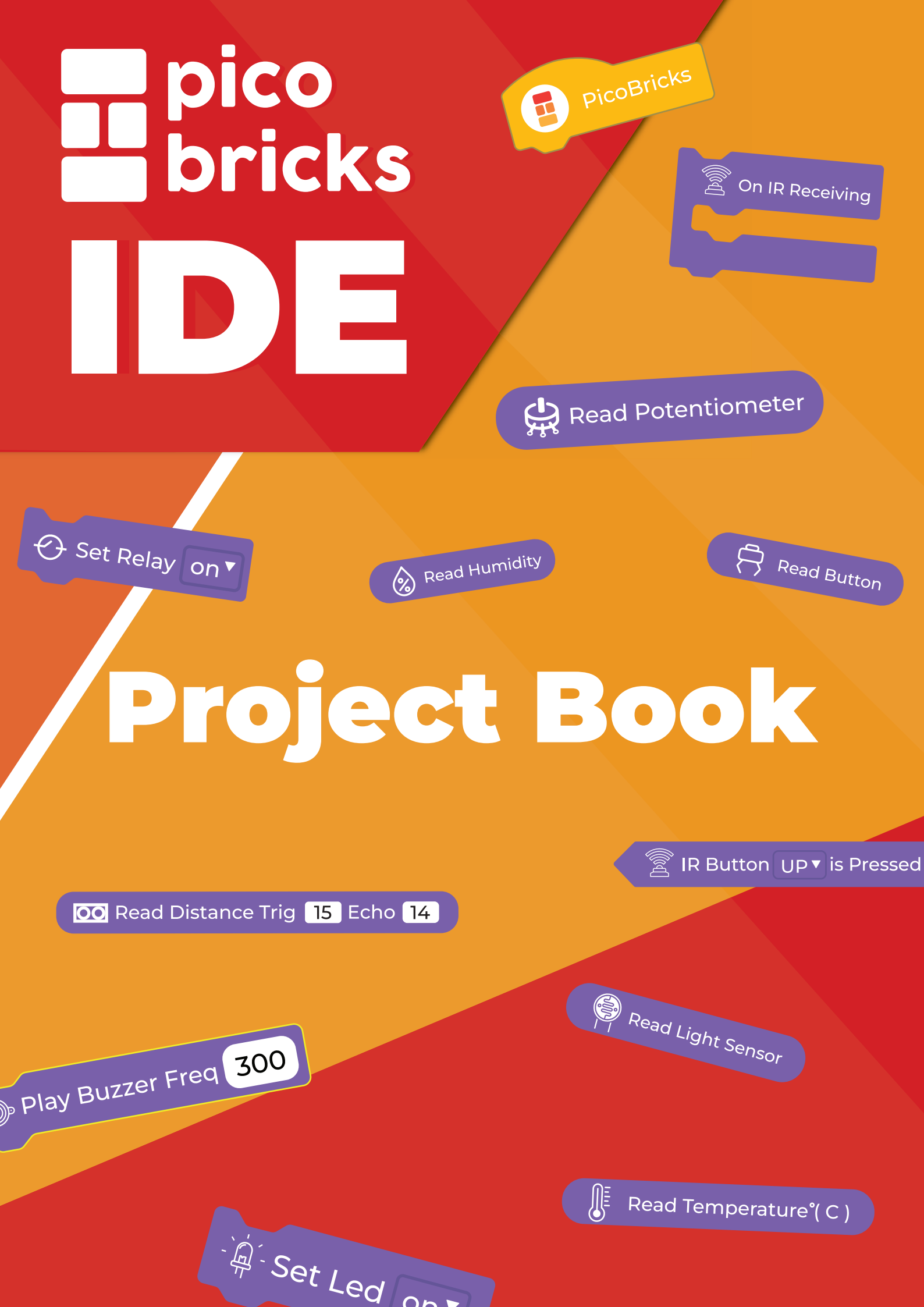
pico bricks IDE

Project Book







Copyright © 2022 Robotistan

Except for commercial usage, you can copy, reproduce and edit photos and content in this book by referring

Contents: Mustafa Kemal Avcı, Abdullah Kaya, Selim Gayretli

Translation: Naze Gizem Özer

Design: Ahmet Gürsu, Elanur Tokalak

Pico Bricks Developer Team

Yasir Çiçek - Project Manager

Yusuf Gündoğdu - Software Developer

Mehmet Suat Morkan - Chief Developer

Mehmet Ali Dağ - Hardware Developer



CONTENTS

Development Environments 9
PicoBricks IDE 9
Pico JR 10
PicoBlockly 11
Pico Py 11
PicoBricks Simulator 11

PROJECTS 17

Blink 18
Project Details and Algorithm 18
Wiring Diagram 18
Project Image 19
Extending This Project 19
Coding the Project with PicoBricks IDE 19

Action - Reaction 21
Project Details and Algorithm 21
Wiring Diagram 21
Project Image 22
Extending This Project 22
Coding the Project with PicoBricks IDE 22

Autonomous Lighting 24
Project Details and Algorithm 24
Wiring Diagram 24
Project Image 25
Extending This Project 25
Coding the Project with PicoBricks IDE 25

Thermometer 28
Project Details and Algorithm 28
Wiring Diagram 28
Construction Stages of the Project 29
Extending This Project 29
Coding the Project with PicoBricks IDE 29

Graphic Monitor 31
Project Details and Algorithm 31
Wiring Diagram 31
Project Image 32
Extending This Project 32
Coding the Project with PicoBricks IDE 32

Dominate the Rhythm 34
Project Details and Algorithm 34
Wiring Diagram 35
Project Image 35
Extending This Project 35
Coding the Project with PicoBricks IDE 35

Show Your Reaction 39
Project Details and Algorithm 39
Wiring Diagram 39
Project Image 40
Extending This Project 40
Coding the Project with PicoBricks IDE 40

My Timer 43
Project Details and Algorithm 43
Wiring Diagram 43
Construction Stages of the Project 44
Extending This Project 44
Coding the Project with PicoBricks IDE 44

Alarm Clock 48
Project Details and Algorithm 48
Wiring Diagram 48
Project Image 49
Extending This Project 49
Coding the Project with PicoBricks IDE 49

Know Your Color 51
Project Details and Algorithm 51
Wiring Diagram 51
Project Image 52
Extending This Project 52
Coding the Project with PicoBricks IDE 52

Magic Lamp 55
Project Details and Algorithm 55
Wiring Diagram 55
Construction Stages of the Project 56
Extending This Project 56
Coding the Project with PicoBricks IDE 56

Smart Cooler 58
Project Details and Algorithm 58
Wiring Diagram 58
Project Image 59
Coding the Project with PicoBricks IDE 59

Buzz Wire Game 61
Project Details and Algorithm 61
Wiring Diagram 62
Construction Stages of the Project 62
Extending This Project 62
Coding the Project with PicoBricks IDE 62

Dinosaur Game 64
Project Details and Algorithm 64
Wiring Diagram 65
Project Image 65

Construction Stages of the Project 66
Extending This Project 66
Coding the Project with PicoBricks IDE 66

Night and Day 67
Project Details and Algorithm 67
Wiring Diagram 68
Project Image 68
Extending This Project 69
Coding the Project with PicoBricks IDE 69

Voice Controlled Robot Car 71
Project Details and Algorithm 71
Construction Stages of the Project 71
Extending This Project 72
Coding the Project with PicoBricks IDE 72

Two Axis Robot Arm 74
Project Details and Algorithm 74
Wiring Diagram 75
Construction Stages of the Project 75
Extending This Project 75
Coding the Project with PicoBricks IDE 80

Smart House 82
Project Details and Algorithm 82
Wiring Diagram 82
Construction Stages of the Project 83
Extending This Project 83
Coding the Project with PicoBricks IDE 85

Hungry Piggy Bank 86
Project Details and Algorithm 86
Wiring Diagram 86
Construction Stages of the Project 87
Extending This Project 87
Coding the Project with PicoBricks IDE 90

Automatic Trash Bin 91
Project Details and Algorithm 91
Wiring Diagram 91
Coding the Project with PicoBricks 92

Digital Ruler 93
Project Details and Algorithm 93
Wiring Diagram 93
Extending This Project 93
Coding the Project with PicoBricks IDE 94

Air Piano **95**
Project Details and Algorithm 95
Wiring Diagram 95
Extending This Project 95
Coding the Project with PicoBricks IDE 96

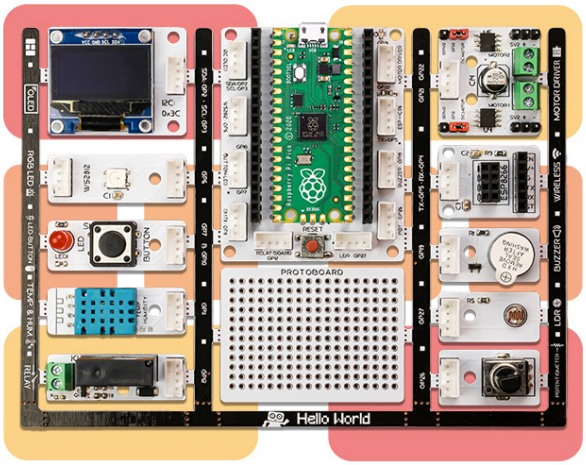
Maze Navigating Robot **97**
Project Details and Algorithm 97
Wiring Diagram 98
Extending This Project 98
Coding the Project with PicoBricks IDE 98

Smart Greenhouse **100**
Project Details and Algorithm 100
Wiring Diagram 101
Extending This Project 101
Coding the Project with PicoBricks IDE 101

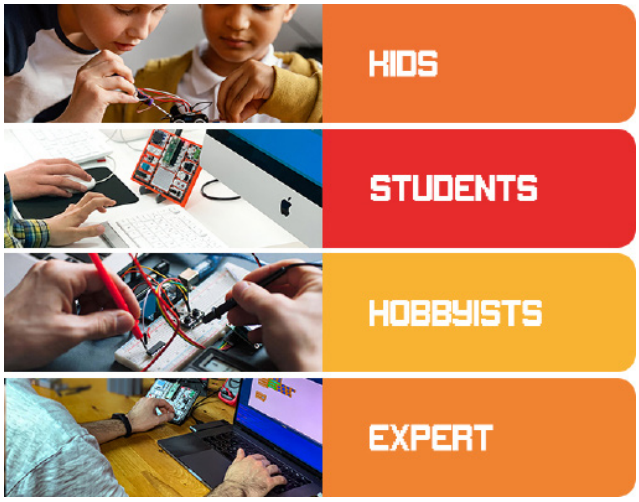
3 Resources
Robot Car Setup Guide 106
Smart Green House Setup Guide 112
3D Models 115

What Is Pico Bricks?

Pico Bricks is an electronic development board + software which is designed for use in maker projects. With ten detachable modules included, Pico Bricks can be used to create a wide variety of projects. It also includes a protoboard that you can use to add your own modules!



Pico Bricks is for everyone interested in electronics and coding. Beginners with no prior experience will find it easy to get started thanks to the modular hardware design, Scratch-like block coding environment, and simulator. Those with experience can dig more deeply into electronics or explore coding in Python. And even the most expert makers will appreciate how quickly they can explore ideas and create prototypes with Pico Bricks.



Unlike other boards, Pico Bricks has an incredible amount of flexibility for every level of makers! Bricks IDE has example codes for different scenarios.

Learn coding from zero to hero with MicroBlocks or the Pico Bricks's drag-n-drop, block coding builder. MicroBlocks is the easiest coding experience ever created and widely known in the maker industry.

Have a question? You can find more information here

POWERED BY
MicroBlocks

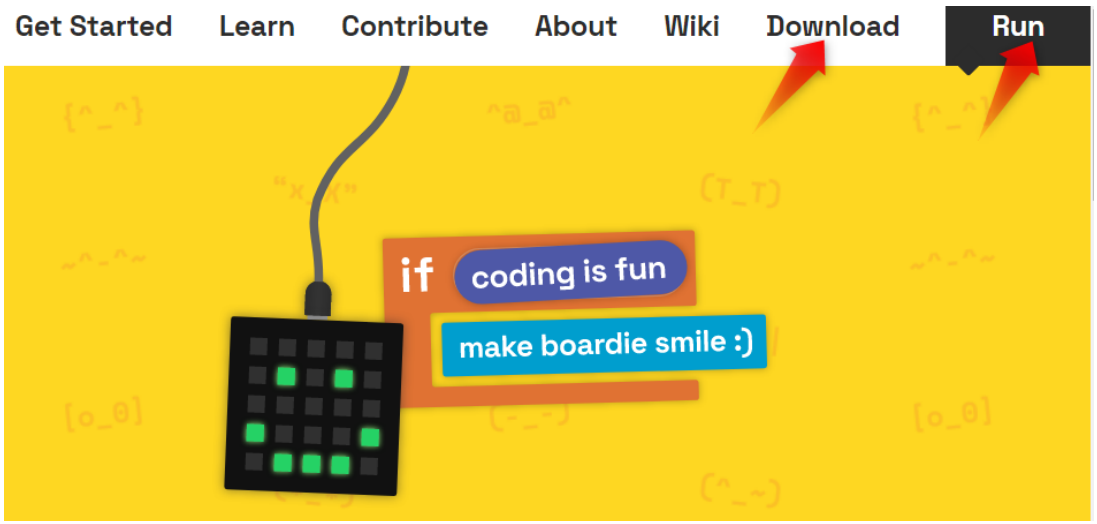
DEVELOPMENT ENVIRONMENT

1. Development Environments

In software, web and mobile application development, the development environment is a workspace with a set of processes and programming tools used to develop the source code for an application or software product. Development environments enable developers to create and innovate without breaking something in a live environment. You can code Picobricks using both text-based and block-based editors. MicroBlocks is a powerful editor with which you can code Picobricks with blocks.

Thonny editor is a free, dedicated IDE(integrated development environment) for Python designed for beginners and one of the best choices for those who are just starting to learn MicroPython. Whether you just code MicroPython or code an electronic circuit board, Thonny provides you with a lot of support.

Arduino IDE is an open-source software, designed by Arduino.cc and mainly used for writing, compiling & uploading code to almost all Arduino Modules. It is an official Arduino software, making code compilation too easy that even a common person with no prior technical knowledge can get their feet wet with the learning process. If you are proficient in Arduino C language or want to learn C language, Picobricks and Arduino IDE will be very good choices for you.



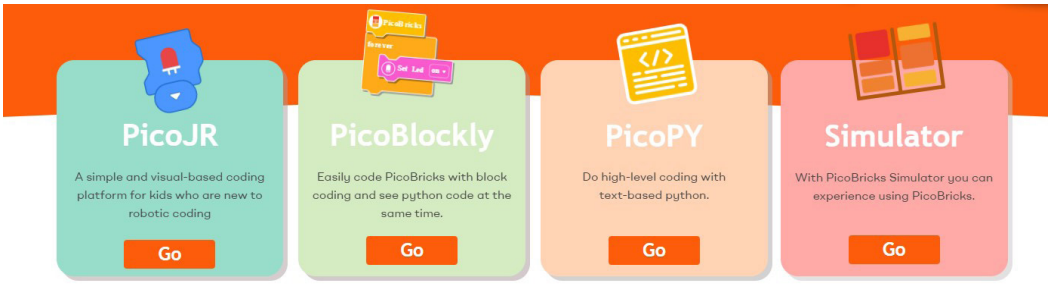
PicoBricks IDE

PicoBricks IDE is a programming editor that you can create some projects by writing code with blocks before switching to the MicroPython programming language. You can get the MicroPython equivalent of every code you create by dragging blocks in PicoBricks IDE. In addition, you can make some projects in Python language by entering the Python editor from the PicoBricks IDE without being dependent on offline editors. PicoBricks IDE is a platform that you can write code by using both block-based and text-based programming in the online environment. PicoBricks IDE is a real-time online programming editor. You can run the code blocks you have created for your project by pressing the “Run” button and you can stop it with the “Stop” button.

You can run PicoBricks with an external power supply by uploading the project code you prepared with PicoBricks IDE into Raspberry Pi Pico. PicoBricks IDE also has a horizontal block structure for younger users. Also, with PicoBricks IDE, even if you do not have any PicoBricks, you can create some projects by using PicoBricks Simulator.

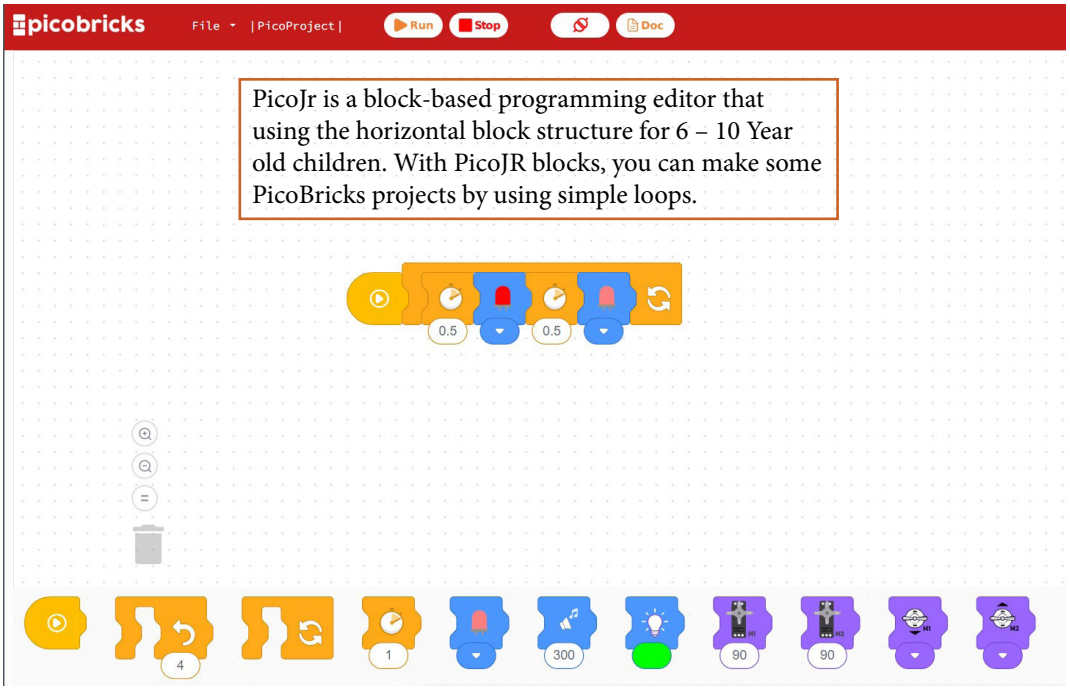
To code PicoBricks with PicoBricks IDE, you can scan this QR code or open http://rbt.ist/ide in the browser.

On this page, there are PicoJR, PicoBlockly, PicoPY and PicoBricks Simulator Editors that you can code in the PicoBricks IDE.

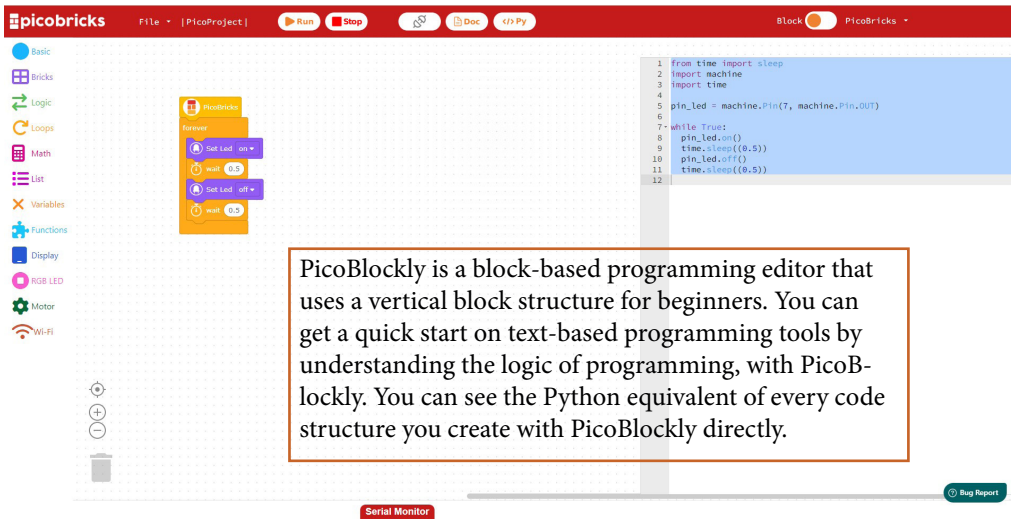


Let's get to know these editors!

