



STEMpedia

YOUNG MAKERS' GUIDE

through the peninsula of DIY-ing

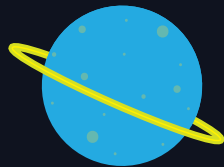
Unleash the innovator in you as you explore:

THE GATEWAY OF ELECTRONICS

THE LAND OF CODE-O-NUTS

THE ROBO RANGES

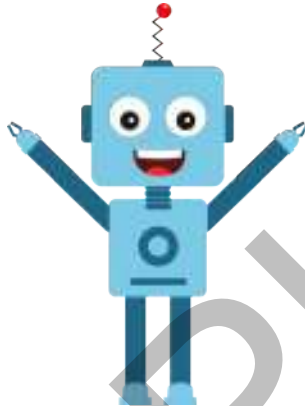
with the Starter Kit,
the perfect companion for an adventure like this.



evive

evive's Starter Kit

Beginning the adventure in 3, 2, 1... Now!



Copyright © Agilo Research Pvt. Ltd., 2019

Contact Us

Agilo Research Pvt. Ltd.
215, Sahajanand Arcade, Helmet Cross
Memnagar, Ahmedabad, Gujarat
India – 380 052

Email: contact@thestempedia.com

www.thestempedia.com

All rights reserved. No part of this publication can be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronics, mechanical, photocopying, recording or otherwise, without prior written permission of the publisher. Some graphics are modified from content created by Freepik. All disputes are subject to the jurisdiction of competent courts in Ahmedabad, India.

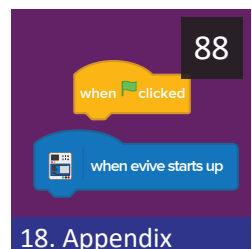
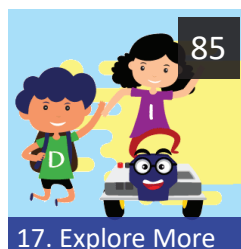
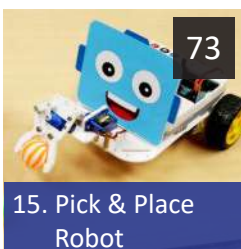
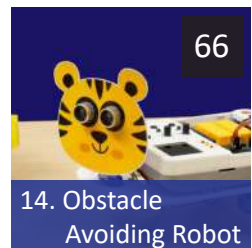
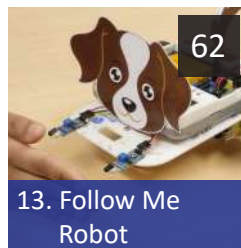
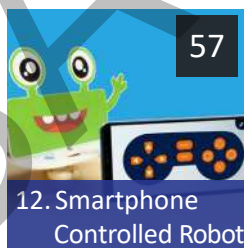
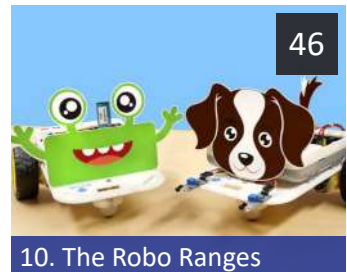
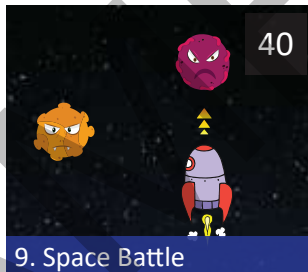
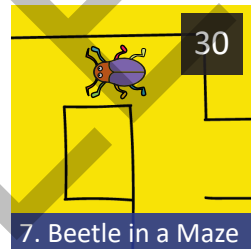
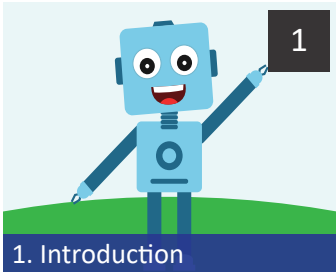
Due care and diligence has been taken while editing and printing the book. Neither the author nor Agilo Research hold any responsibility for any mistake that may have crept in inadvertently.

Please give your suggestions and feedback at contact@thestempedia.com

Second Edition 2019.

Printed in India

Content



Component List

8 x Light Sensor [Photo-resistor]



10 x Switch [Pushbutton]



LEDs

5 x Blue LED, 5 x Red LED

5 x Yellow LED, 5 x Green LED



2 x Micro Servo Motor
(Includes 3 servo horns and self-threading screws)



2 x Capacitor [220 or 100 μ F]



Resistors

15 x 220 Ω Resistor, 10 x 1 k Ω Resistor

10 x 4.7 k Ω Resistor, 15 x 10 k Ω Resistor

10 x 1 M Ω Resistor



6 x Alligator Clips



1 x Joystick



1 x Bluetooth Module



Jumper Wires

20 x 20 cm Jumper Wire (Male-Male)

20 x 20 cm Jumper Wire (Male-Female)

20 x 20 cm Jumper Wire (Female-Female)



1 x Distance Sensor [Ultrasonic Sensor]



1 x Distance Sensor Holder



2 x IR Sensor



2 x Dual Shaft Motor (with 2 Brackets)



2 x Wheel



1 x Base Plate [Chassis]



1 x Base Servo Motor Holder



1 x Servo Mounting Plate



1 x Servo Motor Holder



1 x Light Tracker Plate 1

1 x Light Tracker Plate 2



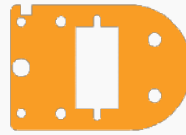
1 x Gripper Claw 1



1 x Gripper Claw 2



1 x Gripper Base Plate



1 x Gripper Plate 1



1 x Gripper Plate 2



1 x Gripper Link



Nuts and Bolts

- 6 x M2 bolts of 12 mm Length
- 25 x M3 bolts of 8 mm Length
- 10 x M3 bolts of 12 mm Length
- 6 x M3 bolts of 25 mm Length
- 4 x M4 bolts of 16mm Length
- 6 x M2 Nut
- 25 x M3 Nuts
- 4 x M4 Lock Nuts



Spacers [Standoff]

- 2 x 15mm M3 (Male to Female)
- 5 x 20mm M3 (Female to Female)
- 4 x 30mm M3 (Male to Female)



Cable Ties



1 x Castor Wheel



1 x Screw Driver
(+ & - Type)



evive (Brain)



Young Makers' Guide (Book)





CHAPTER 1 Introduction

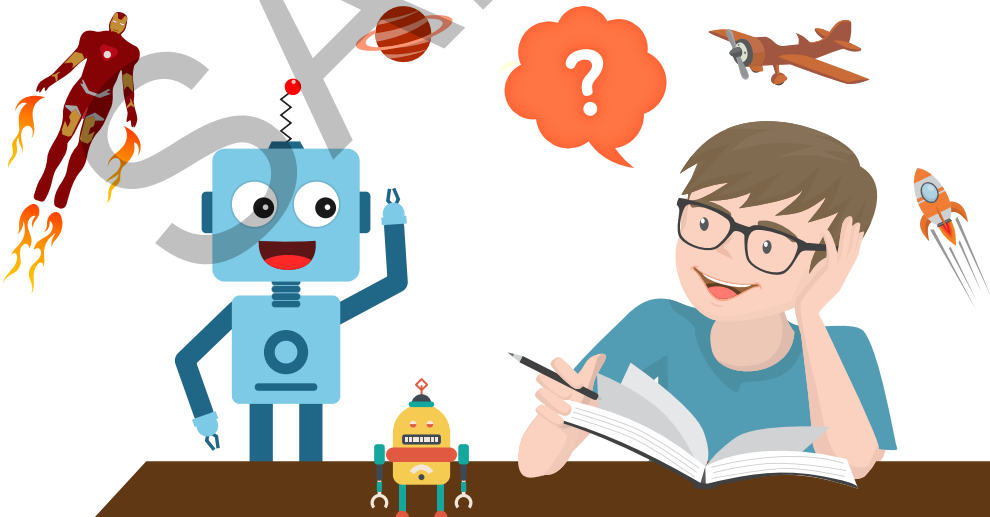
Dear Youngster,

Today, we live in a world full of electronic gadgets, robots, space missions, drones and what not. In a time when technology has taken root in every aspect of our lives, you, being the future, are destined to drive its progress.

Every now and then you must have felt curious about how things work, inculcating in you, a desire to innovate, stimulating your creative mind to come up with innovative ideas to solve everyday problems, to make your life simple and fun, like a fancy Iron Man dream.

But Iron Man didn't become a great innovator overnight; he started from scratch, fell in love with technology, and kept innovating. What if you could make such marvels for yourself, and grow to become the Iron Man?

Sounds exciting, right? So what are you waiting for, start right away and Do-it-yourself!



The All New 'Maker' Culture

Innovators, like you, who make exciting, out-of-the-box things are called makers. A maker is curious, loves technology, and is on a constant lookout for innovative solutions to everyday problems. He ideates, creates, implements, and brings a revolution. When you become a maker, you become a very important contributor to humanity's technological progress.

In this book, we'll take you on an adventurous journey through the peninsula of DIYing and help you become a maker by developing important skills in electronics, robotics, and programming: the important pillars of today's technology.

Once you finish making everything in this book, we'll honor you with the title 'STEMpedia's super-maker', because a maker today, a revolutionist tomorrow!

How Will This Help Me?

This book will take you through a series of exciting projects - from making animated games to beautiful art, from making fun electronic projects to complex robots. The skills you will develop while making them will help you interact with and control things around you; the aptitude gained will help you solve real-life challenges.

