b> BANGALORE	 <b>LABUR INDIA PUBLIC SCHOOL</b> KERALA	 <b>ST PAUL'S SCHOOL</b> RAJKOT	 <b>WHITEFIELD GLOBAL SCHOOL</b> BANGALORE	 <b>NIRMALA CONVENT SCHOOL</b> GUJARAT	 <b>ST XAVIER'S PUBLIC SCHOOL</b> GANDHIDHAM
--	---	--	---	---	---	---	---



Nano Skool Education Pvt. Ltd.  
 HD.172, Wework Prestige Atlanta,  
 Koramangala, Bangalore - 560 034,  
 Karnataka, India.

+91 97786 65473  
 E-mail: info@nanoskool.com