PicoJR

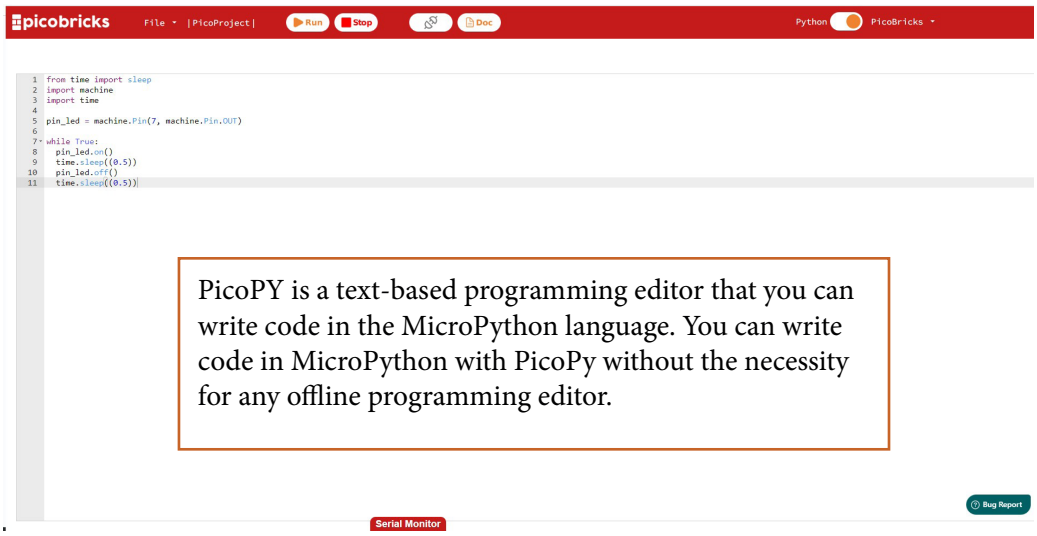


PicoBlockly



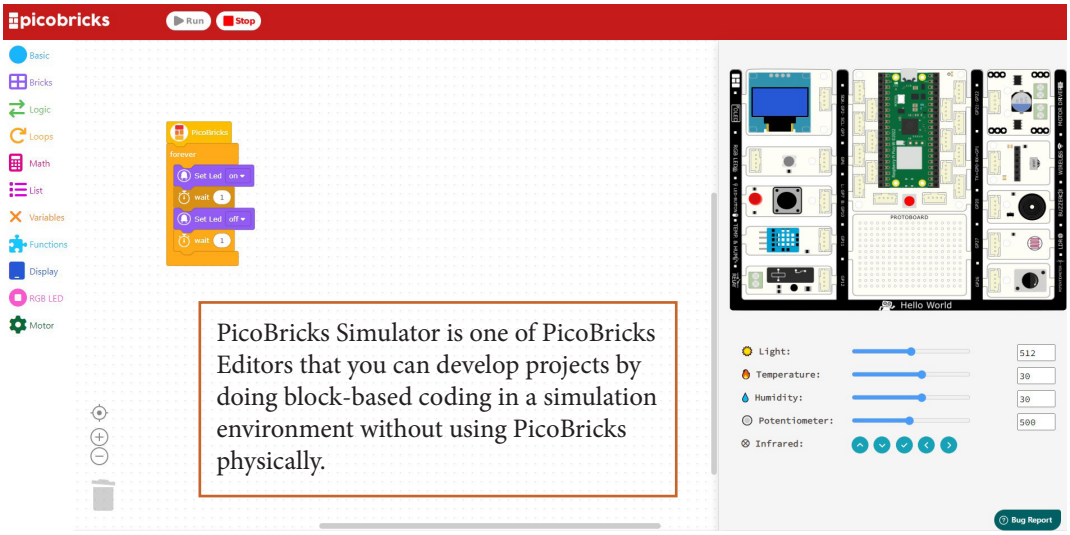
PicoBlockly is a block-based programming editor that uses a vertical block structure for beginners. You can get a quick start on text-based programming tools by understanding the logic of programming, with PicoBlockly. You can see the Python equivalent of every code structure you create with PicoBlockly directly.

PicoPY



PicoPY is a text-based programming editor that you can write code in the MicroPython language. You can write code in MicroPython with PicoPy without the necessity for any offline programming editor.

PicoBricks Simulator

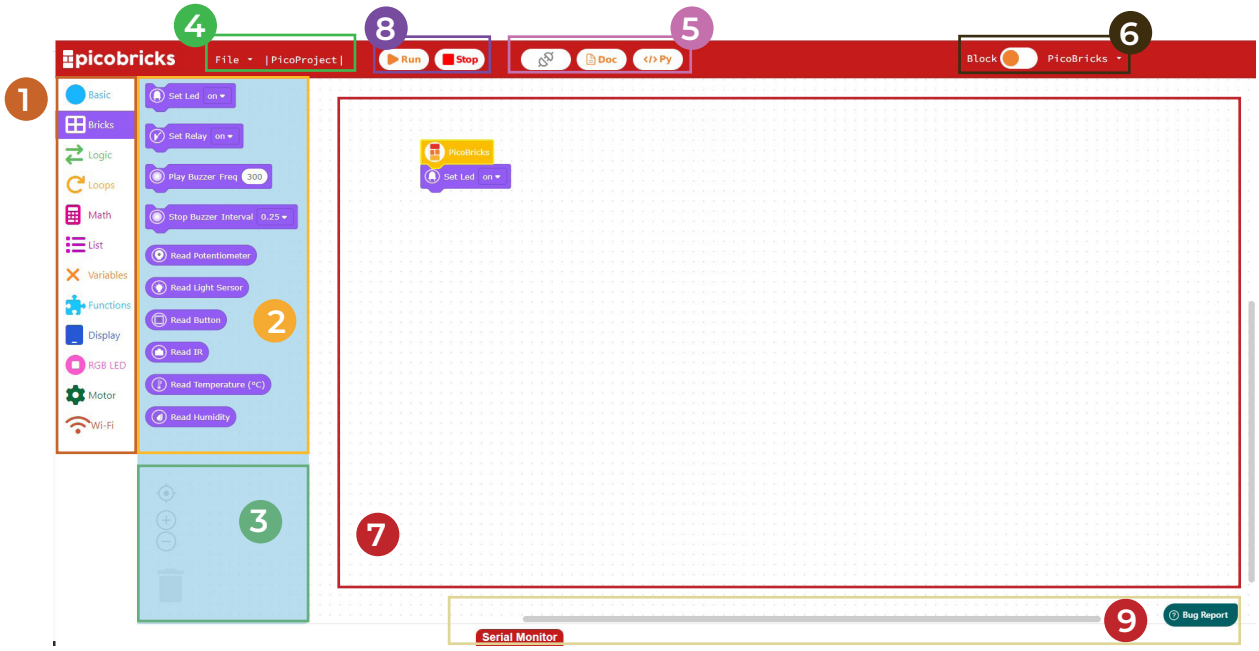


PicoBricks Simulator is one of PicoBricks Editors that you can develop projects by doing block-based coding in a simulation environment without using PicoBricks physically.

Let's Get To Know PicoBlockly

PicoBlockly, is a PicoBricks Programming Editor that has vertical block structure. The examples from PicoBricks Project Book were prepared by using PicoBlockly Editor.

Introduction To The Interface



1. Block Categories: It is the area that the blocks used to code PicoBricks are listed. In this area, there are sensor, condition and loop blocks etc. required to code PicoBricks sensors and you can access the blocks of programming structures by clicking on the relevant structure. All of the categories are separated from each other with different colors and icons. The blocks in the categories are listed in the number 2 block palette area.

2. Block Palette: It is the area where the blocks are listed according to the selected block category. According to the selected category, PicoBricks sensor blocks, if-else structure, conditional structures, loop structures, etc. programming functions have block equivalents.

3. Tool Panel: It is the area where the tools for zooming, centering and deleting the code blocks in the coding area.

4. File Operations Panel: By using this area, you can open a new project page and the projects you have made with PicoBricks IDE previously and save the project code that you created by using blocks to the computer or Raspberry Pi Pico W.

5. Connection and Documentation Panel: By using this area, you can make the connection between PicoBricks IDE and PicoBricks, access the MicroPython access the MicroPython output of the developed code blocks and the documents of PicoBricks projects.

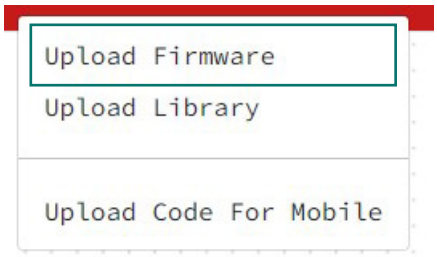
6. Software Installation and Editor Change Area : In this area, you can upload the firmware files of the Raspberry Pi Pico microcontrollers used by PicoBricks and the PicoBricks library prepared with the MicroPython programming language. You can quickly switch between PicoBlockly and PicoPy editors.

7. Coding Area: This is the area where you will prepare the code blocks of your project by dragging and dropping. You can drag the algorithm that is necessary for your project by preparing with PicoBricks IDE Blocks.

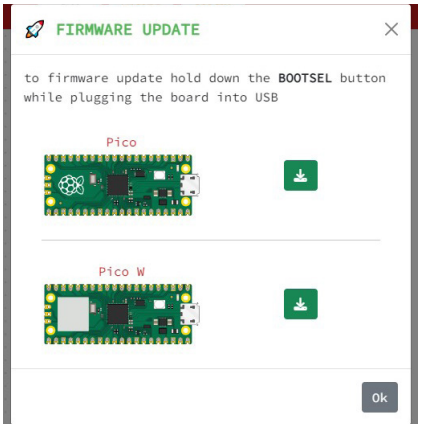
8. Start & Stop Area: You can start or stop the code blocks that you prepared by using these buttons.

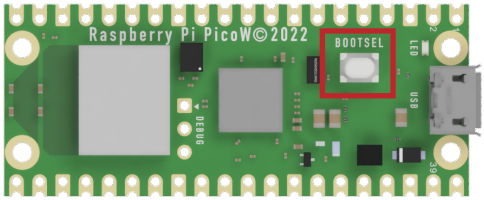
9. Serial Port: In this area, you can see the values you want to print to the serial port.

PicoBricks IDE - PicoBricks Connection and Running



After opening PicoBricks IDE, click the PicoBricks menu and then, click **“Upload Firmware”** tab from PicoBricks Menu. In the window that opens, you should put PicoBricks into Bootsel mode.

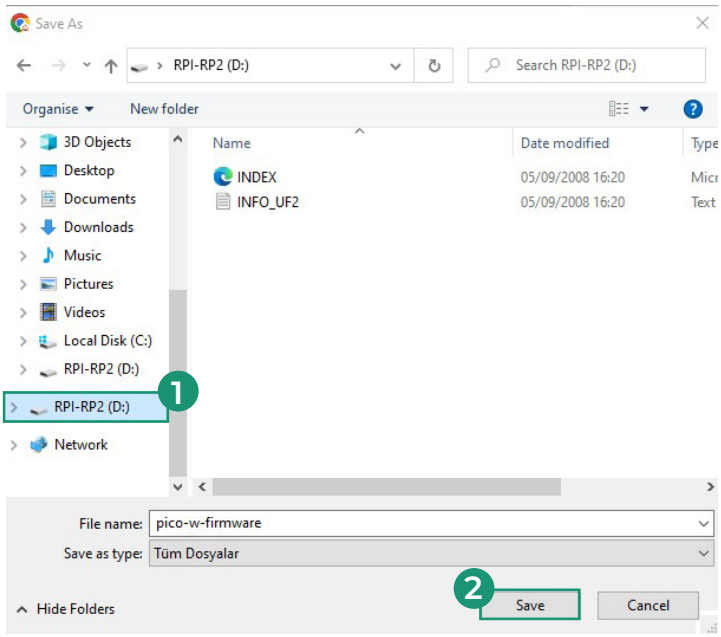


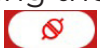


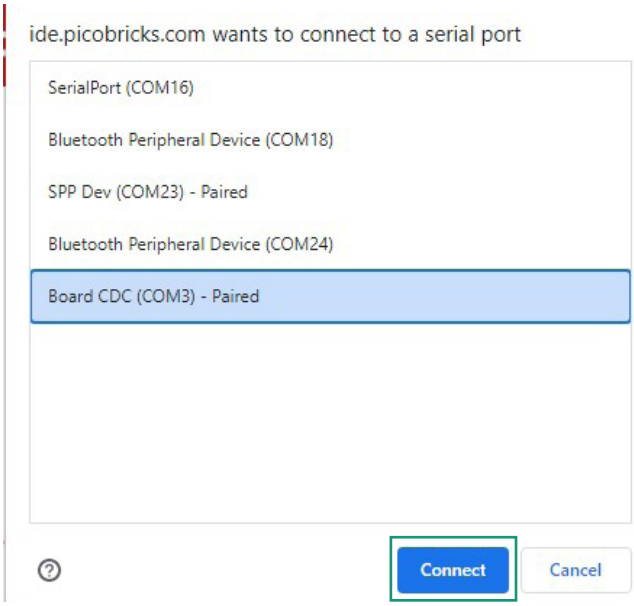
To connect PicoBricks to PicoBricks IDE, you should connect the board to your computer with a USB cable while holding the BOOTSEL button on Raspberry Pi Pico. (If you will not switch to a different IDE, you should do this just once.)

You can close the first window that opens after doing Bootsel PicoBricks. Select the Raspberry Pi Pico module installed on PicoBricks and upload the correct Firmvare file.

Upload the firmware file into “RPI-RP(D:)” driver by following the steps below.

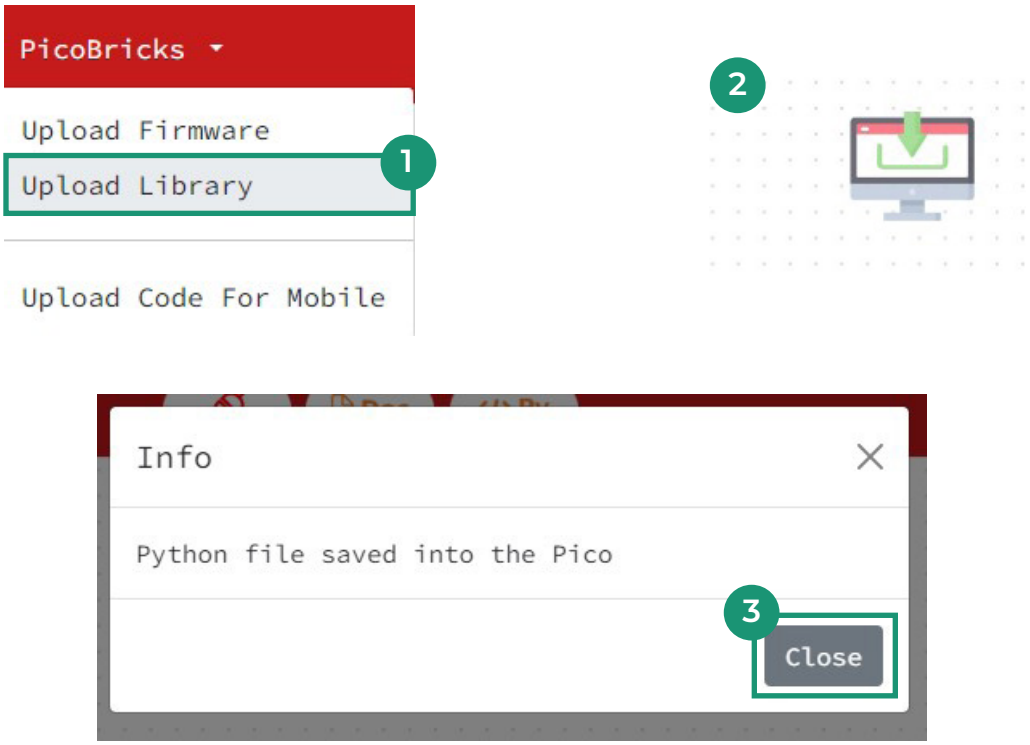


After saving the Firmware file, click the  button to connect PicoBricks. You can make the PicoBricks connection by selecting the right port from the window that opens. After making the connection, the  icon will change.



After you make the connection, let’s upload PicoBricks library to Raspberry Pi Pico microcontroller board. (If you have uploaded the library before and you have not used a different IDE, you should do this step only once.)

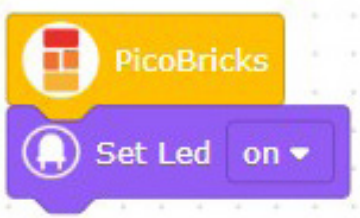
Let's upload the library to Raspberry Pi Pico by following the steps below.



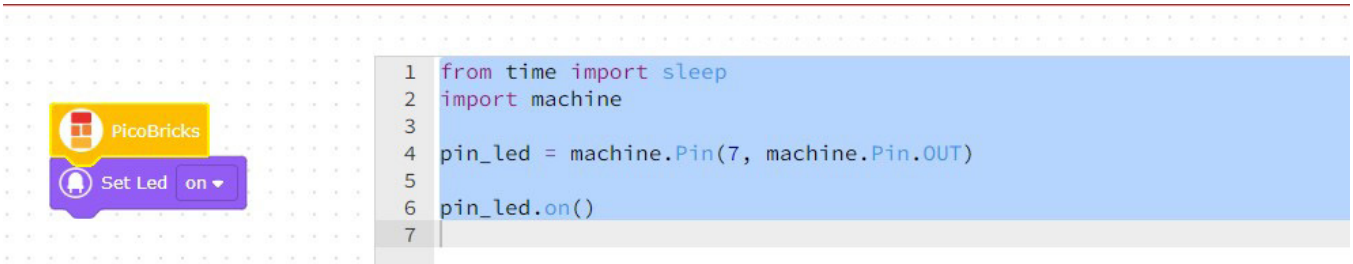
After uploading PicoBricks Library, let's test the steps we made by running red LED module.

Let's prepare the code blocks by following the steps below

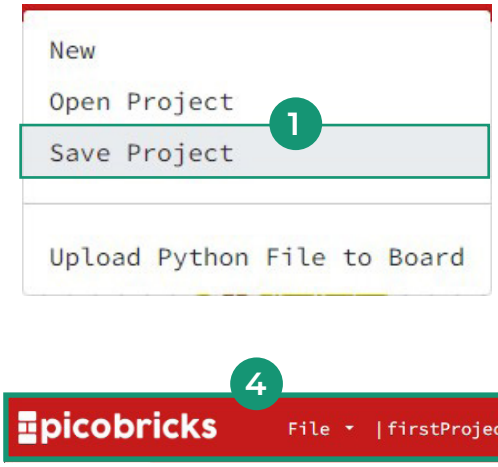
1. Let's drag "PicoBricks" and red LED blocks one under the other from "Basic" blocks.

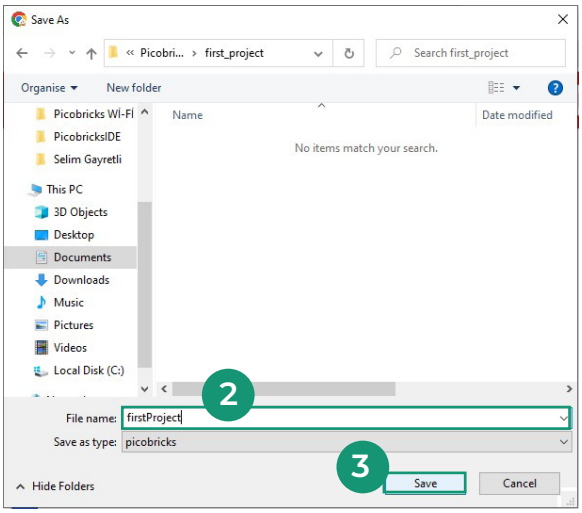


2. You can see the Python output of the code blocks by clicking the </> Py button.



3. After preparing the code, let's run the project by clicking the "Start" button.
4. You can save your project by following the steps below.