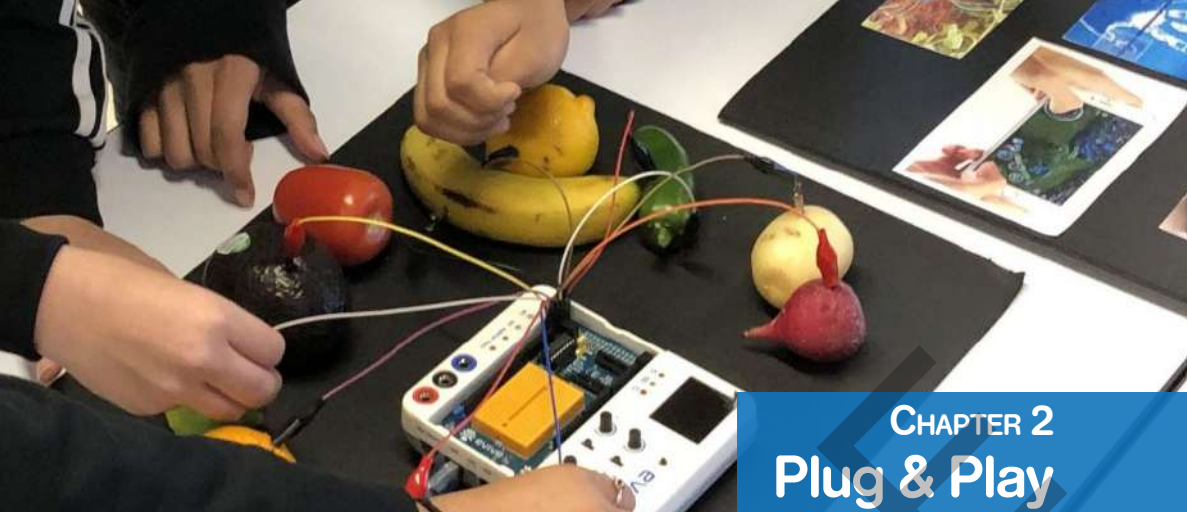
Youngsters like you are the most powerful beings among all humans. They are curious and can put in endless efforts to achieve what they love. Exploiting your interest in technology at an early age will pave way for a brighter future. By the time you become an adult, you will be well-informed, well-equipped, and will be set to perform better.

Pumped Up and Ready?

Your starter kit comes with **evive**, your versatile project-making buddy, and hundreds of components for your projects which you will use them in the coming projects. But how to use them? Don't you worry! You will learn that as well in their respective chapters.

So, ready to innovate? Grab your Starter kit and buckle up for an exciting journey of learning and building!





CHAPTER 2 Plug & Play

Introduction

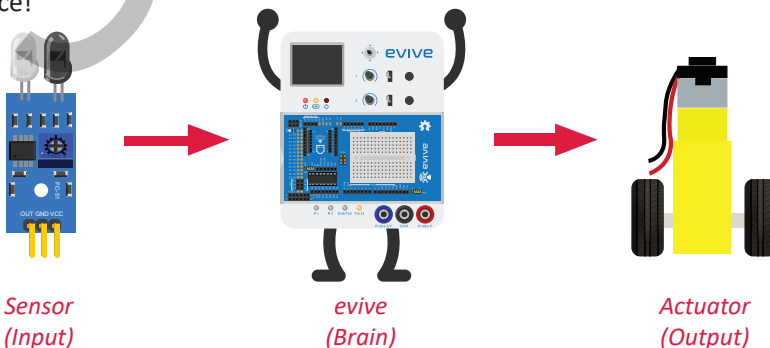
What controls all your movements, how you act and react? And who told you to choose nothing but the best for yourself, like the Starter kit, to take on the ride of your life in the world of DIYing?

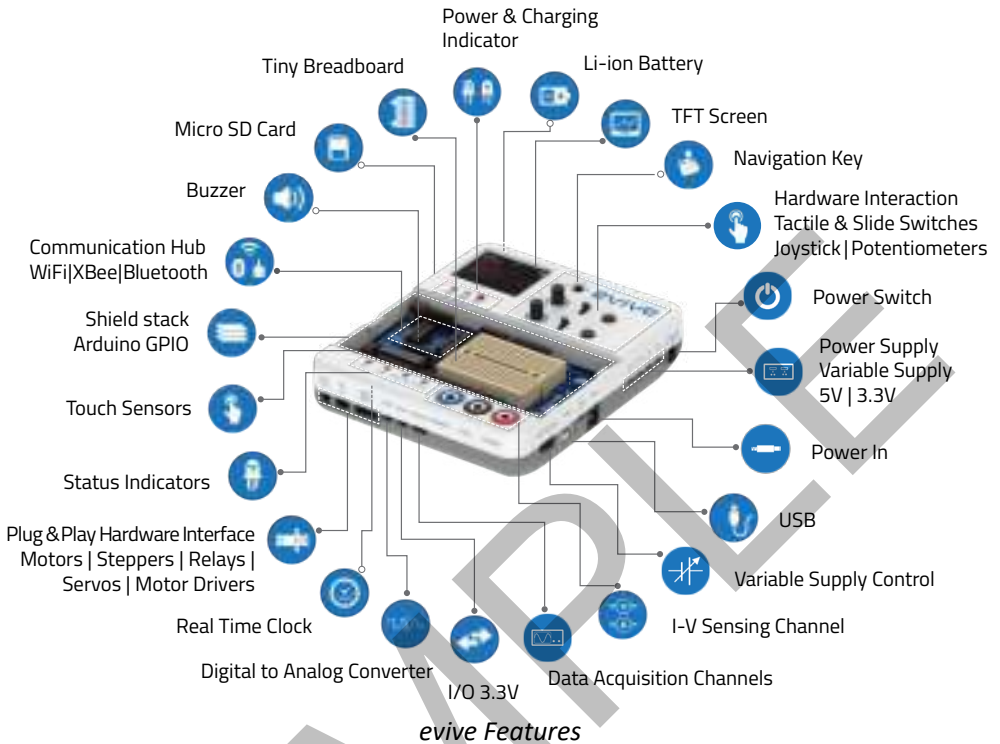
Your brain! Just like your brain controls your body and everything that you do, electronic circuits and robots also need a brain to control them. The Starter kit comes with one of the most hi-tech and intuitive e-brains out there - evive, a plug-and-play prototyping device that makes DIYing fun!

In this chapter, we will introduce you to evive, what it is, how it works, and how to use it for making cool projects and developing the 21st-century skills as you delve deep into project-making.

evive: Your Project's Brain

evive is THE project-making device you will ever need. It is an all-in-one STEM learning and prototyping platform that stays at the center of your projects and controls them. Its plug-and-play interface allows you to easily connect components such as motors and sensors to it and make complex projects with ease. It comes with a variety of features such as an inbuilt battery, display, Bluetooth compatibility, and prototyping essentials, and much more thus helping you make a wide range of projects with just one device!





evive Features

Setting up evive: The very first step is to install the battery. The Starter kit contains a leaflet which describes the whole procedure.

Switching on evive: Slide the Power switch to **INT**; the red power indicator LED will turn ON and a menu will appear on the screen.

Charging the Battery: Slide the power switch UP to **EXT** and connect evive to your computer via a USB cable. You can observe the charging status by monitoring the orange colored LED below the display screen. When evive is fully charged, the LED will stop glowing.



evive's Menu

Now that evive is ready for action, let us perform some fun activities!



If no menu appears on the screen, center press the navigation key. If it still doesn't appear, go to chapter 6 to see how to upload the firmware to evive. Then center press the navigation key; the menu will appear.

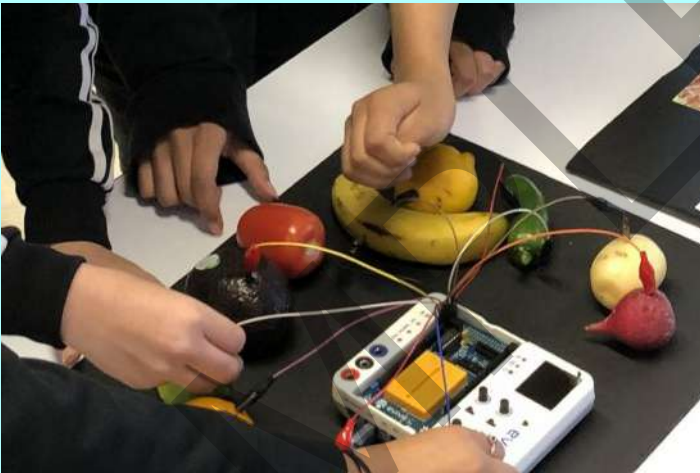
ACTIVITY

SINGING FRUITS



In this activity, we will make an eight-key DIY Piano using evive's capacitive touch sensors. They are the 12 pins present in the bottom left corner under the magic lid. evive's capacitive touch input monitors the capacitance level and notifies the user if the capacitance is above threshold (touched) or below it (not touched).

The sound will be generated by a buzzer which is in-built in evive. We will make a piano that plays the following notes - C4, D4, E4, F4, G4, A4, B4, C5. Let's begin!