PROJECTS

Blink

In real life, the employee, who has just started to learn the job, first undertakes the most basic task.The cleaner first learns to use the broom, the cook learns to use the kitchen utensils, the waiter to carry a tray. We can increase these examples. The first code written by newcomers to software development is known as “Hello World”. Printing “Hello World” as soon as the program starts on the screen or console window in the language they use is the first step in programming. Like a baby starting to crawl... The first step to robotic c oding, also known as physical programming, is the Blink application. It means winking at robotic coding. By simply connecting an LED to the circuit board, the coding is made to keep the LED blinking continuously. Ask people who have developed themselves in the field of robotic coding how they got to this level. The answer they will give you starts like this; it all started with a flashing LED!

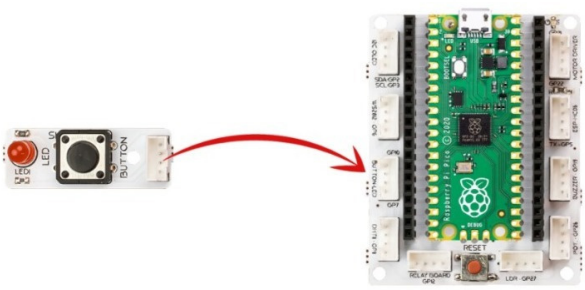
LEDs are the language of electronic devices. Thanks to the LEDs, the programmer tells the users at which stage of the task the device is, what the problem is, if any, and which options are active. In this project, you will learn the types of LEDs on it with Picobricks and learn how to flash them.

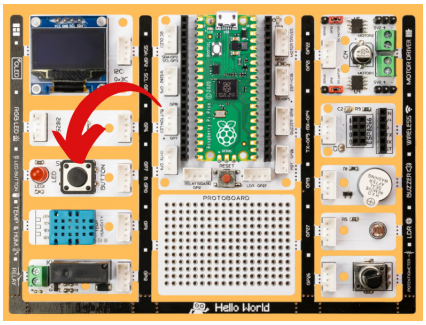
Project Details and Algorithm

There are 1 x 5mm red LED and 1 x WS2812B RGB LED on Picobricks. While normal LEDs can light up in one color, RGB colors can light up in different colors, both primary and secondary colors. In this project we will use the red LED on Picobricks.

In the project, we will write the necessary codes to turn on the red LED on Picobricks, turn it off after a certain time, turn it on again after a certain time, and repeat these processes continuously.

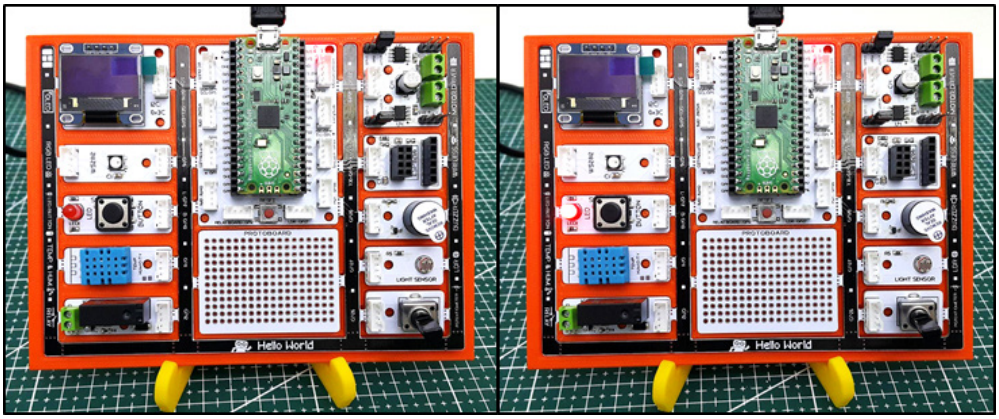
Wiring Diagram





You can code and run Picobricks’ modules without wiring. If you are going to use the modules by separating them from the board, you should make the module connections with grove cables.

Project Image

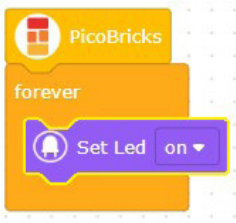


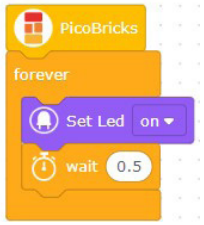
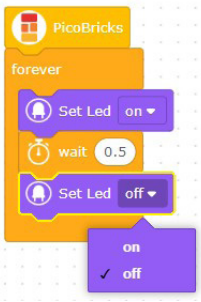
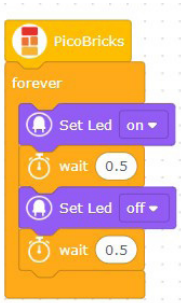
Project Proposal

Can we light the LED with different time intervals? For example; flashing of the LED several times per second, several times every half second.

Coding the Project with PicoBricks IDE

If you make the connection between PicoBricks IDE and PicoBricks, now you can start to prepare the code blocks.

1	First of all, drag the “PicoBricks” block from “Basic” blocks for the project to run when you click the “Start” button or run PicoBricks	
2	Drag the “Forever” block below the “PicoBricks” block to blink the red LED continuously.	
3	Drag the “Set LED on” block into the “Forever” block to light up red LED	

4	Drag the “wait” block under the “Set LED on” block that lights up the red LED so that the red LED will blink at certain intervals. Set the “wait” block to 0.5 for the waiting.	
5	To determine whether the “Set LED” block is on or off, you can set it by clicking the “ok” button on the right of the block.	
6	To wait for half a second after turning off the red LED, turn off the Red LED and drag the half second wait block. Code blocks of the project are ready!	

Action - Reaction

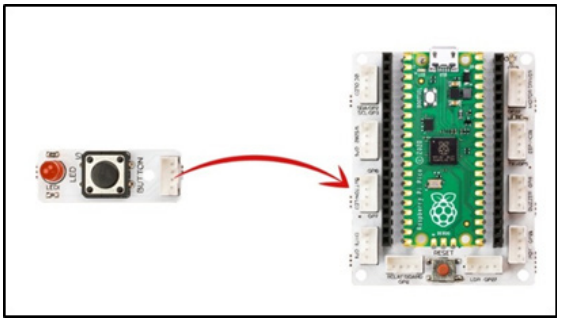
As Newton explained in his laws of motion, a reaction occurs against every action. Electronic systems receive commands from users and perform their tasks. Usually a keypad, touch screen or a button is used for this job. Electronic devices respond verbally, in writing or visually to inform the user that their task is over and what is going on during the task. In addition to informing the user of these reactions, it can help to understand where the fault may be in a possible malfunction. In this project, you will learn how to receive and react to a command from the user in your projects by coding the button-LED module of Picobricks..

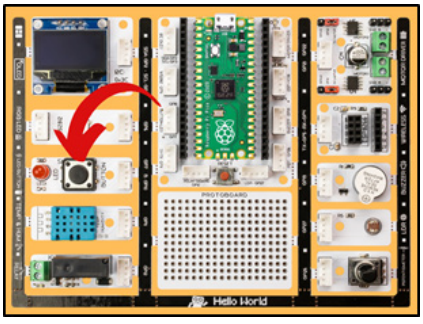
Project Details and Algorithm

Different types of buttons are used in electronic systems. Locked buttons, push buttons, switched buttons... There is 1 push button on Picobricks. They work like a switch, they conduct current when pressed and do not conduct current when released. In the project, we will understand the pressing status by checking whether the button conducts current or not. If it is pressed, it will light the LED, if it is not pressed, we will turn off the LED.

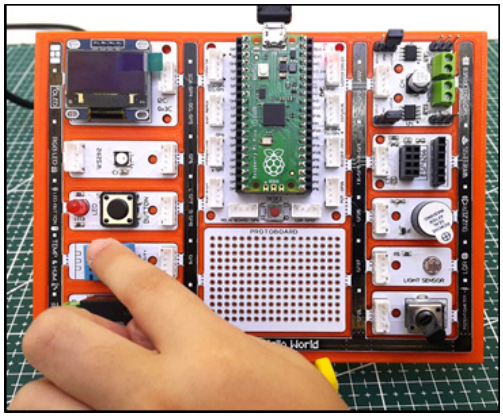
Wiring Diagram

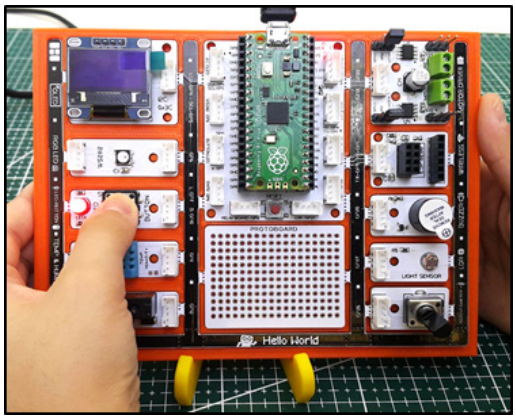
You can code and run Picobricks' modules without wiring. If you are going to use the modules by separating them from the board, you should make the module connections with grove cables.





2.2.3. Project Image



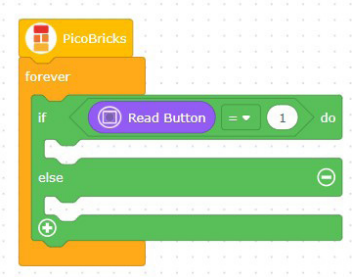
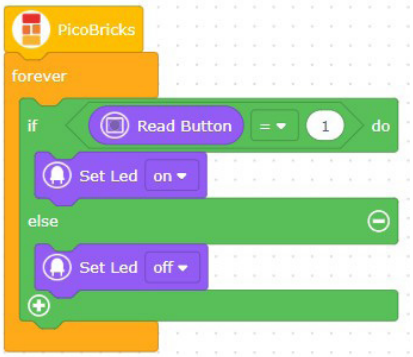


2.2.4. Project Proposal

In this project, the LED turns on when the button is pressed, and the LED turns off when the button is released. You can write the necessary codes for the LED to turn on when the button is pressed once and to turn the LED off when it is pressed again.

2.2.5. Coding the Project with PicoBricks IDE

1	First of all, drag the “PicoBricks” block from “Basic” blocks for the project to run when you click the “Start” button or run PicoBricks.	
2	Drag the “Forever” block to run the processes in the project continuously.	
3	To create the necessary condition structure for the project, drag “if-else” block from “Logic” blocks into “forever” block.	

4	Create the condition statement by dragging the  block from “Logic” blocks and the  block from “Bricks” blocks. Then, drag this structure  into the “if” block.	
5	Drag the “Set LED on” block into the if structure and the “Set LED Off” block into the else structure for the LED to light up when the condition is provided and run your code. The code blocks of the project are ready, you can save it.	

2.3. Autonomous Lighting

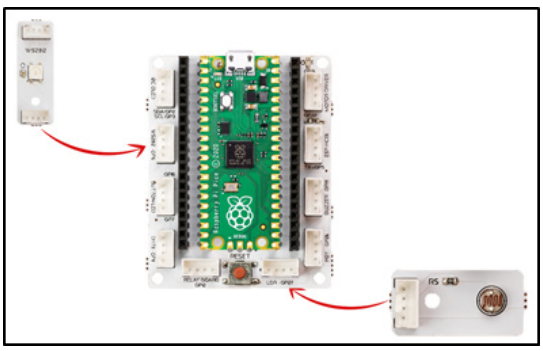
It is called the state of being autonomous when electronic systems make a decision based on the data they collect and perform the given task automatically. The components that enable electronic systems to collect data from their environment are called sensors. Many data such as the level of light in the environment, how many degrees the air temperature is, how many lt/min water flow rate, how loud the sound is, are collected by the sensors and transmitted to PicoBricks as electrical signals, that is data. There are many sensors in Picobricks. Knowing how to get data from sensors and how to interpret and use that data will improve project ideas like reading a book improves vocabulary. In this project, with PicoBricks, we will enable the LED to turn on when the amount of light decreases in order to understand the working systems of the systems where the lighting is turned on automatically when it gets dark.

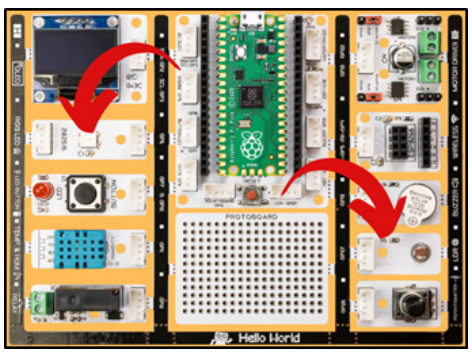
2.3.1. Project Details and Algorithm

Sensors are electronic components that detect data in external environments and send data to microcontrollers. The LDR sensor also detects the amount of light in the environment and sends analog values. In our project, we will first check the incoming data when the environment is light and dark by reading the LDR sensor values, then we will set a limit according to these data, and if the amount of light is below this limit, we will turn off the RGB LED of Picobricks, if not, we will turn off the LED.

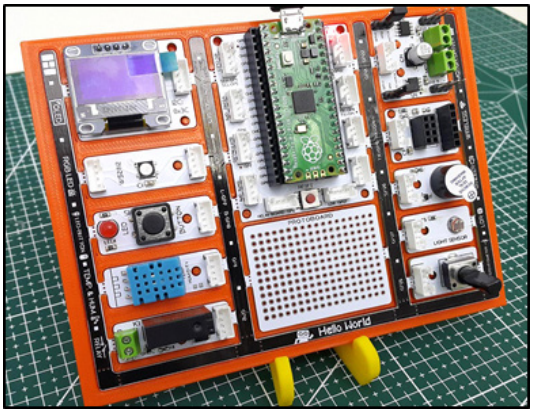
2.3.2. Wiring Diagram

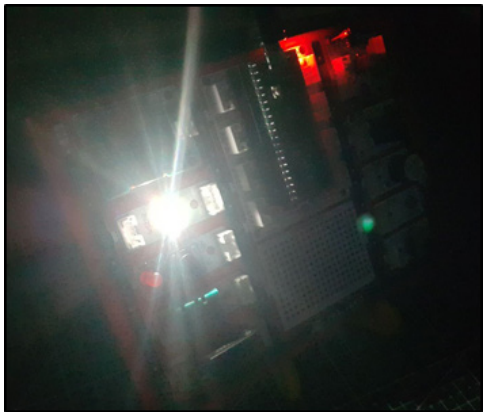
You can code and run Picobricks' modules without wiring. If you are going to use the modules by separating them from the board, you should make the module connections with grove cables.





2.3.3. Project Image

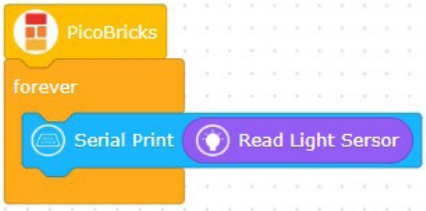


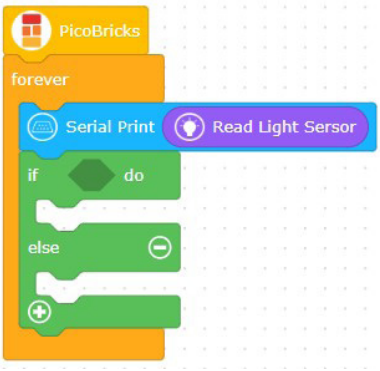
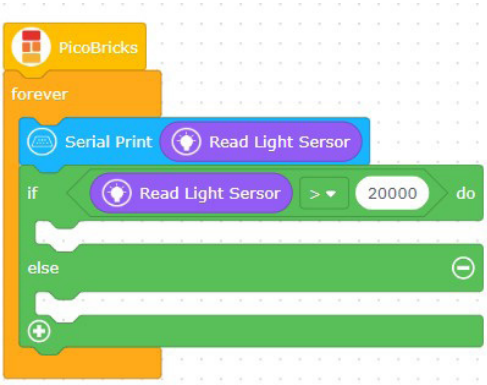
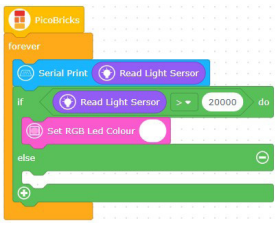


2.3.4. Project Proposal

In this project, we turned on the LED on the LDR sensor data on Picobricks if the environment was dark, and turned off the LED if it was bright. By processing the LDR sensor data, you can code a nightlight or table lamp in your home to turn on automatically in the dark. You can use the relay on Picobricks for this.

2.3.5. Coding the Project with PicoBricks IDE

1	First of all, drag the “PicoBricks” block from “Basic” blocks for the project to run when you click the “Start” button or run PicoBricks.	
2	Drag “Forever” block to run the project until you stop it. The other code blocks are dragged into this block.	
3	By printing the value of the LDR sensor on the Serial Monitor, you can easily detect the value range of the darkness. For this, create the “Serial Print” block from the “Basic” menu and the “Read Light Sensor” block from the “Bricks” menu and drag it into the “Forever” block.	

4	To determine the necessary condition statement for the project, drag the “if-else” block into the “Forever” block.	
5	After creating the condition statement, let’s drag it into the “if” block. To create the condition, use the light sensor value that is detected in the third step. Select the “>” operator from the options by taking the  block from “Logic” blocks. Drag  block from the ”Bricks” blocks on the left side of the equation. Let’s write the value of LDR sensor from the third step(20000) on the right side of the equation.  The blocks inside this condition statement will run when darkness is detected.	
6	For LDR sensor to lighten the environment when darkness is detected, set the Color of RGB LED Color Block to white.	

7	To turn off RGB LED when the light level is less than 20000(in a bright environment), drag the “Clear RGB LED Color” block.	

2.4. Thermometer

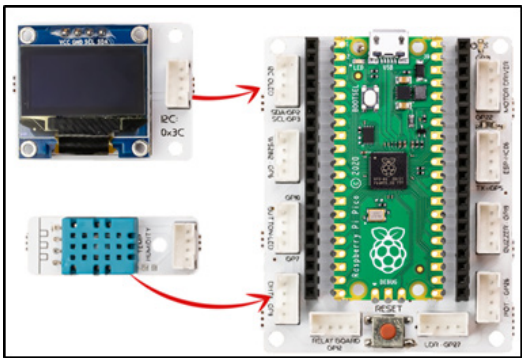
Sensors are the sense organs of electronic systems. We use our skin to feel, our eyes to see, our ears to hear, our tongue to taste, and our nose to smell. There are already many sense organs (sensors) in the picobrix. Also, new ones can be added. You can interact with the environment using humidity, temperature, light and many more sensors. Picobricks can measure the ambient temperature without the need for any other environmental component.

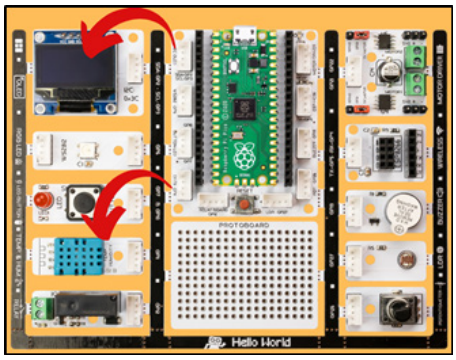
Ambient temperature is used in greenhouses, incubators, in environments used for the transport of drugs, briefly in situations where the temperature change must be constantly monitored. If you are going to do an operation on temperature change in your projects, you should know how to measure the ambient temperature. In this project, you will prepare a thermometer with Picobricks that will display the ambient temperature on the OLED screen.

2.4.1. Project Details and Algorithm

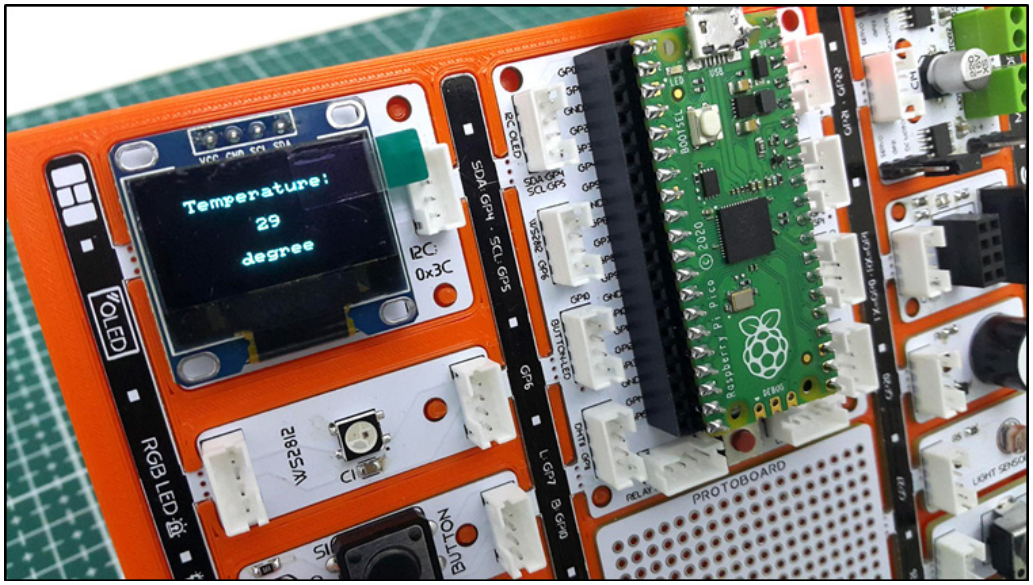
Picobricks has a DHT11 module. This module can sense the temperature and humidity in the environment and send data to the microcontroller. In this project, we will write the necessary codes to print the temperature values measured by the DHT11 temperature and humidity sensor on the OLED screen.