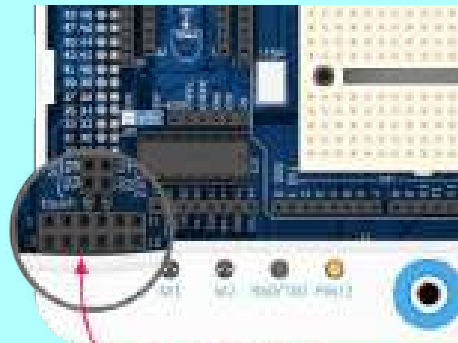


Component Required:

- ▶ evive
- ▶ Male-to-male jumper wires (8)
- ▶ Fruits/vegetables (8)

Steps

1. Take one male-to-male jumper wire and insert one of its ends in a fruit or vegetable.
2. Next, insert the other end inside evive's touch sensor 1.
3. Similarly, connect the other 7 fruits or vegetables.
4. Switch on evive. In the menu, navigate to **Touch Sensor** using the navigation key and touch any fruit/vegetable. You will hear its respective tone. Enjoy



12 Capacitive Touch Inputs

creating fruit and veggie symphonies!



You can use any conducting material with touch sensors like aluminum foil.



If no sound is produced, reset evive. While doing so, do NOT touch the fruits or jumper wires.

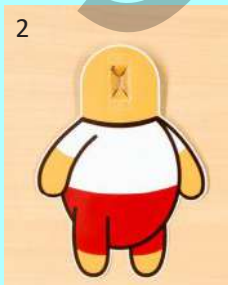
ACTIVITY TOBI'S HEAD MOVES



In this activity, we're going to move the head of a cutout of our learning buddy, Tobi with the help of a servo motor. We will fit the servo motor in the cutout of Tobi's head and control its movement using evive's potentiometer. Let's begin!

Component Required:

- ▶ evive
- ▶ Servo motor
- ▶ Screwdriver
- ▶ Tobi's cutouts
- ▶ Scissors





CHAPTER 5

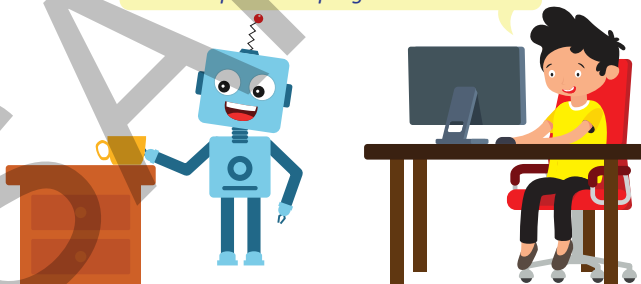
The Land of Code-o-Nuts

After the *Museum of Electronics* and the *Signals City* comes the *Land of Code-o-Nuts*, a place where you will learn the basics of programming and PictoBlox, a programming software that will make learning to program easy and fun for you! Let's get started.

What is Programming?

Programming is the process of writing a set of instructions in a definite order for the computer to follow. These instructions are called programs. Let's look at an example to understand it better. Imagine that you're making a robot for the biggest science fair in your school. After working on it for a long time, you start feeling tired and realize that you need something to boost your energy. But you can't leave work! Suddenly, an idea strikes your mind. You can instruct your robot to make you a cup of coffee by writing a program. Problem solved!

Time to upload the program to the robot!



Programming is an important skill to have today as it helps develop the must-have 21st-century skills such as logical reasoning and problem-solving and prepares budding innovators like you for the future that is going to be loaded with technology.

PictoBlox: The Fun Way to Learn Programming

PictoBlox is a visual programming software that lets you write programs using graphical elements known as blocks. The blocks are like pieces of a jigsaw puzzle that you must put together to solve the puzzle. This approach makes learning to program easy, interactive, and super fun!

Blocks

Puzzle-piece shapes used to write scripts

Stage

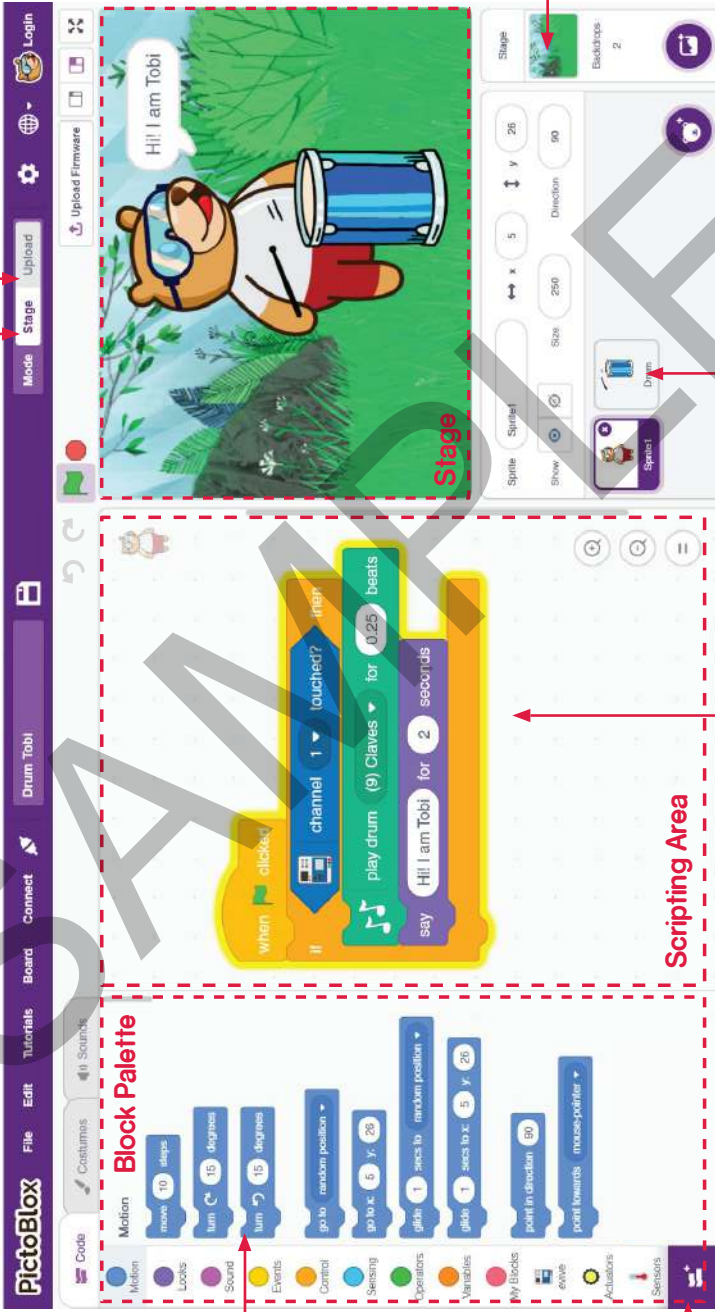
The area where the sprite performs actions based on the script

Stage Mode

Write scripts for the sprite and boards to interact with them in real-time

Upload Mode

Write scripts for the board to control them even when disconnected from the computer



Block Palette

Scripting Area

Add Extension

For adding new palettes to the block palette

Script

A stack of blocks interlocked with one another in a specific order to perform a task

Sprite

An object, or a character, that performs actions based on the script

Backdrops

Possible backgrounds of the Stage

Installation

Let's begin by installing PictoBlox. Follow the steps below to install Pictoblox on your computer:

1. Go to <https://thestempedia.com/product/pictoblox/downloads> or scan the QR code for detailed instructions.
2. Download the version compatible with your OS (Windows, macOS, Linux).
3. Run the .exe or .dmg file and sit back and relax!



Download PictoBlox

You are all set for your programming experience in PictoBlox. Let us take you on a tour of the software.

Block Palette

The block palette consists of different palettes such as **Motion**, **Sound**, and **Control**. Each palette has different blocks that perform functions specified by the palette name.

Below is the list of palettes and the functions their blocks perform:

Palettes	Color	Function
Motion	Medium Blue	To control sprite's movement
Looks	Violet	To control sprite's appearance
Sound	Light Purple	To play sound. Explore Sounds tab to record, upload a music or edit sounds.
Control	Burnt Orange	To control the flow of script (using conditions)
Events	Occur Yellow	To sense events, which trigger scripts to run. These are essential for every project
Sensing	Light Blue	To detect different factors of a project
Operators	Light Green	To do math calculations and handle strings
Variables	Orange	To hold values and string in variables and display them on the screen
My Blocks	Light Pink	To make custom blocks
Pen	Green	To control the pen that allows sprite to draw shapes
evive	Blue	To control evive board
Actuators	Light Green	To control motors such as servos and DC motors
Sensors	Light Blue	To get readings from sensors
Dabble	Purple	To enable projects to communicate with Dabble
Display	Light Red	To control display screen on boards

There are other block palettes which can be loaded from **Add Extension** button located in the bottom-left corner.



*In this book, blocks will be mentioned in **bold** letters, block palettes or software buttons in **bold italics**, and dropdown options in italics.*

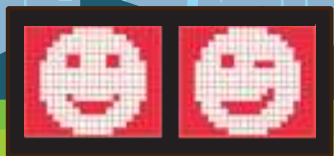
Modes

PictoBlox has two modes:

1. **Stage Mode**: In this mode, you can write scripts for the sprite and boards like evive to interact with sprites in real-time. If you disconnect the board with Pictoblox, you cannot interact anymore.
2. **Upload Mode**: This mode allows you to write scripts and upload it to the board so that you can use even when it is not connected to your computer, for example, you need to upload a script for making moving robots.