2.4.2. Wiring Diagram





2.4.3. Construction Stages of the Project

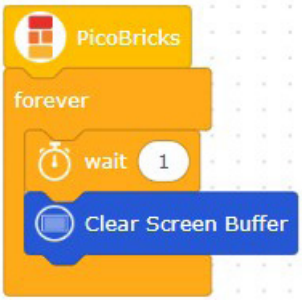
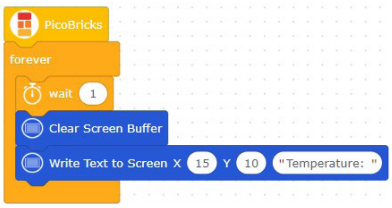
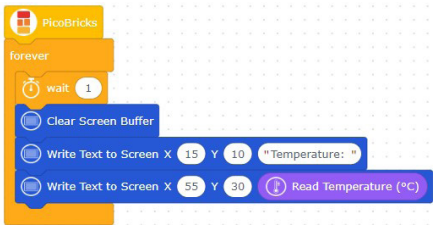
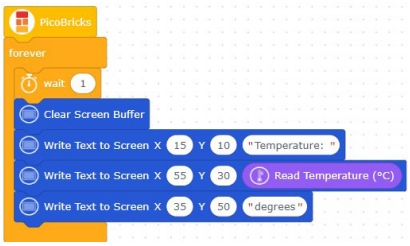
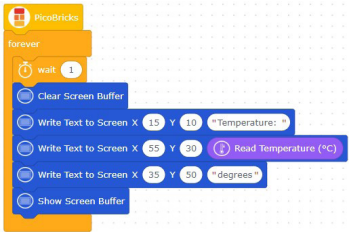


2.4.4. Project Proposal

In order to develop your project, you can make the red LED light up and a warning phrase appear on the screen when the temperature in the environment rises above 30 degrees.

2.4.5. Coding the Project with with PicoBricks IDE

1	First of all, drag the “PicoBricks” block from “Basic” blocks for the project to run when you click the “Start” button or run PicoBricks.	
2	Drag “Forever” block to run the project until you stop it. The other code blocks are dragged into this block.	

3	For the loop to repeat at intervals of 1 second, drag “Wait” block to the first row inside the “forever” block.	
4	Drag “Clear Screen Buffer” block from the “Display” blocks into the “forever” block to clear the screen each time that the loop repeats.	
5	To write “Temperature” text on the X=15 and Y=10 coordinates of OLED screen on PicoBricks, drag the “Write Text to Screen” block from the “Display” blocks and write “Temperature:” in the text writing area.	
6	To print the value that you get from PicoBricks DHT11 sensor on the X=55 and Y=30 coordinates of OLED screen, drag the “Read Temperature” block from the “Bricks” blocks into the “Write Text to Screen” block.	
7	To write “degrees” text on the X=35 and Y=50 coordinates, write “degrees” to the writing area by dragging the “Write Text to Screen” block from the “Display” blocks.	
8	To see the texts you want to print on the PicoBricks OLED screen module, you should initialize the OLED screen. For this, drag the “Show Screen Buffer” block from the “Display” blocks.	

2.5. Graphic Monitor

When we look at the electronic items around us, you realize that they have many replaceable features and they are designed by engineers to be most useful to the user. Such as lighting systems, cooking systems, sound systems, cleaning systems. The way it works, the amount, the method, etc., by many system users. features can be programmed to change.

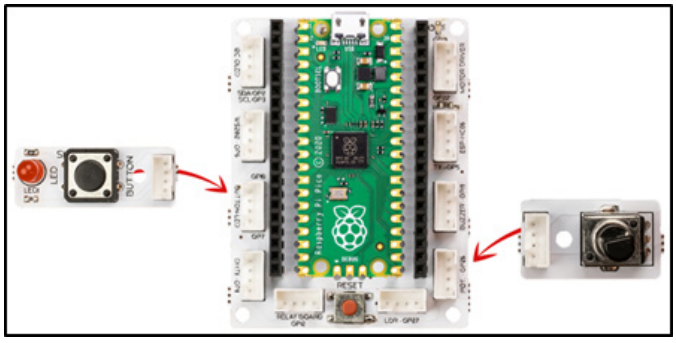
In robotic projects, in the processes of changing the sound level, changing the motor speed, changing the brightness of the light, the electrical voltage is sent in a way that creates a lower or higher effect. By decreasing the frequency of the electrical signal to the component, it can be operated at a lower level, and by increasing the frequency of the outgoing electrical signals, it can be operated at a higher level.

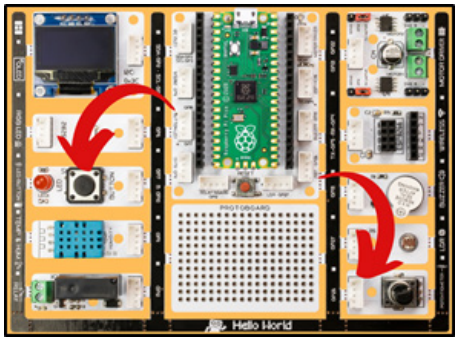
In systems without a screen, real-time graphic monitors are used to monitor some sensors and variables involved in the operation of the system. Graphic monitors make it very easy to detect the fault.

2.5.1. Project Details and Algorithm

In this project, we will prepare a project in which we increase or decrease the brightness of the red LED with a potentiometer. In addition, we will simultaneously monitor the electrical change occurring during this process on the Microblocks graphic monitor. When the picobricks starts, the potentiometer value will be read continuously and the brightness value of the LED will be adjusted. Applications in which the effect of the electrical signal is reduced by changing the frequency is called PWM. We will send the analog values we read from the potentiometer as PWM signals to the red LED and we will be able to adjust the illumination intensity.

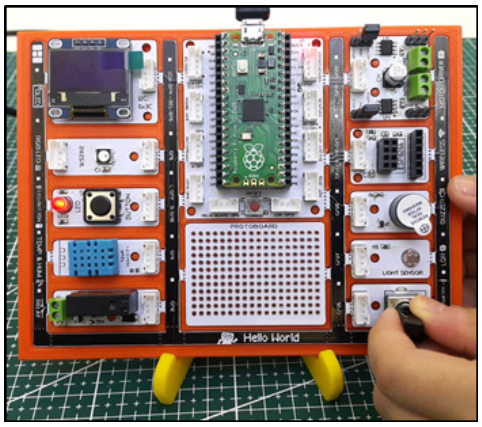
2.5.2. Wiring Diagram





2.5.3. Project Image



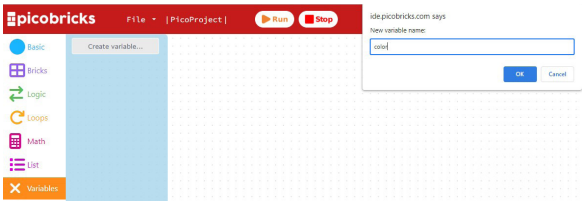
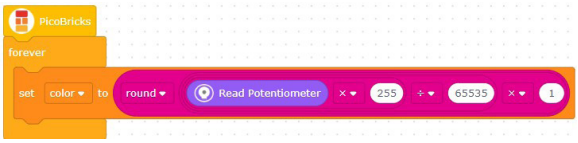
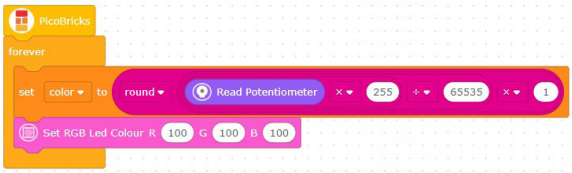
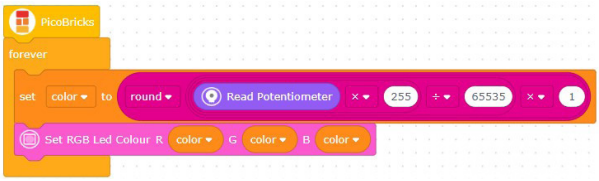


2.5.4. Project Proposal

As you turn the potentiometer, you can prepare a project that changes the volume of the sound coming out of the buzzer and displays the flowing data on the graphic monitor.

2.5.5. Coding the Project with PicoBricks IDE

1	First of all, drag the “PicoBricks” block from “Basic” blocks for the project to run when you click the “Start” button or run PicoBricks.	
2	Drag “Forever” block to run the project until you stop it. The other code blocks are dragged into this block.	

3	By following the steps below, let's create a variable named as "color". When you create the "color" variable, three blocks are created in the "Variables" blocks. 	
4	To set the value of the potentiometer between 1-255, create the mathematical expression below with the "operation" blocks from the "Math" blocks and assign it to the "color" variable as shown below. 	
5	To change the brightness of the RGB LED with the potentiometer, firstly let's drag the "Set RGB LED Color" block that you can change the RGB color values manually, in the RGB LED blocks.	
6	Drag the "color" variable you created into the Red, Green and Blue values to change the light value of the RGB LED according to the value of the potentiometer.	

2.6. Dominate the Rhythm

Many events in our lives have been digitized. One of them is sounds. The tone and intensity of the sound can be processed electrically. So we can extract notes electronically. The smallest unit of sounds that make up music is called a note. Each note has a frequency and intensity. With the codes we will write, we can adjust which note should be played and how long it should last by applying frequency and intensity.

In this project, we will prepare a music system that will play the melody of a song using the buzzer module and adjust the rhythm with the potentiometer module with Picobricks. You will also learn the use of variables, which has an important place in programming terminology, in this project.

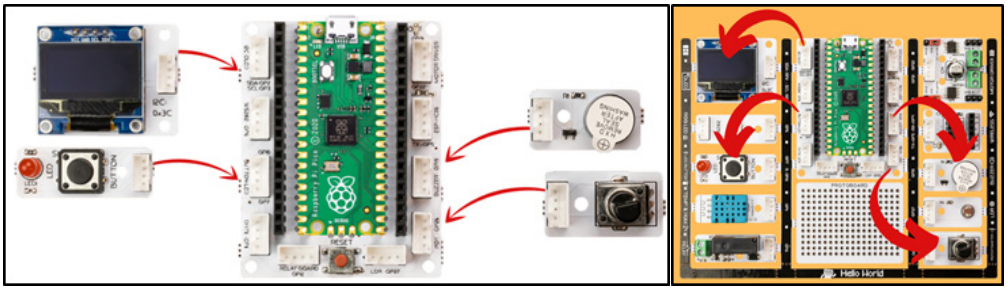
2.6.1. Project Details and Algorithm

With Picobricks you can play any song whose sheet we know. We will use the button-LED module to start the song, the potentiometer module to adjust the speed of the song, and the buzzer module to play the notes.

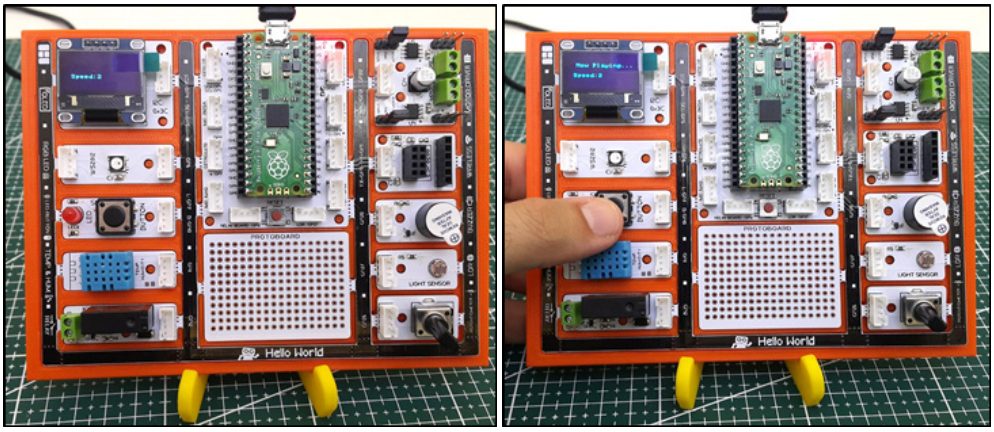
Potentiometer is analog input module. It is variable resistance. As the amount of current flowing through it is turned, it increases and decreases like opening and closing a faucet. We will adjust the speed of the song by controlling this amount of current with codes. Buzzers change the sound levels according to the intensity of the current passing over them, and the sound tones according to the voltage frequency. With Microblock's, we can easily code the notes we want from the buzzer module by adjusting their tones and durations.

We will check the button press status in the project. We will make the melody start playing when the button is pressed. During the playing of the melody, we will use a variable called rthm to increase or decrease the playing times of the notes at the same rate. After Picobricks starts, we will enable the user to adjust the rthm variable with the potentiometer, either while playing the melody or before playing it. As long as Picobricks is on, we will divide the potentiometer value (0-1023) by 128 and assign it to the rthm variable. Variables are data structures that we use when we want to use values that can be changed by the user or sensors in our codes. When the user presses the button to start the song, we will prepare the note codes that will allow the notes to play for the duration calculated according to the rthm variable.

2.6.2. Wiring Diagram



2.6.3. Project Image



2.6.4. Project Proposal

To make your project more visual, you can light a different color LED according to the played note, show the note names and playing speed on the OLED screen.

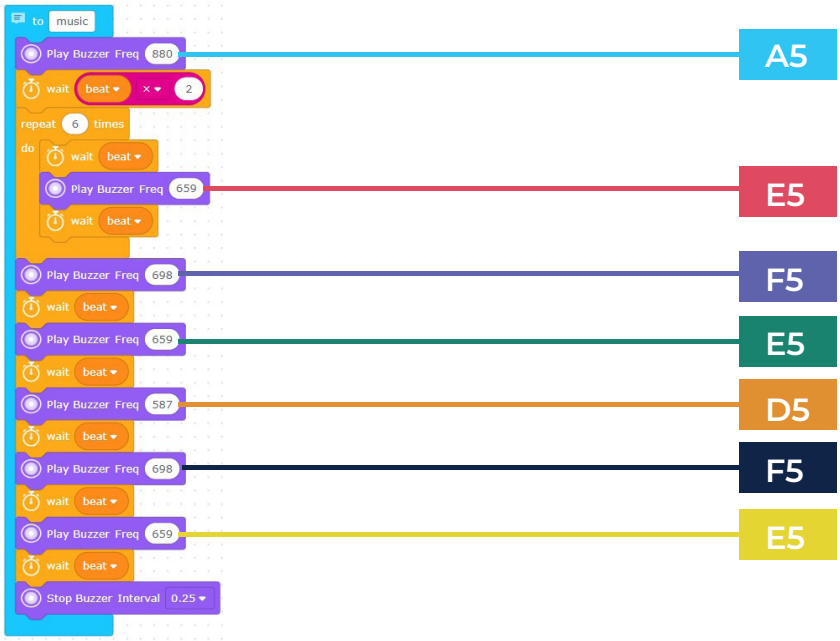
2.6.5. Coding the Project with PicoBricks IDE

First of all, let's start the project by creating the necessary functions for the code.,

- Let's define a function named as "music" that you will use with Dominate the Rhythm project.



- Let's create the melody that plays in the certain interval by using the buzzer blocks in the music function. To create the notes, you can change the frequency value of the "Play Buzzer Freq" block. The frequencies of the notes used in the Dominate The Rhythm project are showed in the music function below.



Now, let's continue the project by creating the other code blocks. What to do in this step:

- After clicking the Start button, drag PicoBricks block and forever block to create processes that run continuously.



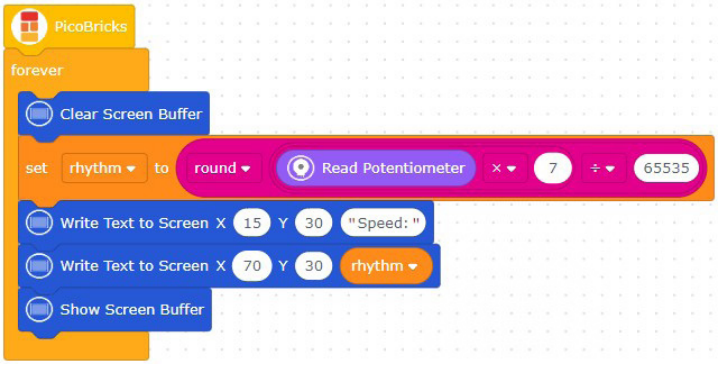
- Let's create a variable named as "rhythm" and replace the 0-65535 value from the "Read Potentiometer" block with the range of 0-7.



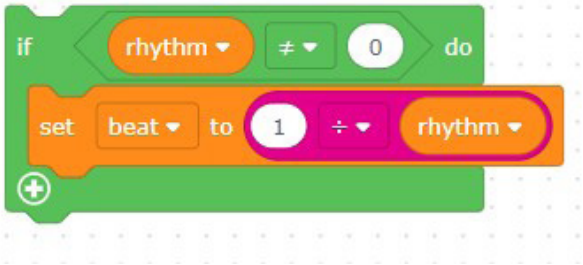
■ You can use the following operation to change the value of the potentiometer with the range of 0-7.



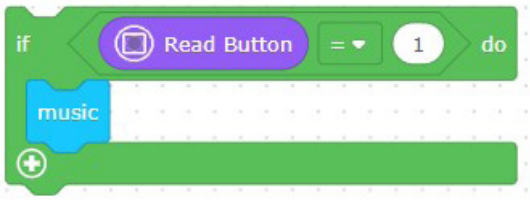
■ Let's print the "Rhtyhm" variable with the text "Speed:" to X:15 Y:30 and X:70 Y:30 coordinates on the OLED screen.



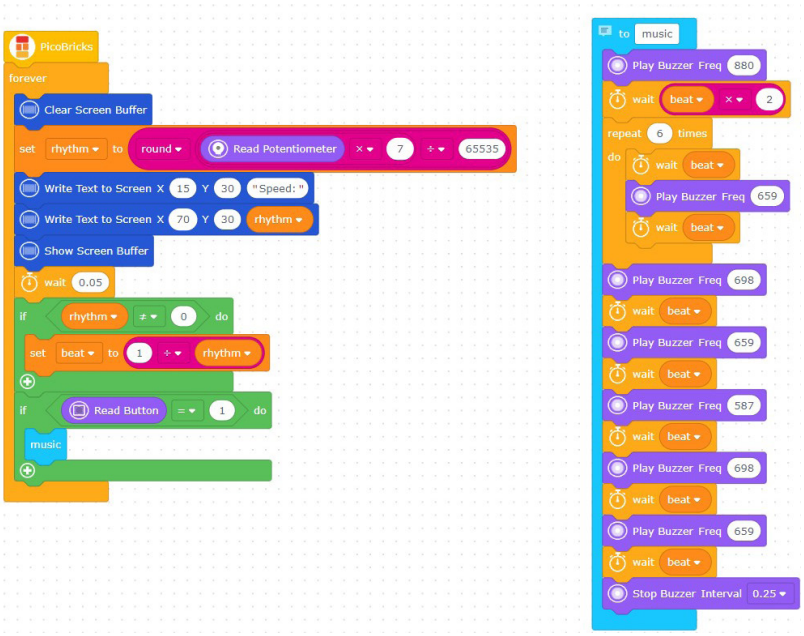
■ Now, let's creat the condition structures. If the value of the "rhythm" variable is not "0", let's create another variable named as "beat" and divide the rhythm value by 1. You can use the "beat" variable to determine the playing time of the tones in the "music" function.



■ Let's create the condition structure with the if block to call the "music" function when the button is pressed.