Conclusion

In this chapter, you learned about programming and PictoBlox, a graphical programming software that makes programming fun. In the next chapter, you'll learn to write scripts in PictoBlox!



CHAPTER 6

Programming with PictoBlox

In the previous chapter, we learned what programming is and explored PictoBlox. Now, let's dig deeper into the world of programming and experience it first-hand with three short and sweet activities. Let's begin!

ACTIVITY

JUMP USING THE SPACE KEY!

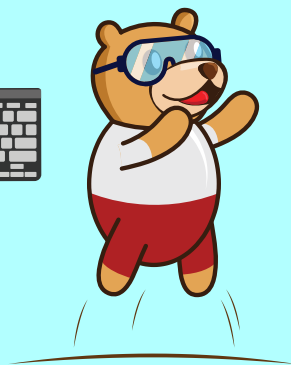


In this activity, we will make Tobi, our coding partner in PictoBlox, jump using the space key on the keyboard. We're going to write this script in the **Stage** mode.

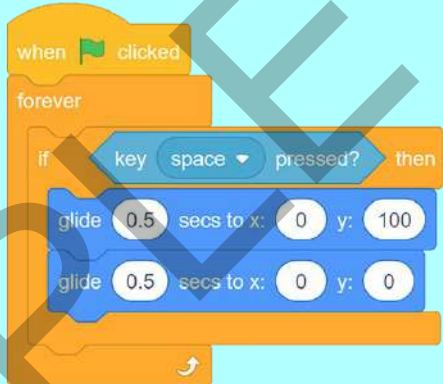
Steps

Follow the steps below:

1. Open Pictoblox and try to recognize all the sections you saw in the previous chapter on *page 22*.
2. Drag and drop the **glide () secs to x:() y:()** block in the scripting area. Set the seconds as 0.5 secs, x as 0 and y as 100. Now, click on the block; you will observe that Tobi glides in upward direction.
3. Now, to bring him back to its initial position, we need another **glide** block. So, duplicate it by right-clicking on it and drop it below the previous **glide** block; change y to 0.
4. Now, click on the two blocks. You will see Tobi jump.



- How do we make Tobi jump using the space key? Using the **key () pressed** block from the **Sensing** palette! And to check whether the space key has been pressed or not, we need an **if** block from the Control palette. Drag and drop both the blocks in the scripting area and arrange them as shown in the figure.
- Now, we want to check continuously whether the space key is pressed or not. For that, we're going to use the **forever** block from the **Control** palette.
- As of now, we have to manually start the script to make it run. To trigger it with a particular event, there are hat blocks in PictoBlox. Generally, in most scripts, the green flag above the Stage is used as a trigger. So, drag and drop the **when clicked** block from the **Events** palette above the **forever** block. That's it!
- Maximize the Stage, click on the green flag to run the script, and make Tobi jump!



ACTIVITY JUMP USING EVIVE!

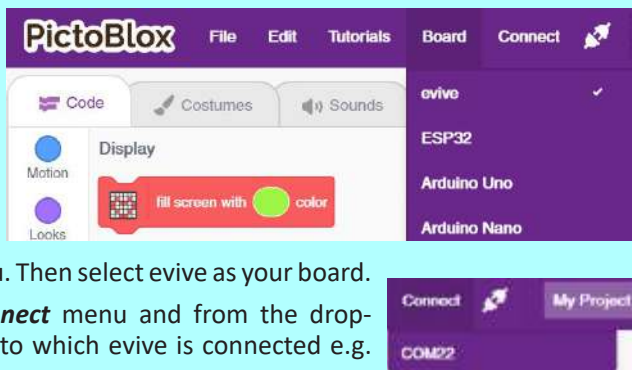


In the previous activity, we made Tobi jump using the space key in **Stage** mode. This time, we're going to do the same with evive, the e-brain. This is where you officially begin your journey as a young innovator!

Steps

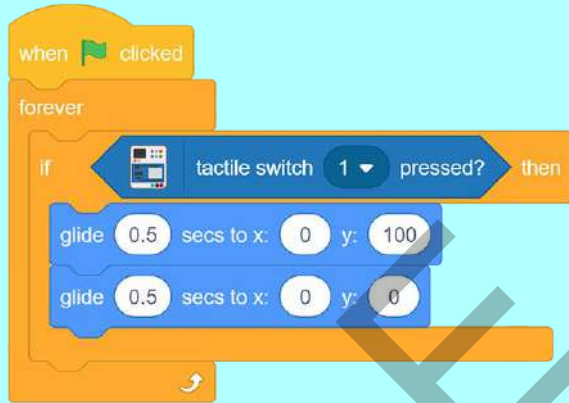
Follow the steps below:

- Connect evive to your computer using the USB cable and open PictoBlox.
- Go to the toolbar and click on the **Board** menu. Then select evive as your board.
- Next, click on the **Connect** menu and from the drop-down, select the Port to which evive is connected e.g. COMXX or ttyXX.



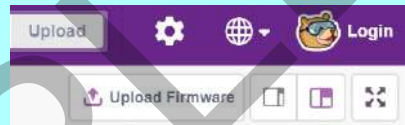
Once you select the port, the icon beside the **Connect** menu will become connected.

Most of the script is going to remain the same. All we need to do is replace the **key () pressed?** Block.



- Go to the evive palette and drag and drop **tactile switch () pressed?** block in place of the **key () pressed?** block in the **if** block.

- To work with evive in the **Stage** mode, we need to first upload firmware to it so that it can communicate with your computer in real-time. For that, click on the **Upload Firmware** button; you will find it above the top right corner of the Stage. The firmware will take a few seconds to upload.



- Click on the **Run** icon to run the script and press **tactile switch 1** on evive to make ToBi jump!



To see if the communication between PictoBlox and evive is happening or not, you can watch the Rx/Tx LED on evive. If it is blinking, all is OK; otherwise, remove the USB cable and reconnect evive.

ACTIVITY MAKE EVIVE EXPRESSIVE!



We wrote the scripts of the previous activities in the **Stage** mode; however, those scripts won't work when you disconnect evive from your computer.

But don't you worry! We have a solution for that. Put your hands together for the **Upload** mode! Writing and uploading the script in this mode lets you take evive

```

Mode Stage Upload
Upload Code
#include <evive.h>
TFT_ST7735 lcd = TFT_ST7735();
void _setup() {
  lcd.init(INITR_BLACKTAB);
  lcd.setRotation(1);
}
    
```

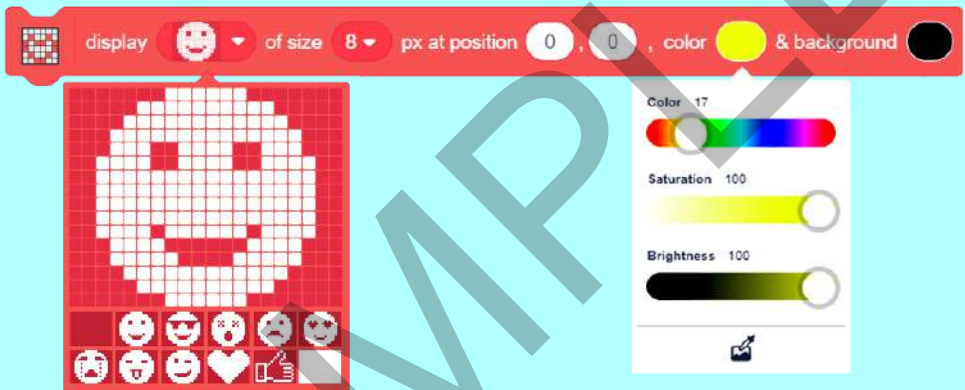


to your friend and show the cool stuff you do or program a robot and control it independently.

In this activity, we are going to write a script to display emojis on evive's TFT display. Let's begin!

Steps

1. Connect evive to your computer and choose the appropriate port in PictoBlox the same way we did in the previous activity. Switch to **Upload** mode by toggling the mode button. You will observe that the Stage is replaced by an editor window.
2. Go to the **Display** palette and drag and drop the **display () of size () at position () () color () & backdrop ()** block in the scripting area.
3. Select or create an expression in the drop-down.




4. Next, set the size as 8, the position as 0, 0, color as yellow, and background as black by moving the sliders in the drop-downs.



5. Go to the **Control** palette and drag and drop a **wait () seconds** block below it.
6. Duplicate all the blocks by right-clicking on them and snap them below; change the expression from the drop-down of the second **display** block.
7. Add the **forever** block from the **Control** palette to run the whole script continuously.
8. To start with a blank screen, add a **fill screen with () color** block above the **forever** block and set the color as black from the drop-down.

Now comes the most important step:

9. Go to the **evive** palette and drag and drop the **when evive starts up** block on top of the script. It is a must for uploading the code to the evive and running it after disconnecting.
10. To upload the code, click on  **Upload Code** button . It will take a few seconds.
11. Remove the USB cable, power on evive, and enjoy!



If uploading fails, then remove the USB cable, reconnect evive and try again.

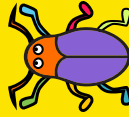
Conclusion

In this chapter, you had your first hands-on experience with PictoBlox with three easy and fun activities in which you worked in both the Stage and Upload modes. However, the real fun will begin in the coming chapter where you will get to create your very own game! See you there! Explore more activities based on PictoBlox available in the tutorial hub by clicking on the **Tutorial** tab in the menu bar.