The code blocks are ready!



2.7 Show Your Reaction

Now we will prepare a game that develops attention and reflexes. Moving quickly and being able to provide attention for a long time are important developmental characteristics of children. Preschool and primary school children do activities that increase their attention span and reflexes, as they are liked by their parents and teachers. The electronic system we will prepare will be a game that increases attention and develops reflexes. After finishing the project, you can compete with your friends. :)

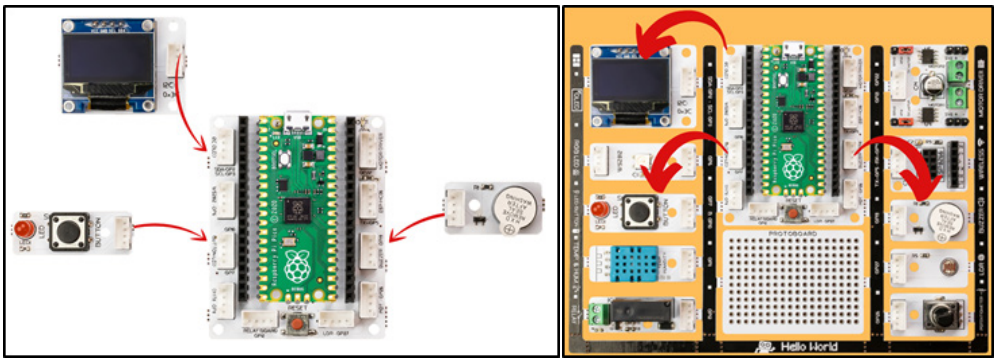
In this project you will learn about the randomness used in every programming language. With Picobricks, we will develop an electronic system using OLED display, Button-LED and Buzzer module.

2.7.1. Project Details and Algorithm

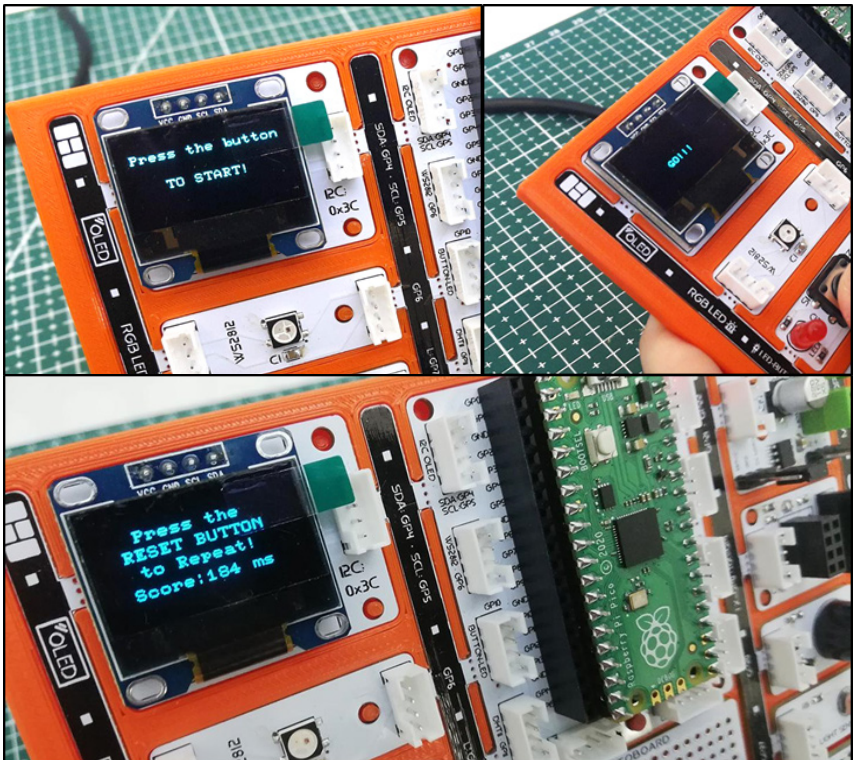
A timer starts running as soon as the Picobricks are turned on. With this timer, we can measure 1 thousandth of a second. One thousandth of a second is called a millisecond. Timers are used in many electronic systems in daily life. Timed lighting, ovens, irons, food processors...

When our project starts working, we will display a welcome message on the OLED screen. Then we will print on the screen what the user has to do to start the game. In order to start the game, we will ask the player to prepare by counting backwards from 3 on the screen after the button is pressed. After the end of the countdown, the red LED will turn on in a random time between 2-10 seconds. We will reset the timer immediately after the red LED lights up. We will measure the timer as soon as the button is pressed again. This value we get will be in milliseconds. We will display this value on the screen as the player's reaction time.

2.7.2. Wiring Diagram



2.7.3. Project Image



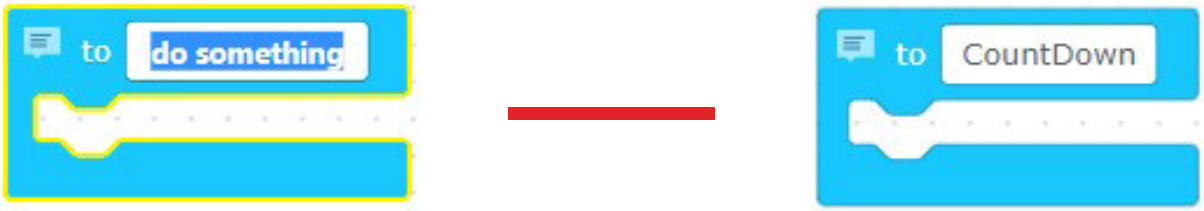
2.7.4. Project Proposal

Picobricks needs to be reset to be able to restart the game. You can develop your project by asking the button to be pressed again to start the game again. You can also have the highest score and the last scorer printed on the screen at the end of the game.

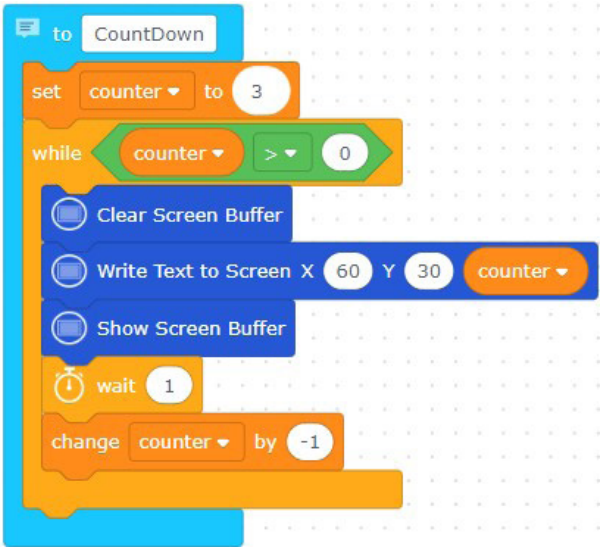
2.7.5. Coding the Project with PicoBricks IDE

First of all, let's start the project by creating the necessary functions for the code.

Let's creat the "CountDown" function that provides to count backwards from 3 on the OLED screen.



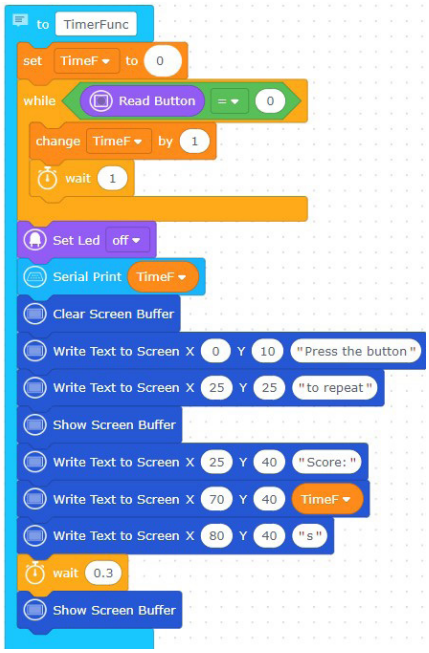
■ Let's define the function by writing "Countdown" into the block from the function blocks.



When we call the function, let's determine it by dragging it into the function block for it to print X=60 Y=30 coordinates 3-2-1 on the OLED screen.

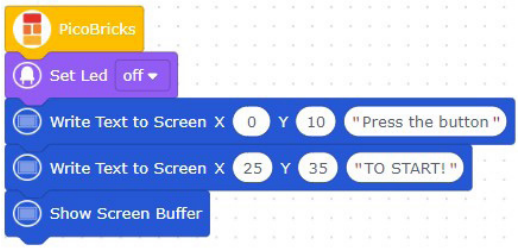
■ **The first function is ready!**

■ **Let's create the second function, "TimerFunc" function.**



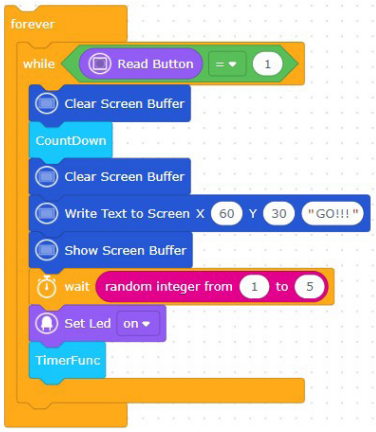
■ Let's drag the code blocks that provide to print "TimeF" variable on the OLED screen when the button is pressed, otherwise to increase the value of "TimeF" variable one by one, into the function block. When you call the function, a timer will run and it will stop when the button is pressed and print the score on the screen.

- Let's continue by creating the other code blocks and start the project by dragging the "PicoBricks" block. Each time the project starts, create the following code blocks to turn off the LED and write "Press the button" and "TO START!" on the screen.

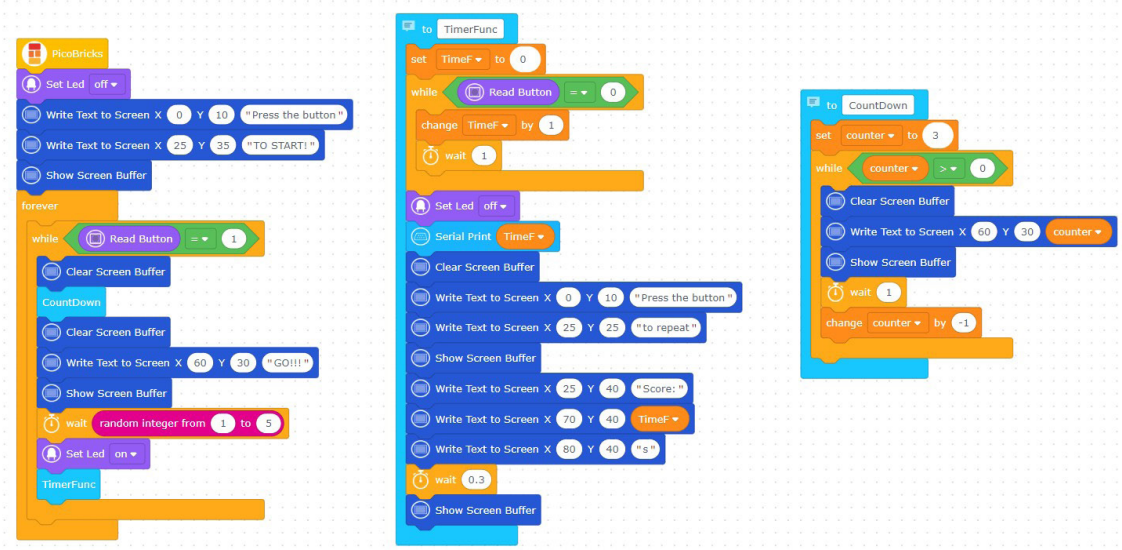


Let's create

- The code blocks that, clear the OLED screen after the button is pressed,
- The code blocks that, start the countdown by calling "CountDown" function,
- The code blocks that, after the countdown, turn on the red LED in a random second between by printing "GO!" on the screen and
- The code blocks that, print the "Score" by calling "TimerFunc" function.



- **The code blocks are ready!**



2.8. My Timer

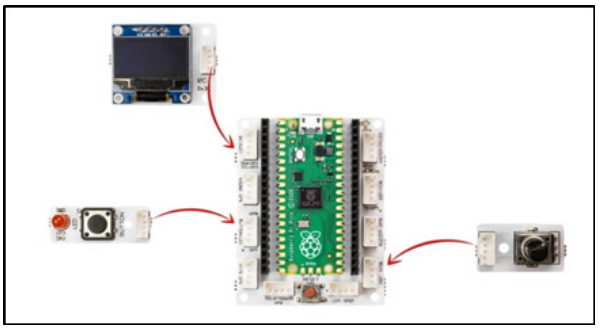
Measuring time is a simple but important task that we do in our daily lives without realizing it. A surgeon in surgery, a business person trying to catch up with a meeting, an athlete trying to win, a student trying to finish an exam or a chess match... Smart wrist watches, phones and even professional chronometers are used to measure time. Time is a variable that should be used very accurately in electronic systems. For example, a washing machine; how long the drum will rotate clockwise, how much counterclockwise, how many seconds water must flow in order to dissolve the detergent are tasks done by measuring time. To develop projects where time is of the essence, you need to know how to use it.

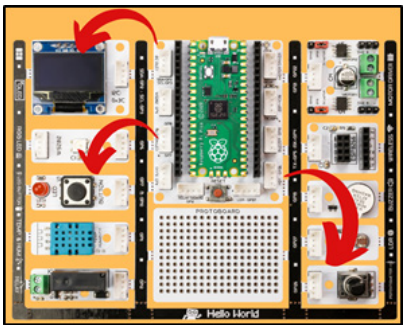
In this project, you will make your own time measuring device using Picobircks, OLED display, button and potentiometer modules. A Timer...

2.8.1. Project Details and Algorithm

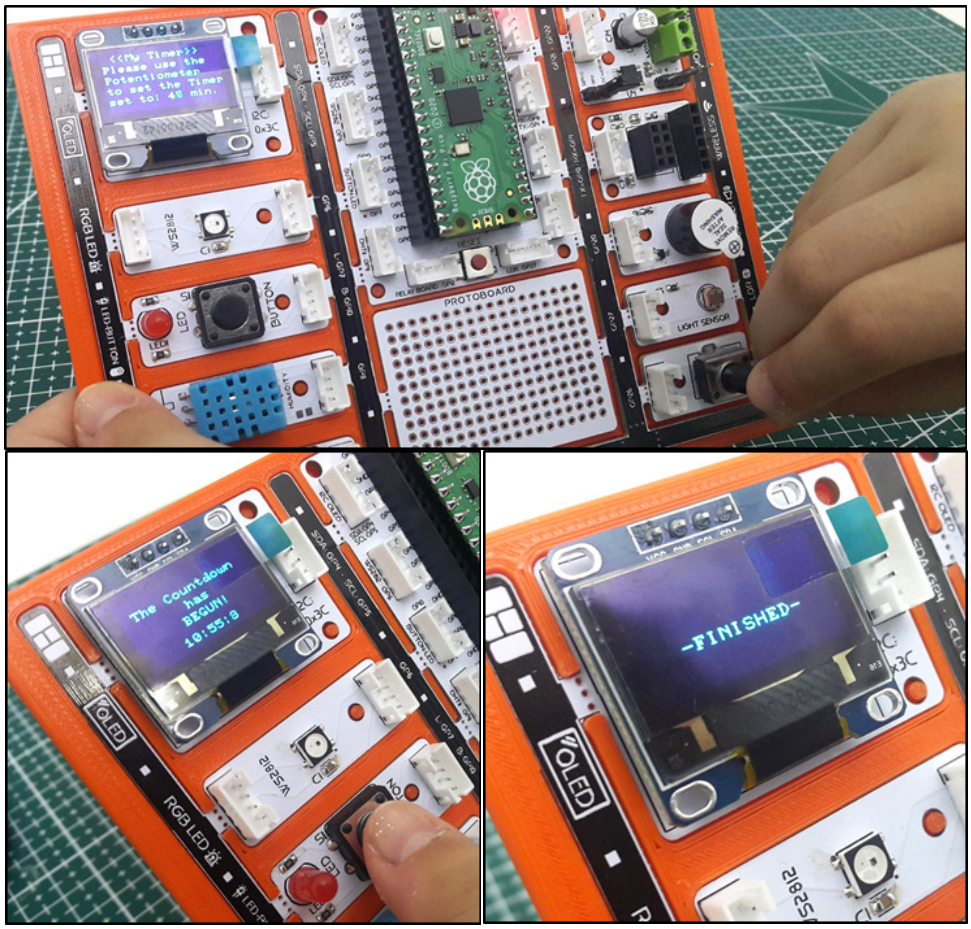
When Picobricks starts, let's put a statement on the screen that introduces the project and contains instructions. As the user turns the potentiometer, it will set a time in the range of 0-60 minutes. When the user presses the button of Picobricks after deciding the time with the potentiometer, it will start counting down in minutes and seconds on the screen. If the button is pressed while the time is running backwards, the Timer will stop and show the remaining time on the screen. If the minute, second and second value reaches zero without pressing the button, a notification stating that the time has expired will be displayed on the screen and the program will be stopped.

2.8.2. Wiring Diagram





2.8.3. Construction Stages of the Project



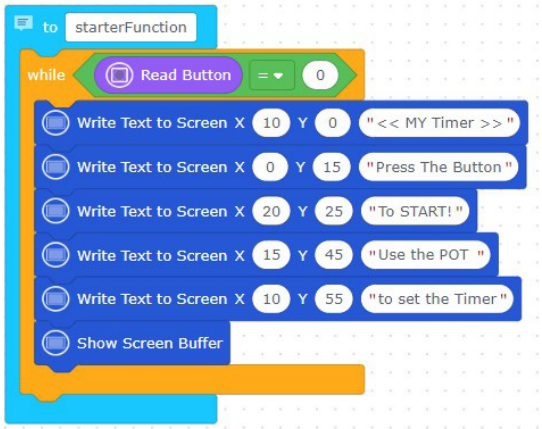
2.8.4. Project Proposal

You can add a beep to the start of the Timer. When the time is reset, you can give different and high tone warnings with the buzzer and announce that the time is up from afar.

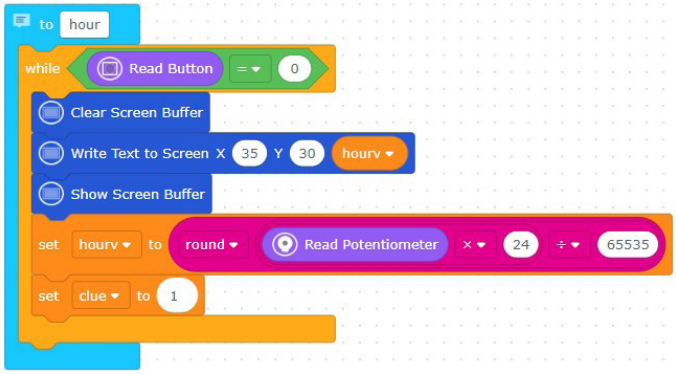
2.8.5. Coding the Project with PicoBricks IDE

First of all, let's start the project by creating the necessary functions for the code.

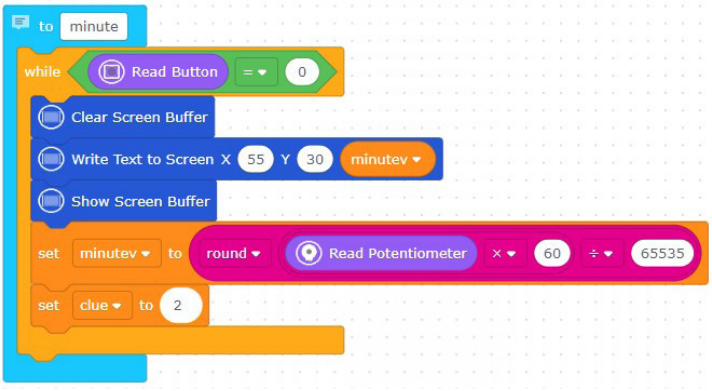
■ Define a function named as “starterFunction” to determine the statements to be written on the screen when you start the project.



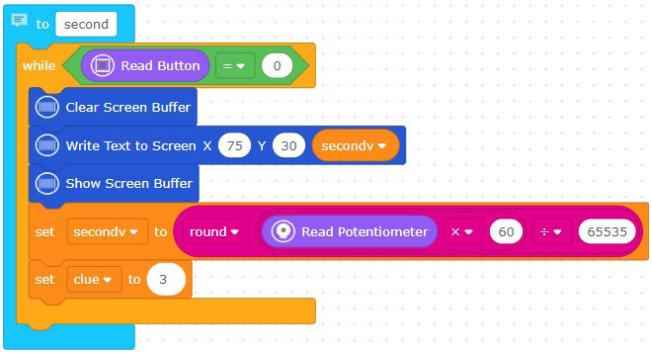
■ Let’s define a function named as “hour” to set the value of hour with the potentiometer.



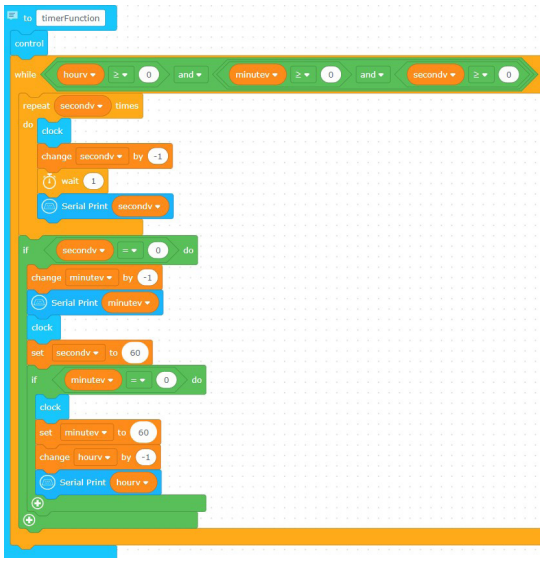
■ Let’s define a function named as “minute” to set the value of minute with the potentiometer.



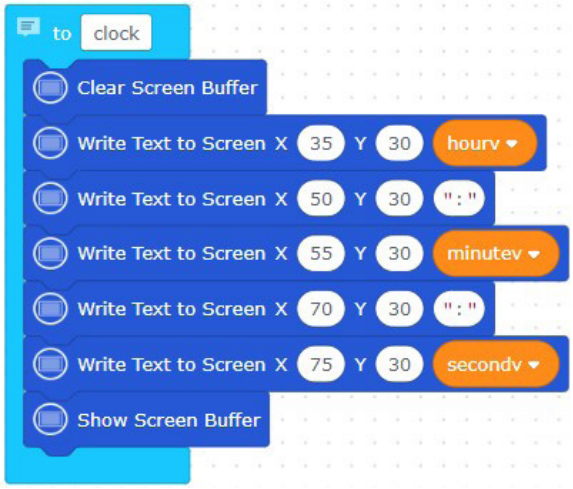
- Let's define a function named as "second" to set the value of second with the potentiometer.



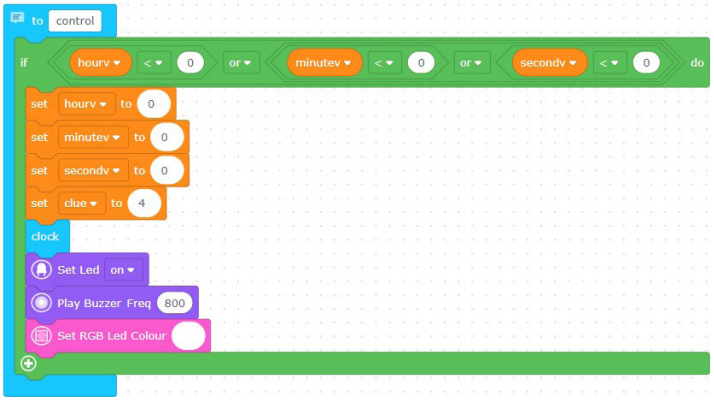
- Let's define a function named as "timerFunction" that start the countdown until all the hour, minute and second values adjusted with the potentiometer are "0" and print these on the OLED screen.



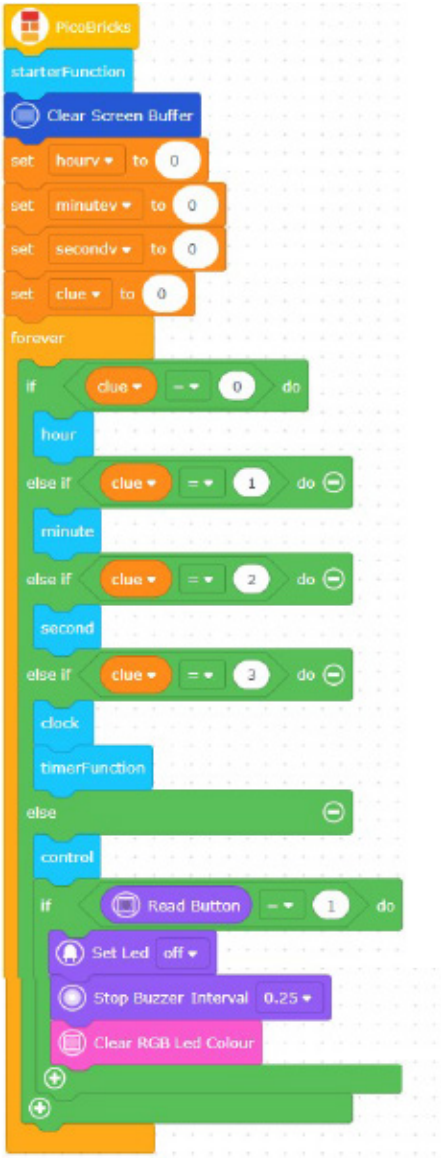
- Let's define a function named as "clock" to print hour, minute and second values on the OLED screen. Define the control structure by creating a function named as "control".



- Define the control structure by creating a function named as “control”.



- **Let’s continue creating the code by dragging “PicoBricks” block.**



2.9. Alarm Clock

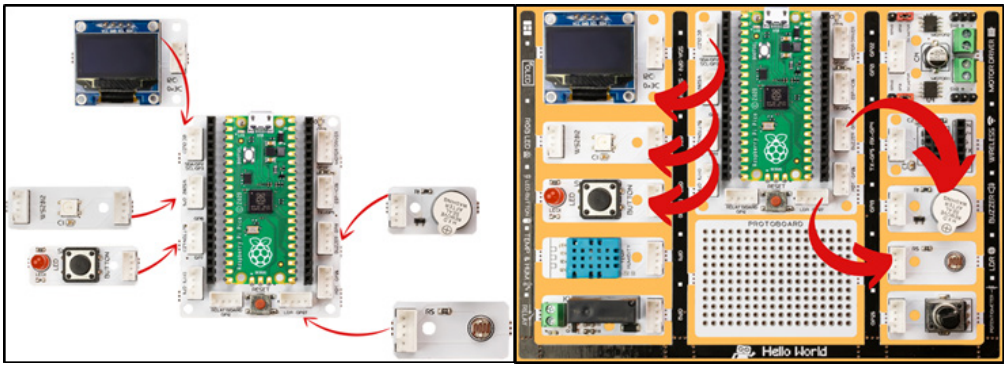
Global warming is affecting the climate of our world worse every day. Countries take many precautions and sign agreements to reduce the effects of global warming. The use of renewable energy sources and the efficient use of energy is an issue that needs attention everywhere, from factories to our rooms. Many reasons such as keeping road and park lighting on in cities due to human error, and the use of high energy consuming lighting tools reduce energy efficiency. Many electronic and digital systems are developed and programmed by engineers to measure the light, temperature and humidity values of the environment and ensure that they are used only when needed and in the right amounts.

In this project, we will create a timer alarm that adjusts for daylight using the light sensor in Picobricks.

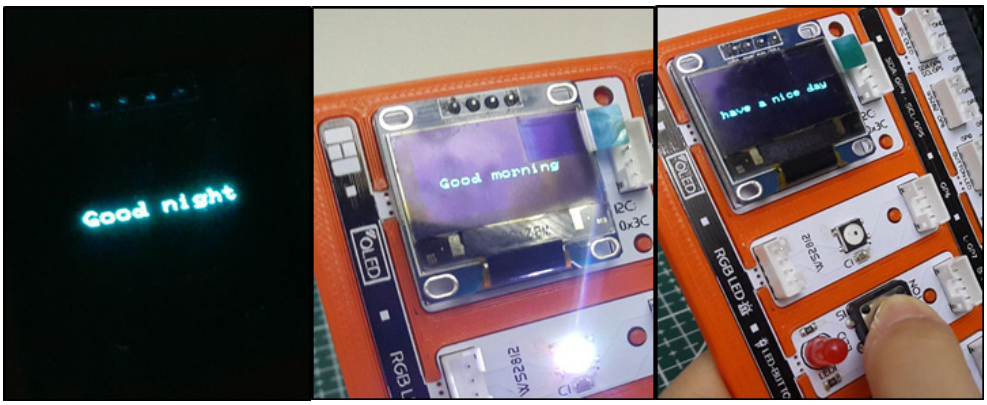
2.9.1. Project Details and Algorithm

In this project we will make a simple alarm application. The alarm system we will design is designed to sound automatically in the morning. For this, we will use LDR sensor in the project..At night, the OLED screen will display a good night message to the user, in the morning, an alarm will sound with a buzzer sound, a good morning message will be displayed on the screen, and the RGB LED will light up in white for light notification. The user will have to press the button of Picobricks to stop the alarm. After these processes, which continue until the alarm is stopped, when the button is pressed, the buzzer and RGB LED will turn off and a good day message will be displayed on the OLED screen.

2.9.2. Wiring Diagram



2.9.3. Project Image

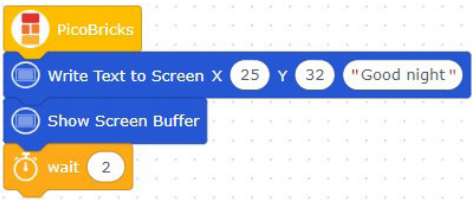


2.9.4. Project Proposal

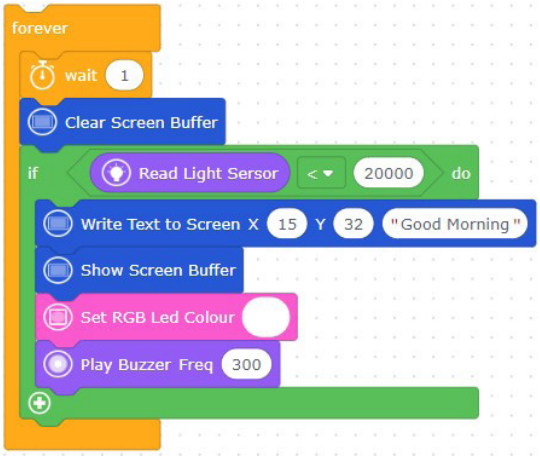
You can improve the project by adding a melody as an alarm sound instead of a beep. Or, instead of an alarm set according to daylight with the LDR sensor, you can develop an alarm that sounds at the specified time, where the time information is arranged via the button and OLED screen.

2.9.5. Coding the Project with PicoBricks IDE

Let's create the code blocks that, print the "Good night" text on the screen and wait for two seconds, when PicoBricks is started.

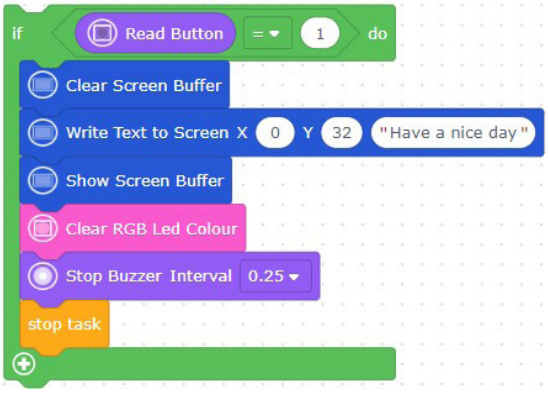


Create the condition and loop structures that are necessary for the project. Let's create the code blocks that, print the "Good Morning" text, turn on the buzzer and RGB LED if the value from the "Read Light Sensor" block is less than 2000.

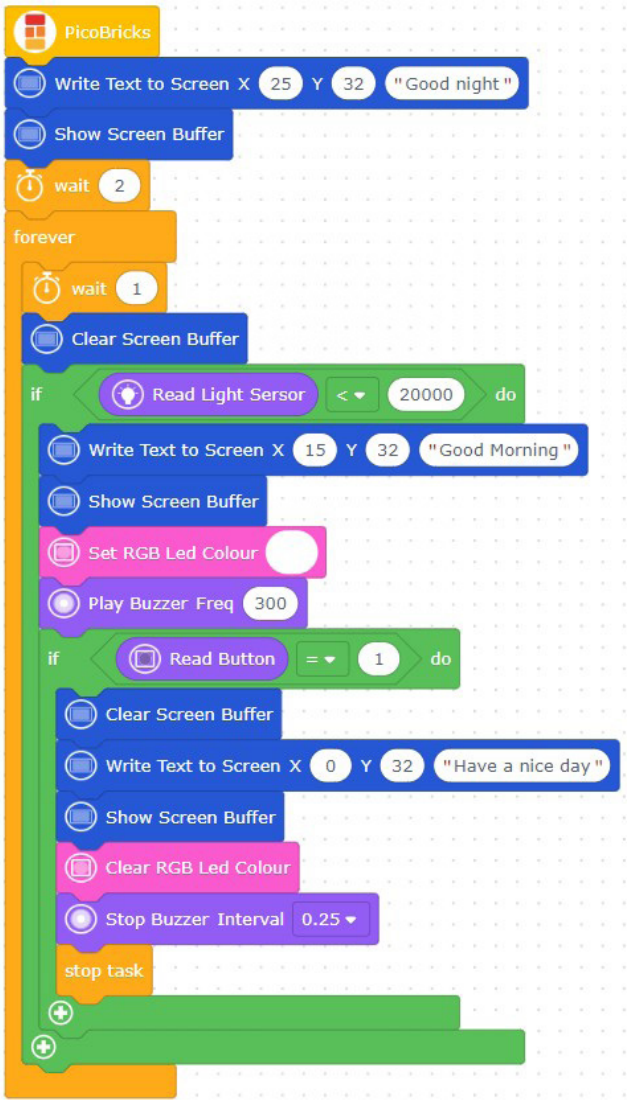


Create the code blocks that, clear the screen, print the “Have a nice day” text on the screen, turn off the RGB LED and stop the buzzer, when the button is pressed.

Let’s drag this block into the “if” block that you created firstly.



The code blocks are ready!



2.10. Know Your Color

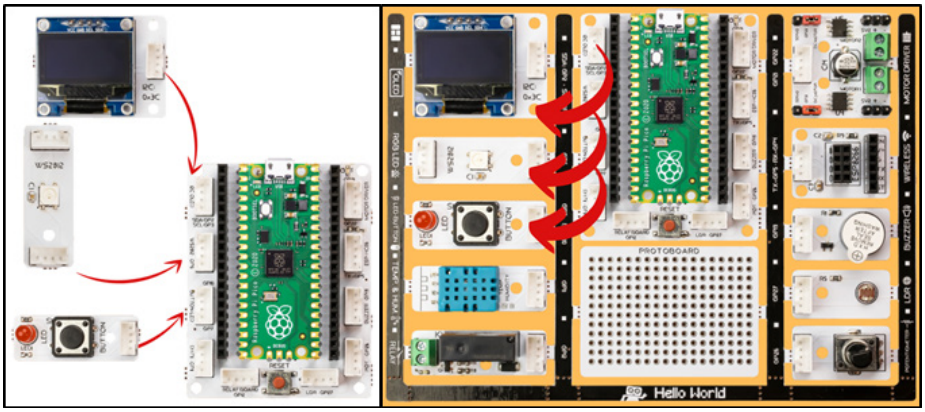
LEDs are often used on electronic systems. Each button can have small LEDs next to each option. By making a single LED light up in different colors, it is possible to do the work of more than one LED with a single LED. LEDs working in this type are called RGB LEDs. It takes its name from the initials of the color names Red, Green, Blue. Another advantage of this LED is that it can light up in mixtures of 3 primary colors. Purple, turquoise, orange...

In this project you will learn about the randomness used in every programming language. We will prepare a enjoyable game with the RGB LED, OLED screen and button module of Picobricks.

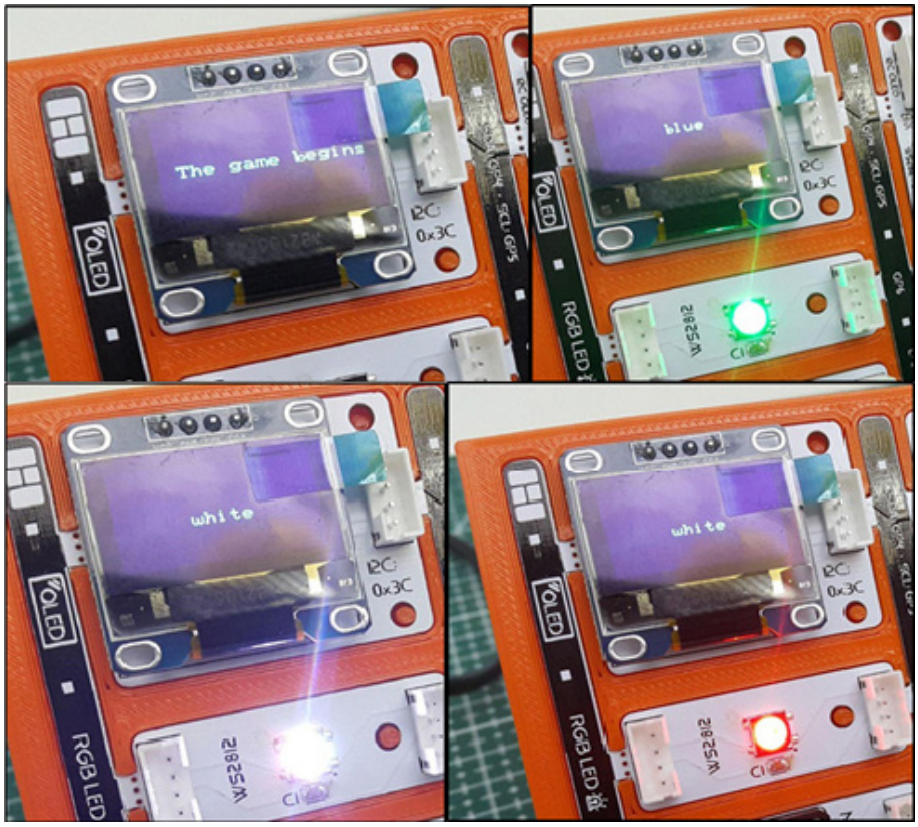
2.10.1. Project Details and Algorithm

The game we will build in the project will be built on the user knowing the colors correctly or incorrectly. One of the colors red, green, blue and white will light up randomly on the RGB LED on Picobricks, and the name of one of these four colors will be written randomly on the OLED screen at the same time. The user must press the button of Picobricks within 1.5 seconds to use the right of reply. The game will be repeated 10 times, each repetition will get 10 points if the user presses the button when the colors match, or if the user does not press the button when they do not match. If the user presses the button even though the colors do not match, he will lose 10 points. After ten repetitions, the user's score will be displayed on the OLED screen. If the user wishes, he may not use his right of reply by not pressing the button.

2.10.2. Wiring Diagram



2.10.3. Project Image



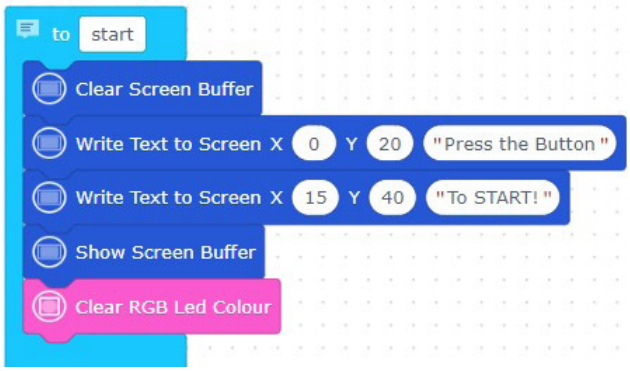
2.10.4. Project Proposal

You can make the game more enjoyable by making it a little more difficult. For example, you can speed up the game by reducing the repetition time of the colors. Or, instead of losing points when the user presses the button in the wrong place, you can finish the game and start it again.