CHAPTER 7

Beetle in a Maze



The time has come to do something super exciting! We are going to create an interesting game in PictoBlox! The aim of the game is to help the beetle reach the apple at the end of the maze without touching the walls. We are going to use evive's navigation key to control the beetle. Let's begin!

Bringing in the Beetle

PictoBlox has an inbuilt library from where you can choose any sprite you want. Follow the steps below to bring in the beetle:

1. Right below the stage, you will see a purple button with a bear on it. Take the cursor over it and select **Choose a Sprite**; the sprite library will open.
2. Select the beetle sprite from the library.

Choose a Sprite

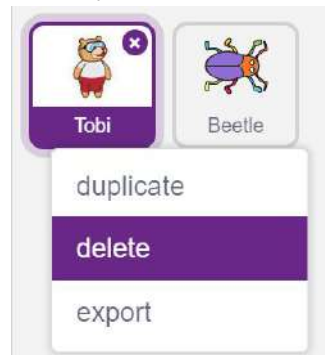
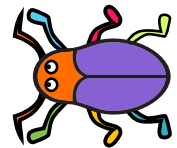


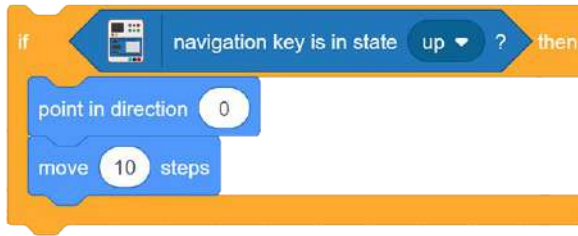
Now, there are two sprites: beetle and Tobi. Right click on Tobi's icon and click on the delete option.

Giving the Beetle Its Moves

Let's write the script for the beetle. Since we're controlling it using evive's navigation key, the beetle must move in the direction we push the key. Let's begin!

1. Connect evive to your computer and open PictoBlox. Then, select evive as your board from the **Board** menu in the menu bar and select the appropriate port from the **Connect** menu. In doing so, some palettes specific to evive will appear in the block palette.
2. To make the beetle point in a direction, we will use the **point in direction ()** block from the **Motion** palette. Drag and drop it in the scripting area and write **0** in the space. **0** corresponds to the upward





direction.

3. To move the beetle, drag and drop **move () steps** below the **point in direction (0)** block. Now, click on these blocks; the beetle will move 10 steps forward.
4. From the **evive** palette, drag and drop the **navigation key is in state () ?** block and select up from the drop-down.
5. To make the beetle move forward only when we press the navigation key, we will use the **if** block from the **Control** palette.
6. Place the **navigation key is in state () ?** block inside the space in the **if** block. Place the **point in direction ()** and **move () steps** blocks inside the C shape of the **if** block.
7. The steps for writing the script for the down, right, and left directions are similar to those for right. So, instead of dragging and dropping each block, just duplicate the script as shown by right-clicking on it and make changes in the state in the **navigation key is in state () ?** and **point in direction ()** blocks as given below:
 - ▶ For 'down', set the direction as 180.
 - ▶ For 'right', set the direction as 90.
 - ▶ For 'left', set the direction as -90.



8. Now stack all the **if** blocks together.
9. To make evive continuously check and execute these instructions, drag and drop the **forever** block from the **Control** palette above the script we've written so far.
10. Finally, go to the **Events** palette and select the **when clicked** block; place it above the **forever** block. The script is ready!



- To test the script, upload the firmware to evive using the [Upload Firmware](#) button and click on the green flag . The beetle should move as you press the navigation key.

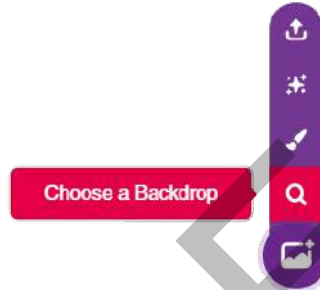
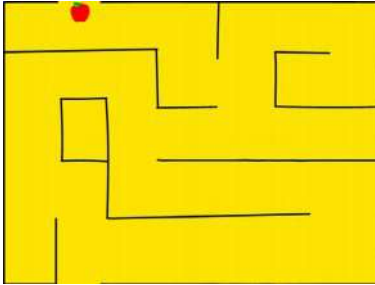


NEVER centre press the navigation key. Otherwise, communication between evive and your computer will stop. In case you press it by mistake, stop the script, reset evive, and run the script again.

Setting Up the Stage

PictoBlox has an inbuilt library from where you can choose any backdrop you want. Follow the steps below to add the maze backdrop:

1. Click on the purple colored icon which says **Choose a Backdrop**.
2. Select the maze backdrop from the library.



Whenever you add a new backdrop or a sprite, the scripting area becomes empty. To get back to the script you were writing, click on the icon of the sprite you were working on, in our case the beetle.

Sensing the Environment

Now, it's time to write the script for safely taking the beetle to the end of the maze. Follow the steps below to write the script:



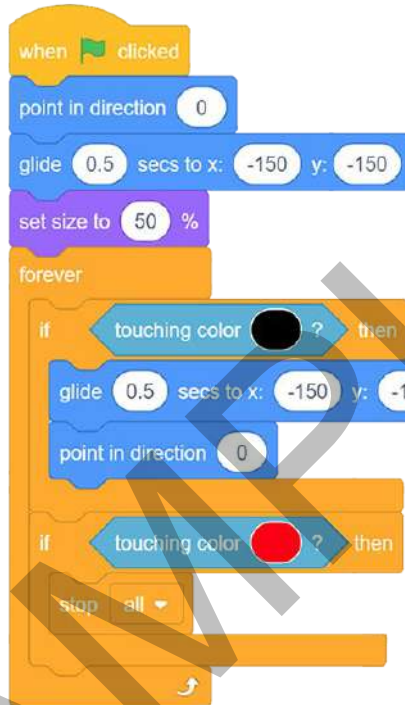
Make sure that the beetle's icon is selected in the sprite palette.

1. Go to the **Events** palette and drag and drop the **when clicked** block in the scripting area.
2. From the **Motion** palette, drag and drop the **point in direction ()** block below the **when clicked** block.
3. Then, drag and drop the **glide () secs to x: () y: ()** block and set the seconds as 0.5, x as -150 and y as -150. The x and y values are the coordinates of the beetle's initial position.
4. Drag and drop the **set size to () %** block from the **Looks** palette to adjust the beetle's size.
5. Whatever we write next must execute continuously because we want to keep track of whether the beetle touches the walls or not. So, drag and drop a **forever** block.
6. Drag and drop an **if** block inside the **forever** block and the **touching color () ?** block from the **Sensing** palette inside the space of the **if** block. This block will help us check if the beetle touches the walls. Select the **black** color of the boundaries from the drop-down.



7. Whenever the beetle touches any wall it should return to its initial position and if it safely reaches the end, i.e. touches the color **red**, the game should stop. For that, we will use the **stop ()** block.

Below is the complete script.



Enjoy the game!

Facing any difficulties? Scan the QR Code.

Conclusion

In this chapter, you made your very first game in PictoBlox! In the coming chapter, we're going to make another interesting game which will be a little challenging than this one and double the fun!





CHAPTER 8

The Hungry Shark

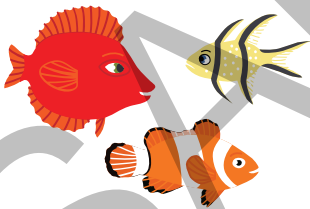
Sharks are hungry creatures. And so is the one in our game! The aim of the game is to feed the shark as many fish as you can by controlling it with evive's potentiometer. More the number of fish it eats, higher will be the score.

Setting Up the Stage

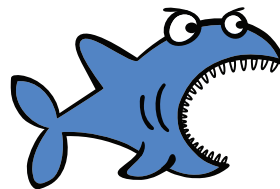
Since we have a shark and a lot of fish in our game, our background should be a suitable environment for them. Therefore, choose any of the underwater backdrops from PictoBlox's backdrop library as we did in the previous chapter.

Bringing in the Sprites

In this game, we need two sprites: the shark and the fish. Open the sprite library as we did in the previous chapter and choose the **fish** and **Shark2** sprites.




Fish



Shark2

Assigning Movements to Shark

Follow the steps below to write the script:

1. Connect evive to your computer and open PictoBlox. Then, choose it as your board and select the appropriate port from the **Connect** menu. Don't forget to upload the firmware because we are working in **Stage Mode**!
2. Drag and drop the **when**  **clicked** block to scripting area from the **Events** palette.
3. Drag and drop the **set size to () %** block from the **Looks** palette to adjust the shark's size. Set it as 80% of the original.



CHAPTER 10

The Robo Ranges

Welcome to the Robo Ranges! In this chapter, we are going to talk about robots, specifically mobile robots, and make one of our own! But, what IS a robot?

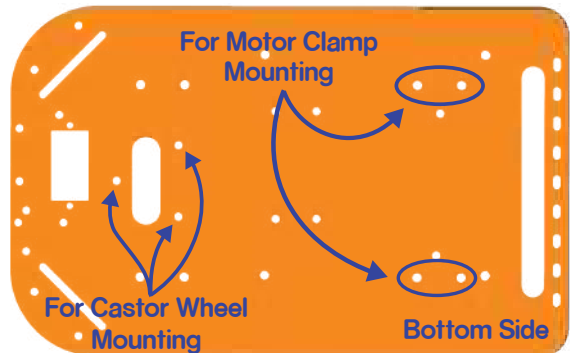
A robot is a machine that can take decisions and perform tasks on its own but only after being told how to using programming. Mobile robots are robots that can move from one place to another just like us. Though making a robot like us will take a lot of knowledge and efforts, what you will be learning here will give you a good idea of robotics.