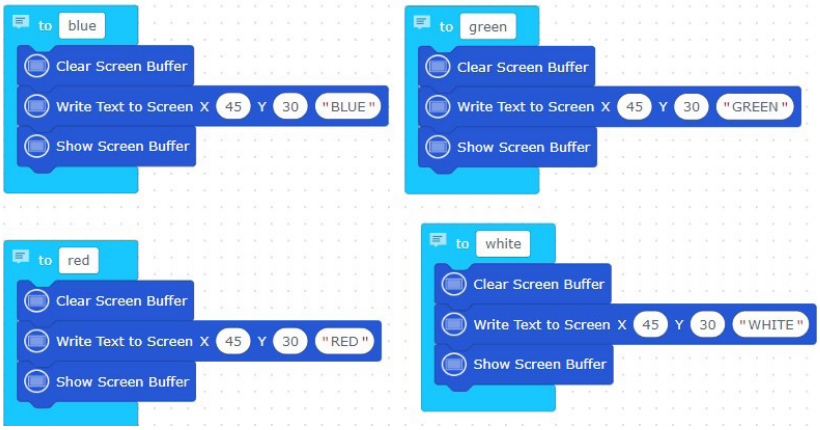
2.10.5. Coding the Project with PicoBricks IDE

Firstly, let's start the project by creating the necessary functions for the code.

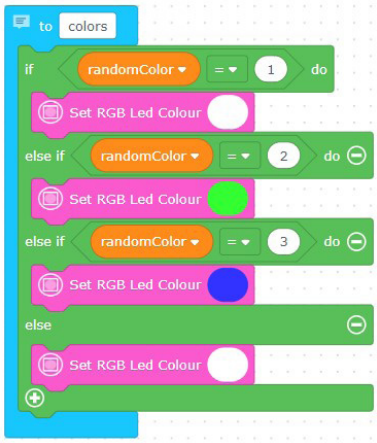
Let's turn off the RGB LED and determine the texts that is printed when the project starts by defining a function named as "start".



Define the color functions to print the name of each color you use in the project on the OLED screen.



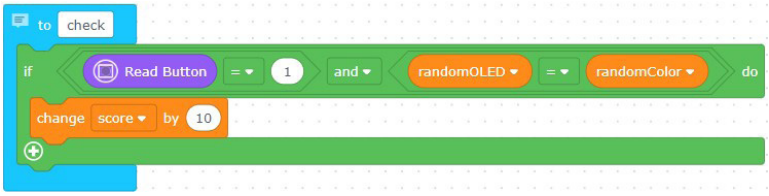
Turn on randomly one of four colors you have determined in the RGB LED module by defining a function named as “colors”.



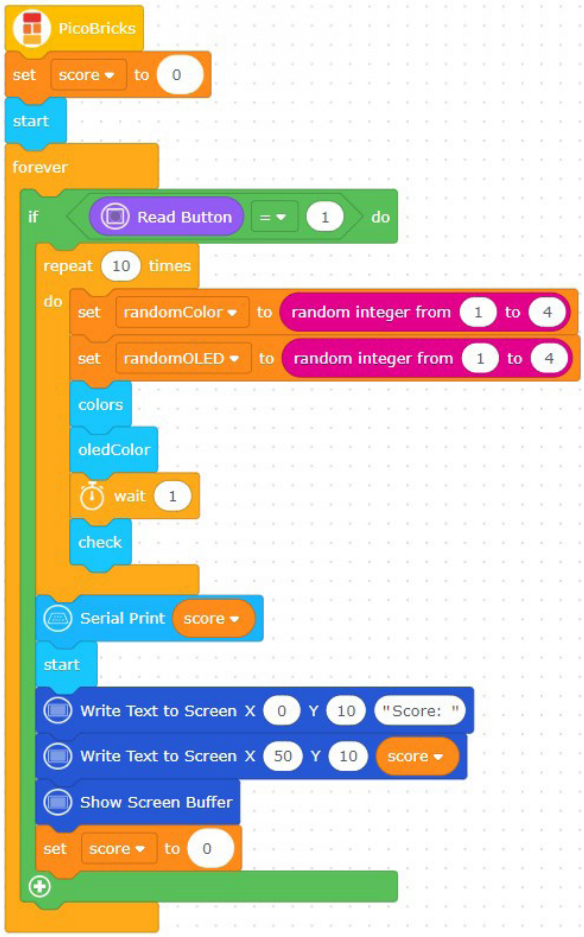
Print 4 colors you determined randomly on the screen by defining a function named as “oledcolor”.



By defining a function named as “check”, let’s create code blocks that increase the “score” variable by 10 if the color on the OLED screen and the color on the RGB LED are the same when the button is pressed.



Let’s define and call the functions you created by using loop and condition structures and complete the code.



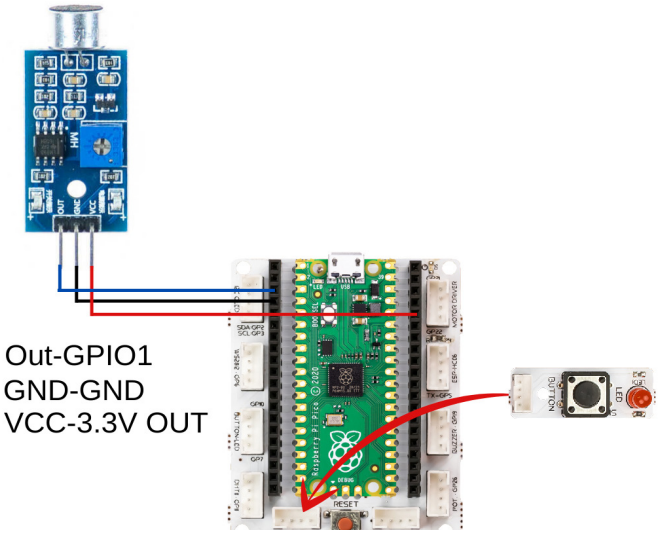
2.11. Magic Lamp

Most of us have seen lamps flashing magically or doors opening and closing with the sound of clapping in movies. There are set assistants who close these doors and turn off the lamps in the shootings. What if we did this automatically? There are sensors that convert the sound intensity change that we expect to occur in the environment into an electrical signal. These are called sound sensors.

2.12.1. Project Details and Algorithm

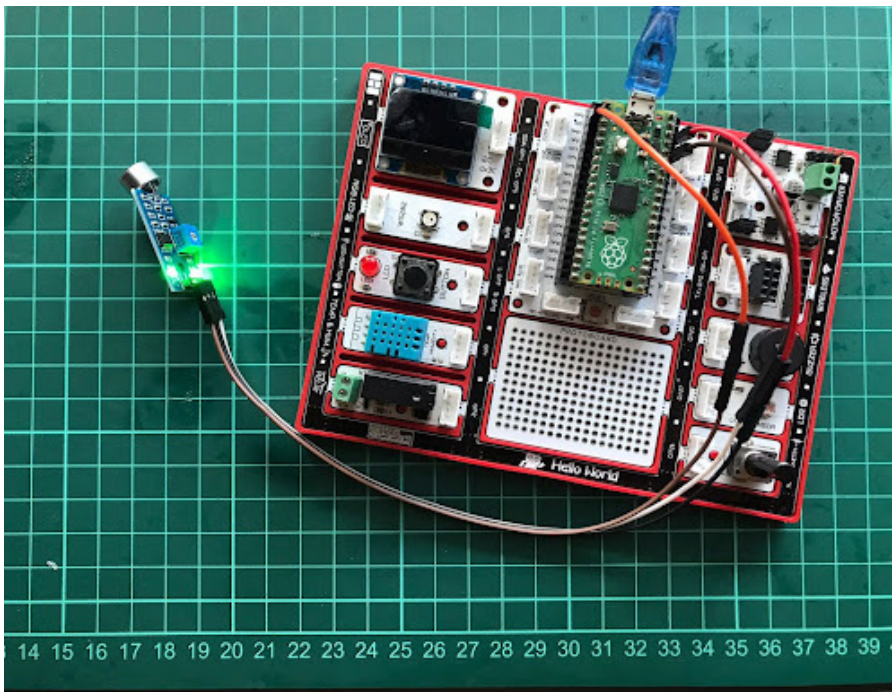
In this project, we will turn the LED module on the picobricks board on and off with the sound. In our project, which we will build using the Picobricks sound level sensor, we will perform the on-off operations by making a clap sound. As in previous projects, in projects where sensors are used, before we start to write the codes, it will make your progress easier to see what values the sensor sends in the operations we want to do by just running the sensor, and then writing the codes of the project based on these values.

2.12.2. Wiring Diagram



2.11.3. Construction Stages of the Project

During the construction of the project, two wire sockets and sockets were used. The two ends, which were cut by cutting the phase cable, were connected to the relay. You should pay attention to the insulation with electrical tape so that a dangerous situation does not occur when you cut the other wire. If you use a three-wire socket, you must cut the brown wire with the phase lead and connect it to the relay.

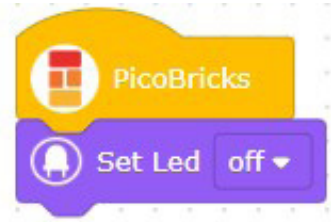


2.11.4. Project Proposal

You can present the player with instructions and notifications on the OLED screen. In addition, you can prepare a more exciting game by showing on the OLED screen how many milliseconds after the game starts, the game is over.

2.11.5. Coding the Project PicoBricks IDE

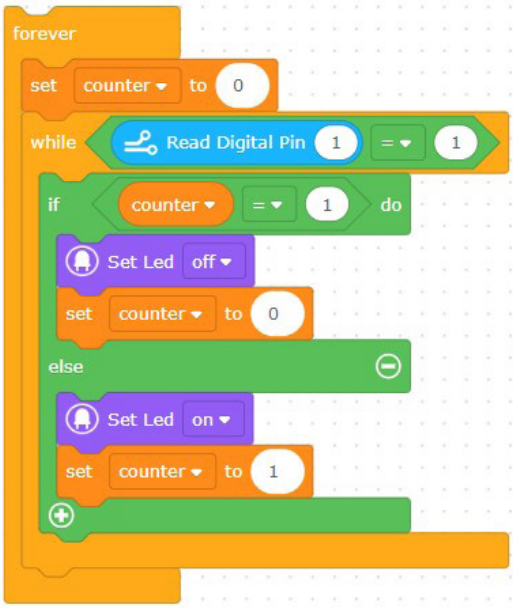
Before creating loop and condition structures, let's drag the code blocks that, turn off the LED, when the project is started.



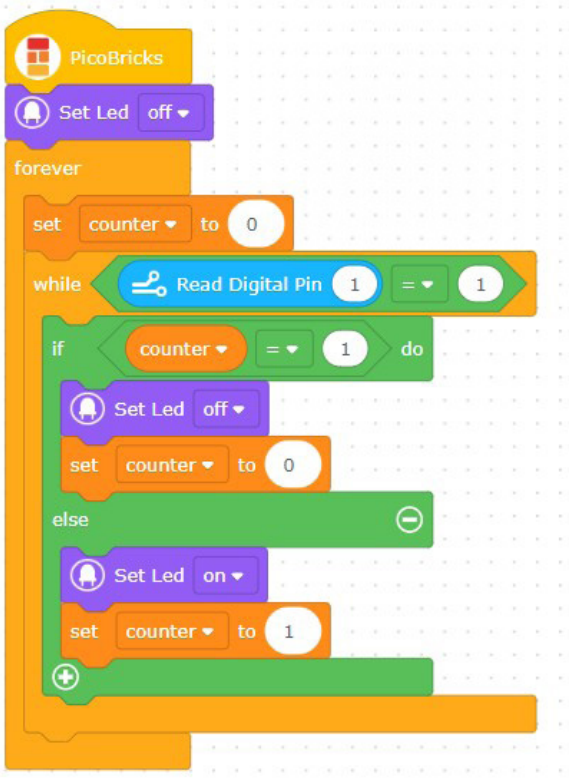
Let's determine the condition structures and operations that will repeat in the project. To use the digital data from the sound sensor in condition and loop structures, we will use the "read digital pin" block from the "basic" blocks. Write the pin which we connected the sound sensor in the field inside the block as follows.



Create loop and condition structures that turn on the LED if the digital data from the sound sensor is “1” and the LED is off, and turn off the LED otherwise.



The code blocks are ready!



2.12. Smart Cooler

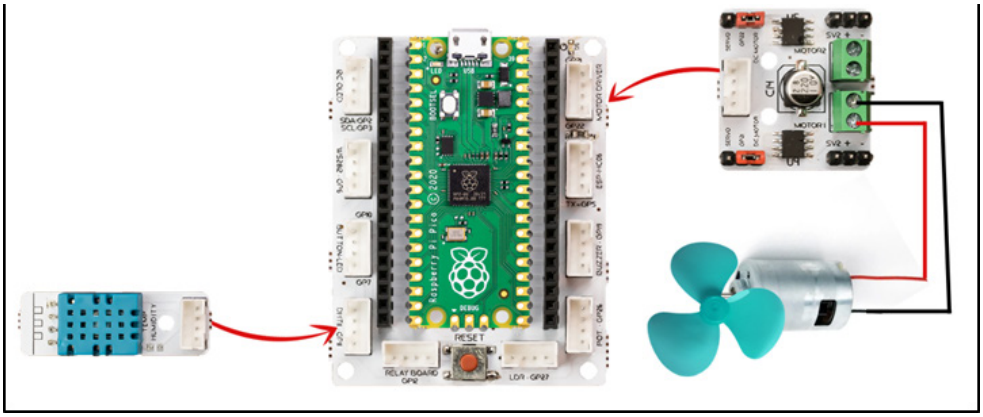
Air conditioners are used to cool in the summer and warm up in the winter. Air conditioners adjust the degree of heating and cooling according to the temperature of the environment. While cooking the food, the ovens try to rise to the temperature value set by the user and maintain that temperature. These two electronic devices use special temperature sensors to control the temperature. In addition, temperature and humidity are measured together in greenhouses. In order to keep these two values in balance at the desired level, it is tried to provide air flow with the fan.

In Picobricks, you can measure temperature and humidity separately and interact with the environment with these measurements. In this project, we will prepare a cooling system that automatically adjusts the fan speed according to the temperature with Picobricks. In this way, you will learn the DC motor operating system and motor speed adjustment.

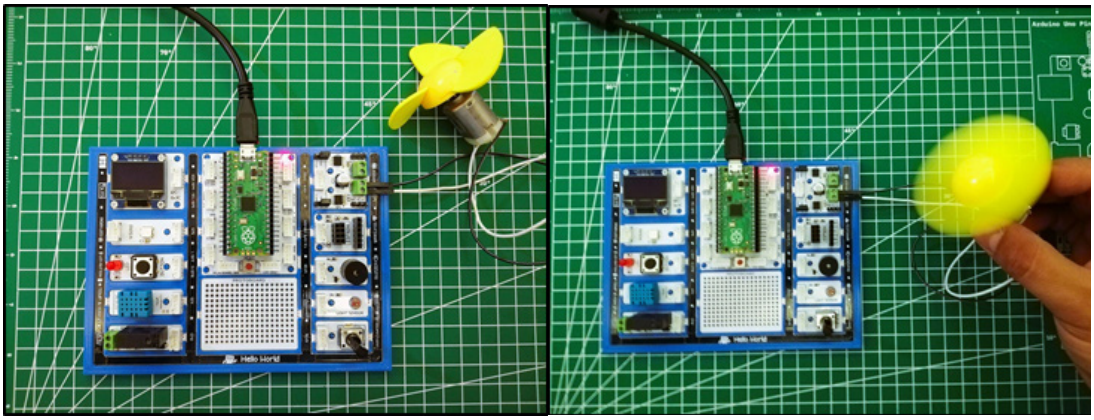
2.12.1. Project Details and Algorithm

In our project, we will firstly display the temperature values measured by the DHT11 temperature and humidity sensor on Picobricks. Then, we will define a temperature limit and write the necessary codes for the DC motor connected to Picobricks to start rotating when the temperature value from the DHT11 module reaches this limit, and for the DC motor to stop when the temperature value falls below the limit we have determined.

2.12.2. Wiring Diagram



2.12.3. Project Image



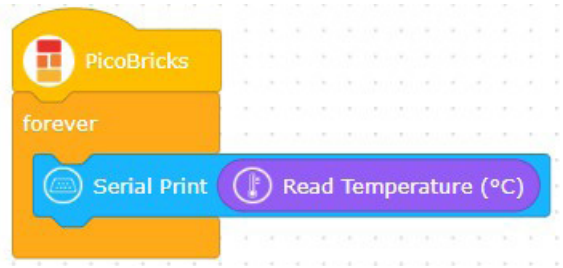
2.12.4. Project Proposal

Using the OLED screen on Picobricks, you can print the temperature on the screen and keep track the temperature at which the fan is activated.

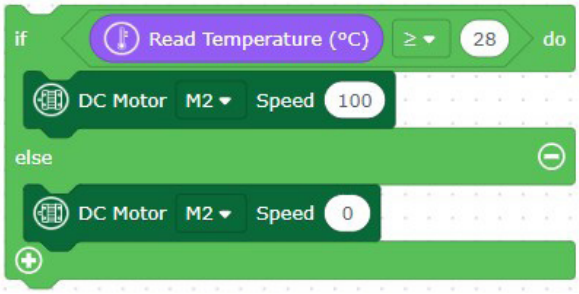
Picobricks has a modular structure, modules can be separated by breaking and can be used by connecting to Pico board with grove cables. By mounting the smart cooling circuit we made in our project to the robot car chassis, you can develop a project that navigates autonomously in your environment and cools the environment at the same time.

2.12.5. Coding the Project with PicoBricks IDE

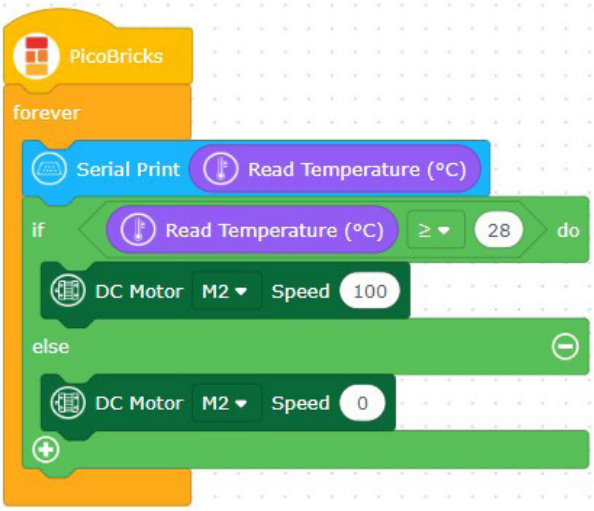
In order to use the analog data from PicoBricks DHT module in the loop and condition structure, determine the environment temperature by printing the value from DHT module to the serial monitor.



Let's create the code blocks that, run the second DC motor if the value of "Read Temperature" block is 28 or more than 28, otherwise stop the second DC motor.



The code blocks are ready!



2.13. Buzz Wire Game

Projects don't always have to be about solving problems and making things easier. You can also prepare projects to have fun and develop yourself. Attention and concentration are features that many people want to develop. The applications that we can do with this are quite interesting. How about making Buzz Wire Game with Picobricks?

You must have heard the expression that computers work with 0s and 1s. 0 represents the absence of electricity and 1 represents its presence. 0 and 1's come together with a certain number and sequence of combinations to form meaningful data. In electronic systems, 0s and 1s can be used to directly control a situation. Is the door closed or not? Is the light on or off? Is the irrigation system on or not? In order to obtain such information, a status check is carried out.

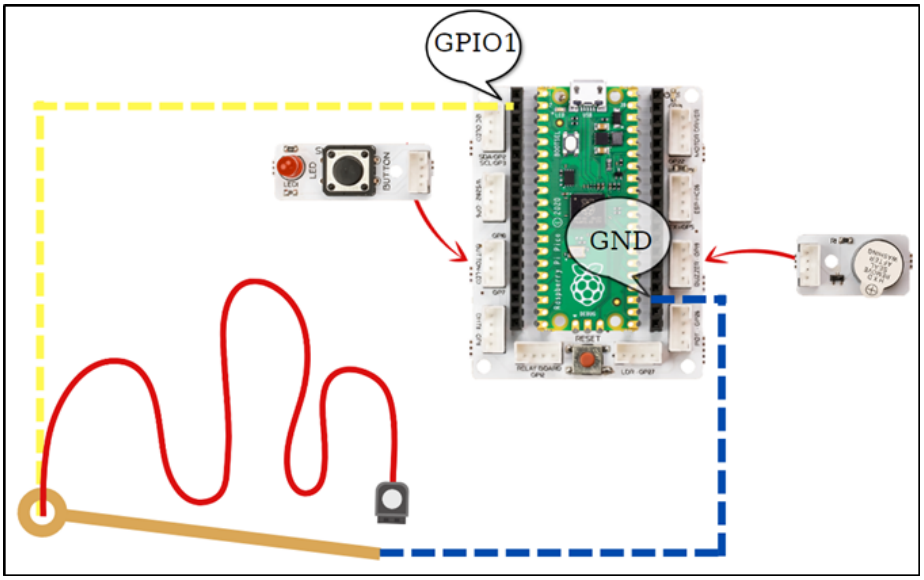
In this project, we will electronically prepare the attention and concentration developer Buzz Wire Game with the help of a conductor wire using the buzzer and LED module with Picobricks. While preparing this project, you will have learned an input technique that is not a button but will be used like a button.