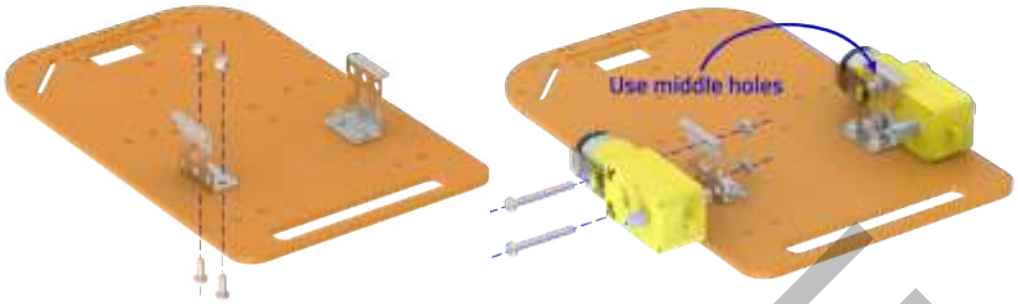
In this chapter, you will learn how to assemble and test a mobile robot. In the coming chapters, you will learn how to control it using different methods such a Joystick and a Smartphone. Then, we will provide some senses to the robot and make it autonomous to make an object-following robot and an obstacle avoiding robot. Lastly, we will make a pick and place robot which will have a gripper mounted in front of it, just like your arms. Let's begin!

Assembly

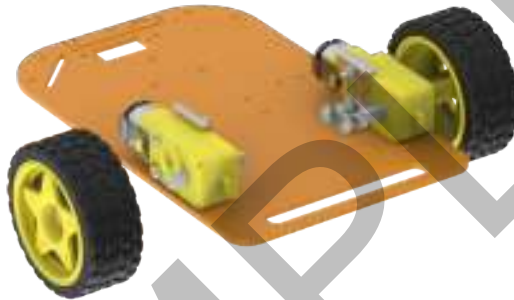
The annotated base is shown on right. This is the bottom side. There is evive's logo on the top side for your reference. So be careful while using the base.

1. Fix the motor mounting brackets to the chassis using **M3 bolts of 8mm length** and **M3 nuts**. Be careful about the holes you are using; refer to the figure of the base plate.
2. Attach the two motors, one to each bracket, side by side and fasten using **M3 bolts of 25mm length** and **M3 nuts**. Observe the *placement* and *orientation* in the figure given below (right) carefully.
3. Now, fit the wheels into the protruding motor shafts. The motor shafts would fit



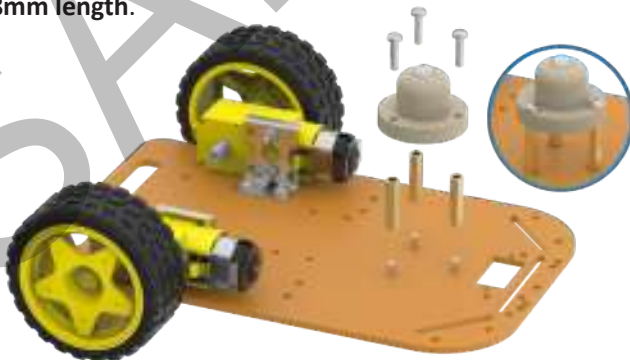


perfectly into slots provided for them on the wheels. If required, adjust the motor brackets.



The motor wires should be inside, otherwise they will collide with the wheel. Also, while assembling the robot, be careful about the holes you are using. Fasten all the nuts and bolts tightly for smooth functionality.

4. We'll attach the Castor wheel now. Remember, any body needs a minimum of 3 contact points to be stable. First, we will mount the M3 standoffs (20 mm) on which the Castor will be attached. Fasten the standoffs to the chassis using **M3 bolts of 8mm length**.



5. Place the Castor on top of the standoffs in the configuration shown and fasten using **M3 bolts of 12 mm length**.



Keep in mind, that what we have assembled so far (motors and castor) will be pointing downwards.

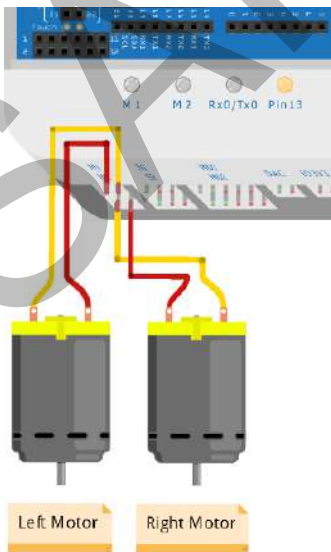
6. Flip the assembly and place evive on the top of the chassis.
7. Using the holes on the back of evive (align evive so that you can see at least 3 holes matching), fasten it to the chassis using **M3 bolts of 12mm length**. There are nuts already built in evive, hence you just have to fasten it using bolts. See the figure in the case of any confusion.



Your basic mobile robot is now ready. This is the framework for almost every robot that we'll work with. Modify it according to your need to use it as a Bluetooth controlled robot, obstacle avoiding robot, or light-tracking robot.

Connections

The circuit of the mobile robot is very simple; you just have to connect the left motor to motor channel 1 (marked as **M1**) and the right motor to channel 2 (marked as **M2**).



Testing and Debugging the Robot

Follow the steps below:

1. Power on evive. In the menu, navigate to **Controls** → **Motors** → **Motor 1 & 2**.



Center-press the navigation key, if the menu is not visible. Otherwise, upload the firmware using PictoBlox as done in chapter 6.

2. Adjust VVR to maximum by rotating the **VAR VOLTAGE** knob in the upward direction.

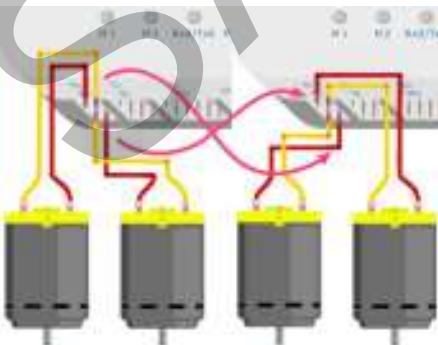


VVR will be displayed at the top of the screen.

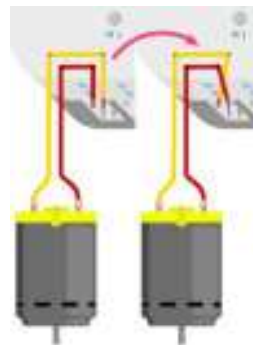


3. Control the motors using the slide switches and potentiometers as we did in the second chapter while making the Turntable.

When both the slide switches are in the upward direction, the wheel connected to the left motor should rotate anticlockwise and that connected to the right motor should rotate clockwise when you look at them from the sides. If not, then reverse the connections of the motors as shown below:



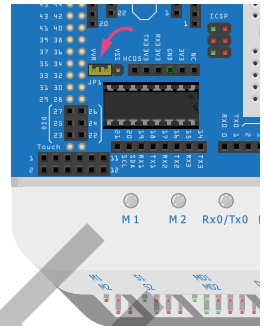
Exchanging the left and right motors by exchanging their pair of wires



Reversing the direction of a motor by swapping its wires



Keep the VSS-VVR jumper (JP1) on the VVR side, otherwise your motors will not run on internal battery.

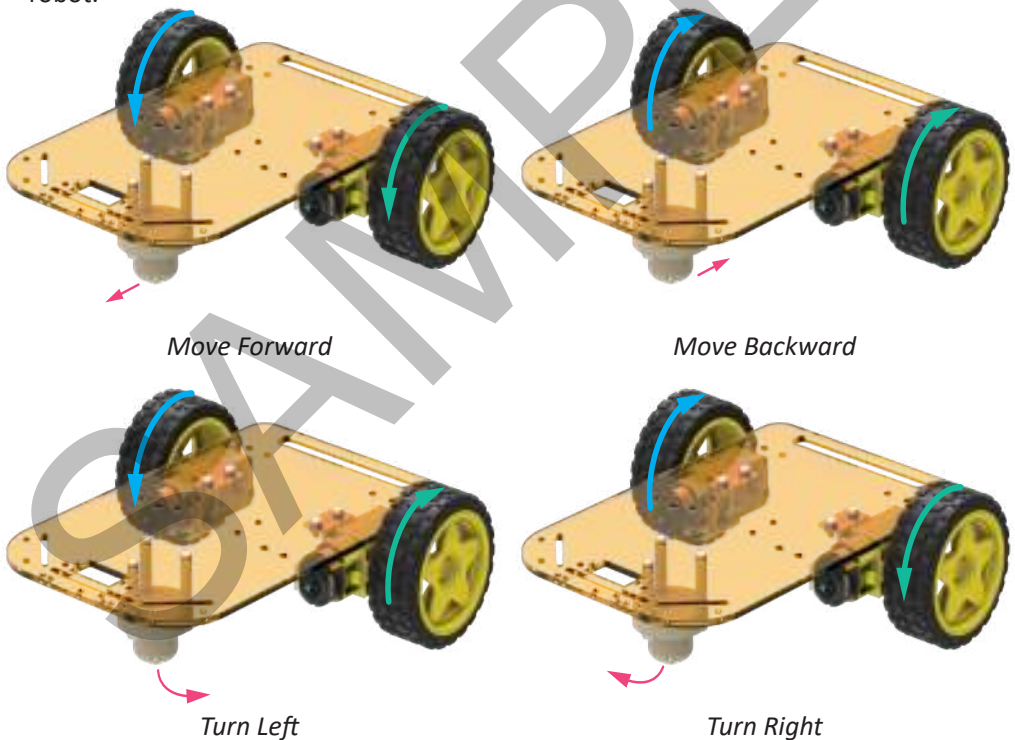


Differential Drive: The Concept Behind Movement

A differential drive robot is a mobile robot whose movement is based on two separately driven wheels placed on either side of its body. It can change its direction by varying the relative speed of its wheels and hence does not require an additional steering motion as we see in our cars.

If both the wheels are driven in the same direction and speed, the robot will go in a straight line. If both the wheels rotate with equal speeds in opposite directions with respect to each other, the robot will rotate.

Our robot has a configuration similar to that of a three-wheeled differential drive robot.



Conclusion

In this chapter, you made your very first robot! In the coming chapter, you will learn how to control it using a Joystick. See you in the next chapter!



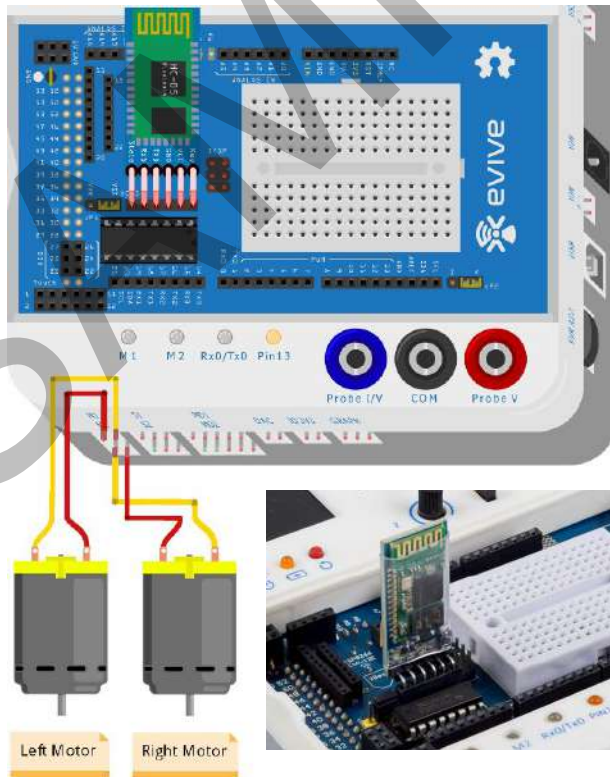
CHAPTER 12

Smartphone Controlled Robot

In this chapter, we're going to write a script in PictoBlox to control the mobile robot we built when we entered the Robo Ranges wirelessly, using a Smartphone and decorate it using the cut-outs given in the kit. Let's begin!

Connections

For this robot we have to connect the Bluetooth module (HC05/HM10). In the following figure you can observe where the Bluetooth module's 6 pin connectors are plugged on evive.

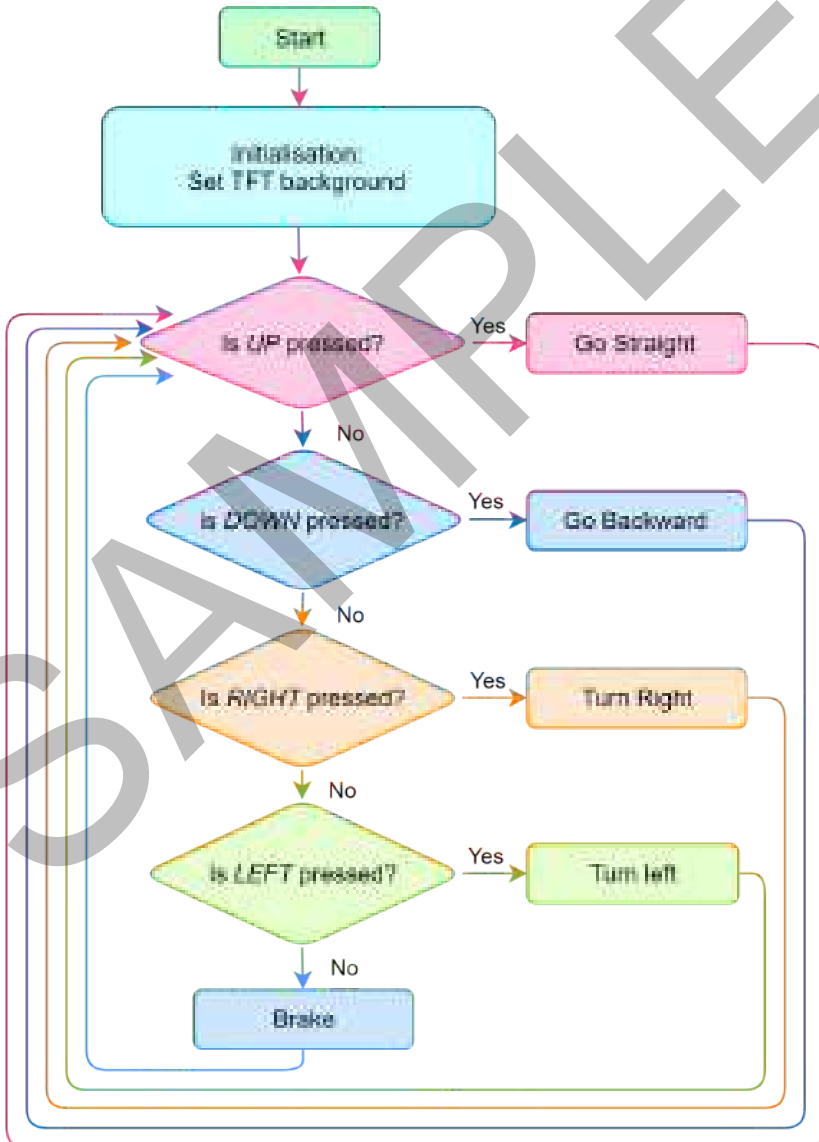




Bluetooth module's **RX** pin will go to **TX3V3** pin on evive and others correspondingly. Note that on powering evive, a red LED will start blinking on module. If you do not connect the module correctly, it may get damaged.

Logic and Flow Chart

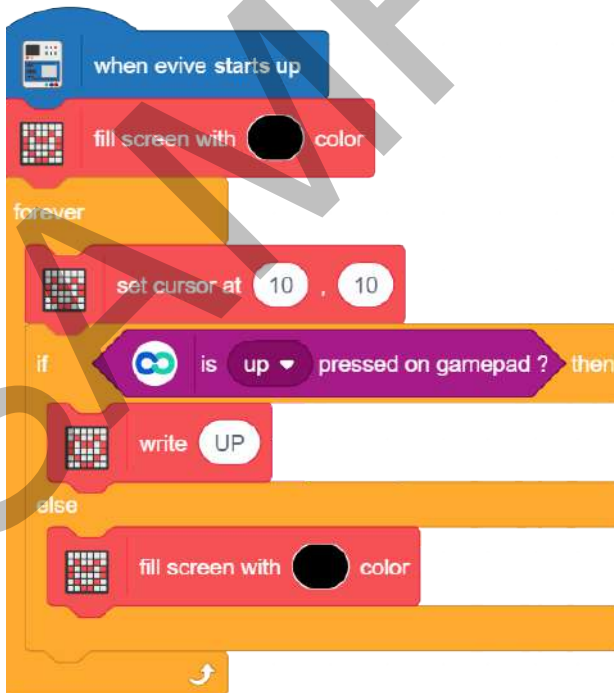
In this case we will communicate wirelessly. The user gives instructions via the Gamepad module in the *Dabble* app. The robot takes actions according to the button pressed. For example, if **Down** is pressed the robot will move backward; if none of the buttons are pressed, the robot will stop. Flowchart is shown below.



Scripts

Let us follow the following steps to make the code:

1. Select the **Upload** mode from the menu bar. Then, select the board as **evive**. Now, from evive palette, drag and drop the **evive starts up** hat block into the scripting area.
2. Set the TFT screen background to black using the **fill screen with (black) colour** block.
3. Add a **forever** block; everything inside will repeat forever.
4. Set the TFT screen cursor to 10, 10 using **set cursor** block.
5. Drag and drop the **if-else** block from the **Control** palette below the **set cursor** block. Next, drag and drop the **Is () pressed on gamepad?** block inside the diamond-shaped space in the **if** arm of the **if-else** block and select UP from the drop-down menu.
6. Then, drag and drop the **write ()** block below the **if** arm to display and write UP in its space. If the button is not pressed, then the screen should remain blank. Below is the test script. For testing it, connect evive to your computer and select the appropriate port from the **Connect** menu. Then, upload the script to evive.



7. After uploading, you need to pair the HC05/HM10 Bluetooth module with your Smartphone. Download and install *Dabble* using the QR code shown on the next page or from Play Store.



Download Dabble



Gamepad module in Dabble

8. Open Dabble and click on **Connect-Disconnect** icon and select the appropriate device from the list. You can find it by matching the last few characters of the ID written on the Bluetooth module. After successful connection, a message will be shown for the same. If not, then go to the Bluetooth settings of your phone and pair using default passwords (**1234** or **0000** or **123456** or **000000**). Then, go to the Gamepad module.
9. Click on the UP button; you will observe UP printed on the TFT screen. This means your communication is working and we are good to go.
10. Now, we have to assign action according to the button pressed. For that, we will use multiple **Is () pressed on gamepad?** and **if-else** blocks.
11. To make the robot move, create blocks named **Go Straight**, **Go Backward**, **Turn Right**, **Turn Left** and **Brake** as done in chapter 11.
12. The complete script is shown on the next page.
13. After completing the main script, upload the code to evive and run the robot.



You can add extra batteries to speed up your robot.

Scan the QR code to download the complete PictoBlox code.





Explore More!

Congratulations! In these 16 chapters, you've performed 13 exciting activities, created 3 games, and made 6 robots, i.e. 22 projects in total! But the buck doesn't stop here. There are loads of other projects that you can make using the evive Starter Kit; the possibilities are truly endless! Here are several different types of projects that you try your hands on!

DIY Activities

If DIY activities are totally your thing, try out these interesting activities at your school or home:



Touch-Based Piano

The piano that you made when you first used evive was a very simple one where all you had to do was plug the fruits and play the notes. You can try doing the same but this time by writing a script in PictoBlox.



Intrusion Detection System

The intrusion detection system is an interesting application of the ultrasonic sensor and evive's buzzer. Whenever the ultrasonic sensor detects someone lurking around, evive will sound the buzzer!



Smart Trash Bin

Managing garbage will now be an easy and clean affair with the smart trash bin project. This bin isn't any normal bin; it can sense when you're going to put trash in it and keep track of how much trash it holds.



Automatic Railway Crossing

Train sets are one of the most enjoyable toys. How about adding a DIY twist to it so that you can also learn while playing with it? The DIY automatic railway crossing gives an idea of how railway crossings work whenever a train passes by.

Science Experiments

You can even perform physics experiments with the Starter Kit! Who thought science could be this interesting?



Understanding Electromagnetic Induction

Ever wondered how electricity, on which most of our lives depend, works? Explore the principle that generates the electricity that powers up our world with the help of simple and fun science experiment.



Visualising an RC Circuit on evive

This project is for the Physics enthusiast in you! Learn the basics of an RC circuit, visualize its charging and discharging, and its response to AC supply using evive's oscilloscope and beta function generator.

Advanced Games

There's no doubt that the games you made in this book were beyond exciting. How about leveling up and make some advanced games?



Shoot the Bat

The name of the game says it all: you have to aim at the (not so) nasty-looking bats that appear from anywhere in space. The more bats you aim at in the given time, the higher will be your score.



Dodge

The Dodge game is one of the classic running games. The aim is to control the kid and dodge the objects coming his way for as long as possible. The more obstacles you avoid, the higher will be your score.

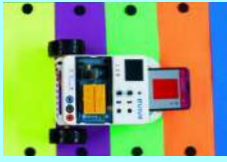
Advanced Robots

Now that you're familiar with robots and how to make them, how about trying hands on some complex ones?



Soccer Playing Mobile Robot

Playing soccer is fun, but playing soccer with robots is awesome! Gather your friends, team up, and make your own Smartphone-controlled soccer robots to challenge each other to an exciting game of robot soccer.



Color Detection Robot Using Smartphone

There are robots that can detect colors those that can play music. But this robot can detect colors AND play different sounds depending on the colors! All you need are the Starter kit, PictoBlox, Dabble, and loads of DIYing!



Gesture Controlled Robot Using Smartphone

Till now you made robots which used hand-held controllers. This time, control your robots with the help of hand gestures. Turn your wrist forward and the robot moves forward; turn it back and the robot moves back!



Line Following Robot

If you have seen WALL-E, then you have seen line following robots (remember those cleaning robots who follow a green line?). A line follower, as the name suggests, follows a line to move forward by using IR sensors.



Line Follower Robot Using Smartphone Camera

This one's similar to the previous robot. The only difference is that it uses a Smartphone's camera to detect the line. Dabble's camera module will control the Smartphone's camera and assist the line follower in its mission.



RGB LED Hand Band

Using the RGB LED strip given, you can make this cool hand band, which is going to be like no other that you've ever worn before! Control it using evive to make cool patterns and wear it proudly as you show it off to your friends!



We hope your maiden journey through this adventurous peninsula of DIYing was amazing and that it made you fall in love with DIYing.














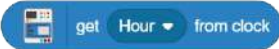
Keep tinkering and transform your ideas into reality!

- Team STEMpedia

APPENDIX

PictoBlox Blocks

evive Palette

	Every script to be uploaded to evive begins with this block.
	It checks if the specified tactile switch on evive is pressed. If yes, it returns <i>True</i> ; else <i>False</i> .
	It checks whether the specified slide switch is in the specified state. If yes, it returns <i>True</i> ; else <i>False</i> .
	It returns <i>True</i> if the navigation key is pressed in the specified direction; else <i>False</i> .
	It returns the analog value (0-1023) of the specified potentiometer of evive.
	It returns <i>True</i> if the specified touch input is touched; else <i>False</i> .
	It reads the state of the digital pin and checks whether the digital pin is HIGH or LOW.
	It reads the analog value (0-1023) from an analog pin on the board.
	It sets the output of the specified digital pin to the specified state (HIGH or LOW).
	It sets the output of the specified PWM pin to the specified PWM (square wave) value.
	It plays the specified tone on the piezo buzzer.
	It sets the time on RTC to the time specified.
	It sets the date and day on evive's RTC.
	It returns the calendar and clock data from evive's RTC.



get voltage reading from red channel

It measures the voltage between evive's red/blue and black channel.



get current reading from blue channel

It measures the current between evive's blue and black channel.



get timer value

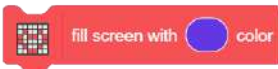
It returns the timer value since last reset.



reset timer

It resets the timer value.

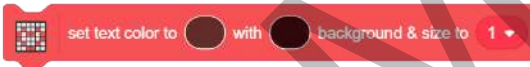
Display Palette for evive's TFT Screen



It changes the color of the screen to the one selected from the drop-down menu.



It sets the cursor at the specified coordinate. Origin (0,0) is at the top left corner of the screen. X-axis is towards the right while Y-axis is downward.



It sets the text color, background color, and size of the text.



It is used to write text or values on evive's TFT screen.



It draws a line of the selected color between the specified coordinates.

Actuators Palette for evive



Run motor at M1 or M2 channel.



Free/Lock (or stop) motor at M1 or M2 channel.



Set servo to a specified angle on S1 or S2 channel .

Dabble App Palette for evive



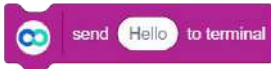
It enables LED control or pin state of any analog or digital pin.



It compares the string/data received from the Terminal module with the specified string/data.



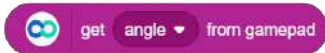
It returns the number received from the Terminal module.



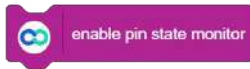
It sends data to the Terminal module.



It checks whether the specified button has been pressed in the Gamepad module or not.



It returns the specified data in the Joystick mode of the Gamepad module.



It enables the Pin State Monitor module.



It checks if the specified tactile switch is pressed in the Inputs module.



It checks whether the specified slide switch is in the specified state in the Inputs module.



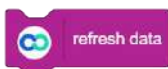
It returns the a value between 0 and 1023 of the specified potentiometer in the Inputs module.



It enables the Motor Control module.



It returns data of the specified phone sensor e.g. accelerometer, light sensor, and sound sensor.



It refreshes the internal library variables to the new data available from app.

Sensors Palette for evive



get ultrasonic sensor distance (cm) | trig 2 , echo 3

It returns the distance sensed by the ultrasonic sensor.



read analog sensor light / photoresistor at A2

It reads the selected analog sensor connected to the specified pin e.g. joystick and photo resistor.



read digital sensor IR(proximity) at 2

It reads the selected digital sensor connected to the specified pin e.g. proximity and PIR.



get temperature from DHT sensor at pin 2

It returns the temperature or humidity value from the DHT11 sensor.

There are many blocks and palettes compatible with evive, Arduino, and other boards in PictoBlox. Explore more at <https://thetempedia.com/docs/pictoblox>.

STEMpedia blends theory with experiential learning which helps them open their minds and develop curiosity, creativity, and problem-solving skills – the must-haves in 21st century. It is the key to transform the youth of today into innovators of tomorrow.

OTHER DIY KITS



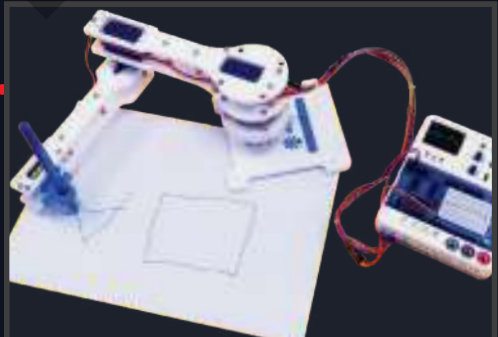
Bio-Medi Kit



Internet of Things Kit



Agri Tech Kit



Robotic Arm Kit

Agilo Research Pvt. Ltd.

Ahmedabad, Gujarat, India - 380 052

+91-8726 533 960 | +91-8765 696 060

www.thestempedia.com