2.13.1. Project Details and Algorithm

To prepare the project, you need 2 male-male jumper cables and a 15 cm long conductor bendable wire. When the player is ready, it will be asked to press the button to start the game. If the jumper cable touches the conductor wire in the player's hand when the button is pressed, Picobricks will detect this and give an audible and written warning. The time from the start of the game to the end will also be displayed on the OLED screen.

We reset the timer after the user presses the button. Then we will give a voltage of 3.3V to the conductor wire connected to the GPIO1 pin of Picobricks. One end of the cable held by the player will be connected to the GND pin on the Picobricks. If the player touches the jumper cable in his hand to the conductive wire, the GPIO1 pin will drop to the Passive/Off/0 position. Then, it will announce that the game is over, and there will be light, written and audio feedback, then the elapsed time will be shown on the OLED screen in milliseconds. After 5 seconds, the player will be prompted to press the button to restart.

2.13.2. Wiring Diagram



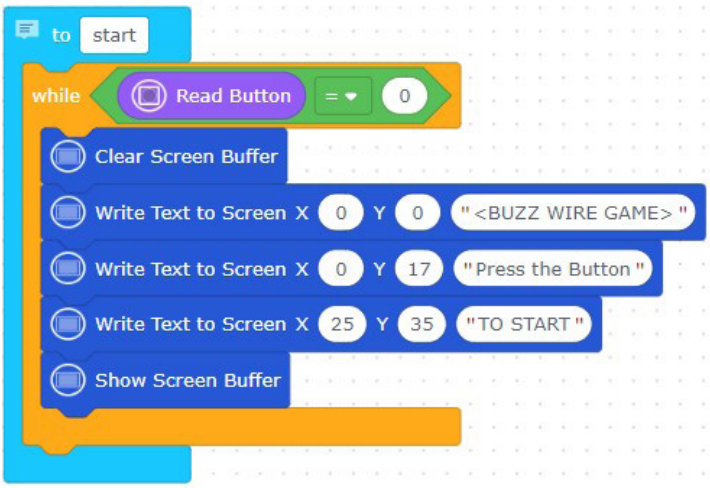
2.13.3. Project Proposal

You can make physical and software improvements to the project. By covering the start and end points with insulating tape, you can prevent the player from having problems starting and finishing the game. In terms of software, when the player brings the cable to the other end without touching the wire, press the button and you can see the score on the OLED screen.

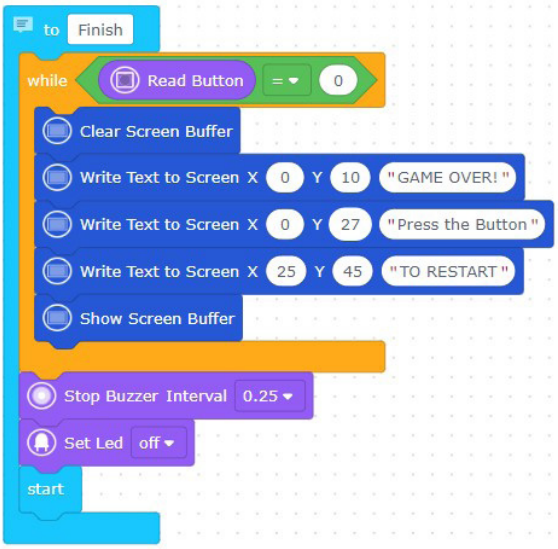
2.13.4. Coding the Project with PicoBricks IDE

Firstly, let's start the project by creating the necessary functions for the code.

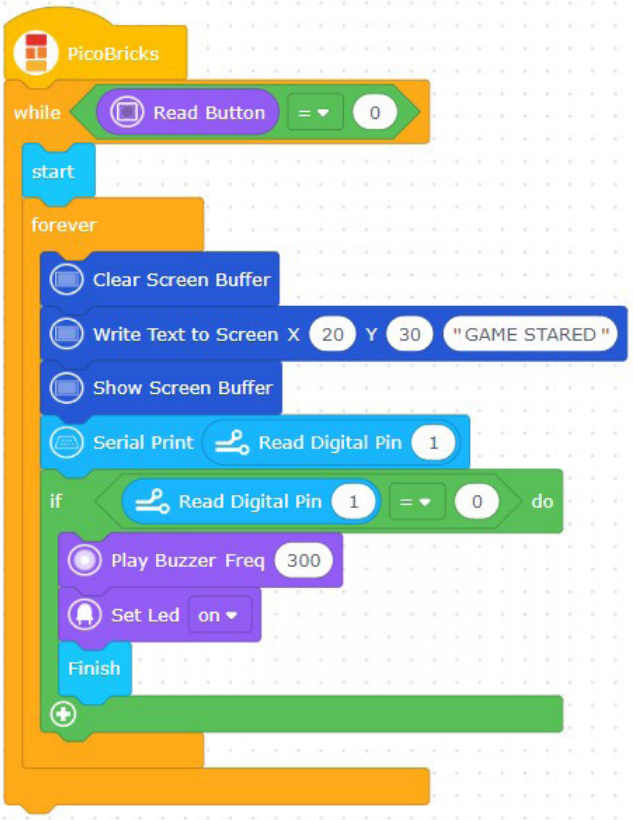
- Let's define a function named as "start" to determine the statements that will be printed on the screen, when the project starts. In this function, the texts on the screen will not be deleted until the button is pressed.



Define a function named as “finish” that, determines the statements which will be printed on the OLED screen when the project is finished. In this function, “start” function will run again after the button is pressed.



Let’s call the functions created by using the necessary condition and loop structures. Create the code blocks that, run buzzer, turn on LED and call the “finish” function if the value from GPIO 01 pin is “0”.



2.14. Dinosaur Game

If the electronic systems to be developed will fulfill their duties by pushing, pulling, turning, lifting, lowering, etc., pneumatic systems or electric motor systems are used as actuators in the project. Picobricks supports two different engine types so that you can produce systems that can activate the codes you write in your projects. DC motor and Servo motors in which the movements of DC motors are regulated electronically. Servo motors are motors that rotate to that angle when the rotation angle value is given. In RC boats, servo motors are used with the same logic to change the direction of the vehicle. In addition, advanced servo motors known as smart continuous servos, which can rotate full-round, are also used in the wheels of the smart vacuum cleaners we use in our homes.

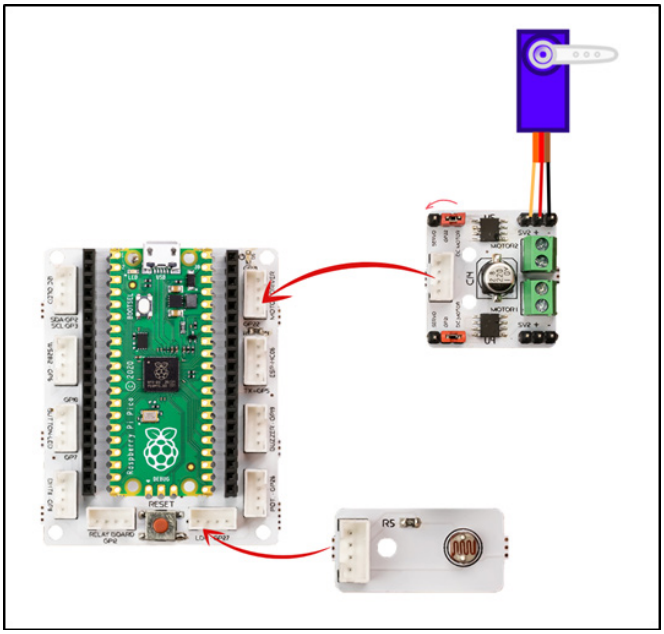
In this project you will learn how to control Servo motors with PicoBricks.

2.14.1. Project Details and Algorithm

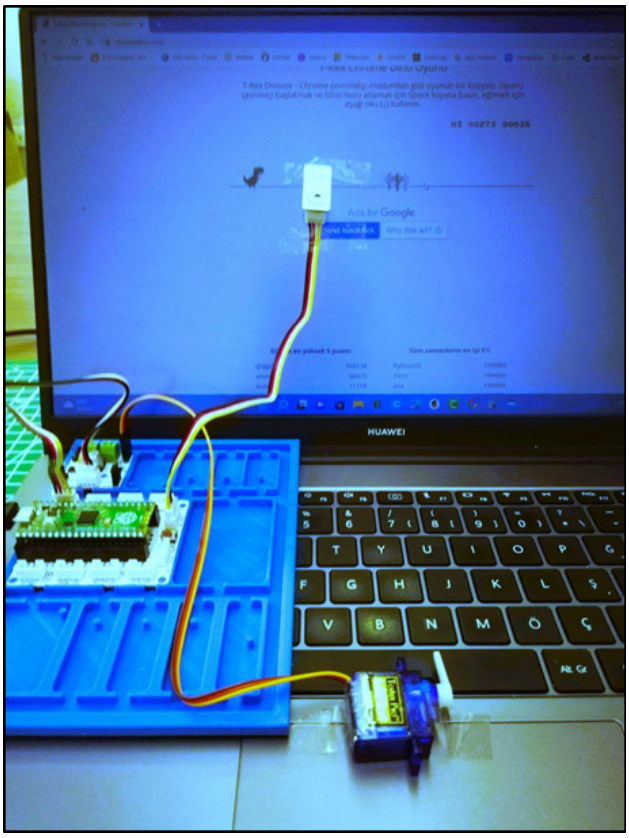
While writing the project codes, we will first fix the LDR sensor on the computer screen and read the sensor data on the white and black background, then write the necessary codes for the servo motor to move according to these data.In this project, we will automatically play Google Chrome offline dinasour game to picobricks. In the game, Picobricks will automatically control the dinosaur's movements by detecting obstacles. We will use the picobricks LDR sensor to detect the obstacles in front of the dinosaur during the game. LDR can send analog signals by measuring the amount of light touching the sensor surface. By fixing the sensor on the computer screen, we can detect if there is an obstacle in front of the dinosaur by taking advantage of the difference in the amount of light between the white and black colors. When an obstacle is detected, we can use a servo motor to automatically press the spacebar on the keyboard. In this way, the dinosaur will easily overcome the obstacles. While writing the project codes, we will firstly fix the LDR sensor on the computer screen and read the sensor data on the white and black background, then write the necessary codes for the servo motor to move according to these data.

2.14.2. Wiring Diagram

Note: There are triple pins on the right and left side of the motor driver grove cable entry and these pins are short-circuited with 2 jumpers. When using a DC motor, the jumper that should be attached on the DC motor side should be removed when using a servo motor and attached to the servo side.



2.14.3. Project Image



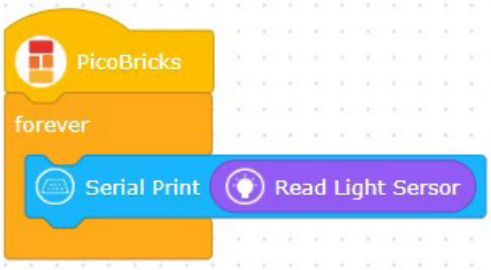
2.14.4. Project Proposal

At first in the game, the ground color is white and the figures are black. After a certain stage, the colors are reversed. For this reason, LDR sensor data is changing. To solve this problem, you can use variables and functions to run one code group when the game is on a white background, another code group when it is on a black background, or you can install a second LDR sensor to detect this difference.

Picobricks and its modules allow us to develop many projects from simple to complex. You can also use it in different games such as minecraft by developing this project, which we automatically play a computer game that we play in daily life on Picobricks.

2.14.5. Coding the Project with PicoBricks IDE

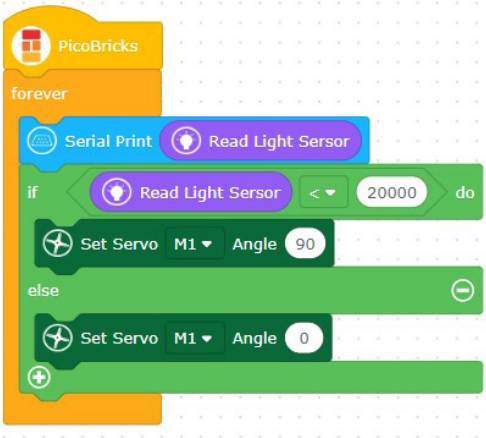
Let's determine the light value that you will use in the project by printing the light value of the environment to the serial monitor.



You should control the angles of the servo motor by using the light value in the condition and loop structures.



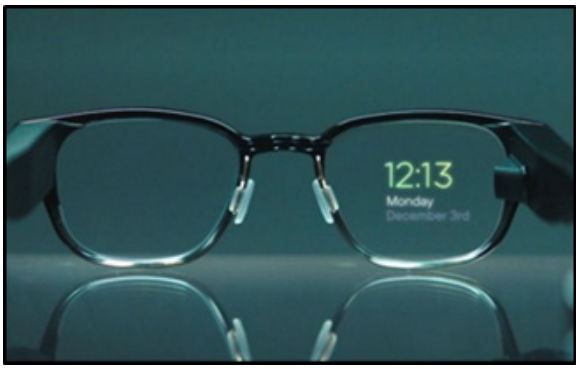
The code blocks are ready!



2.15. Night and Day

How about playing the Night and Day game you played at school electronically?The game of night and day is a game in which we put our head on the table when our teacher says night, and raise our heads when our teacher says day. This game will be a game that you will use your attention and reflex. In this project, we will use a 0.96" 128x64 pixel I2C OLED display. Since OLED screens can be used as an artificial light source, you can enlarge the characters on the screen using lenses and mirrors and reflect them on the desired plane. Systems that can reflect information, road and traffic information on smart glasses and automobile windows can be made using OLED screens.

Light sensors are sensors that can measure the light levels of the environment they are in, also called photodiodes. The electrical conductivity of the sensor exposed to light changes. We can control the light sensor by coding and develop electronic systems that affect the amount of light.

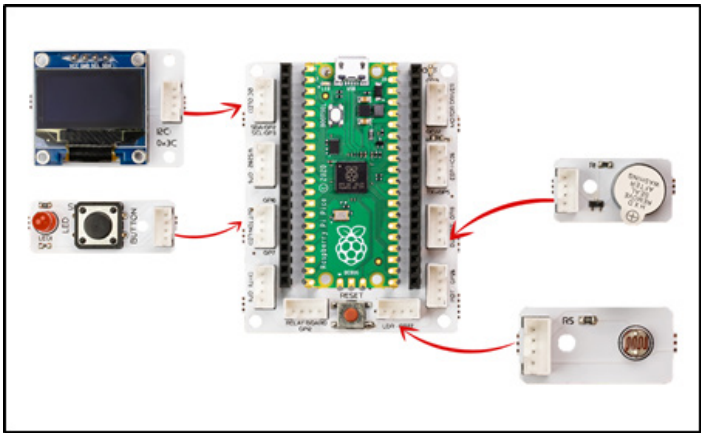




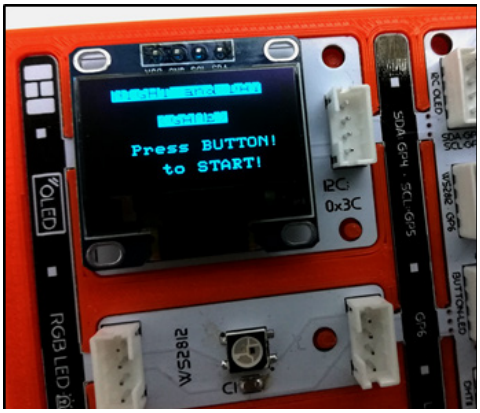
2.15.1. Project Details and Algorithm

First we will ask the player to press the button to start the game. Then we will make the OLED screen of PicoBricks display NIGHT and DAY randomly for 2 seconds each. The player should cover the LDR sensor with his hand within 2 seconds if the word written on the OLED screen is NIGHT, and if the word DAY is written on the OLED screen, the player should raise his hand over the LDR sensor. Each correct response of the player will earn 10 points. In case of wrong response, the game will be over and there will be a written statement on the screen stating the end of the game, a different tone will sound from the buzzer, and the score information will be displayed on the OLED screen. If the player gives a total of 10 correct responses and gets 100 points, the phrase "Congratulation" will be displayed on the OLED screen and the buzzer will play notes in different tones.

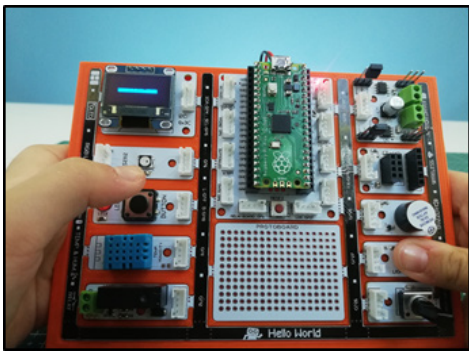
2.15.2. Wiring Diagram

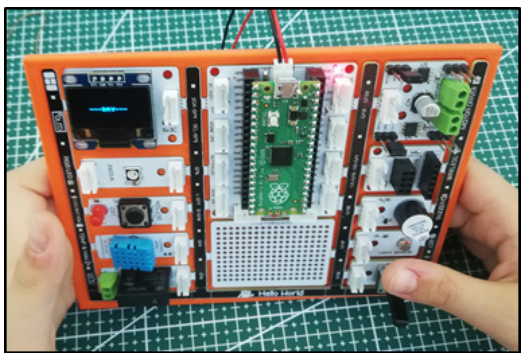


2.15.3. Project Image











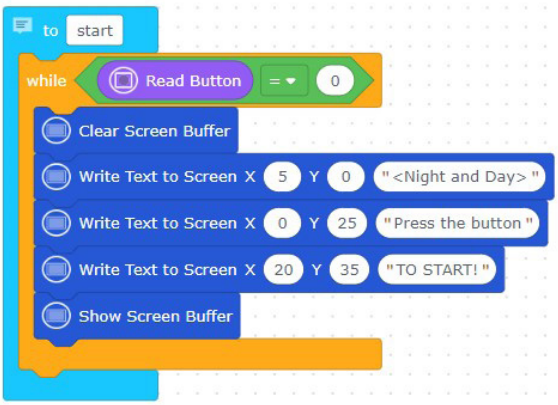
2.15.4. Project Proposal

You can develop the project by taking the values that the LDR sensor sends to the project according to the environment you are in, and automatically determining the limit to be processed according to the sensor value in the game, that is, by adding LDR sensor value calibration codes. You can add difficulty level to the game. With the potentiometer, the difficulty level can be selected as easy, medium and hard. When easy is selected, the change time for words can be 2 seconds, 1.5 seconds when medium is selected, 1 second when hard is selected.

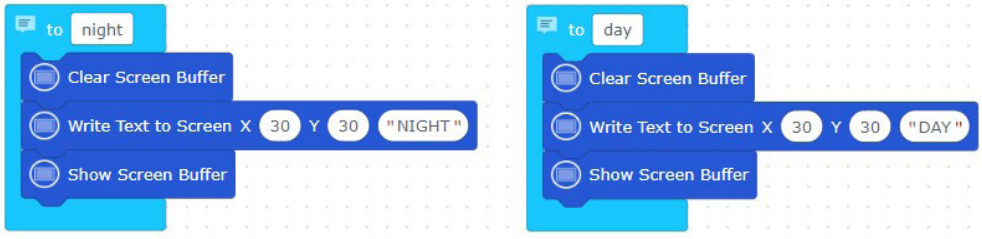
2.15.5. Coding the Project with PicoBricks IDE

Firstly, let's start the project by creating the necessary functions for the code.

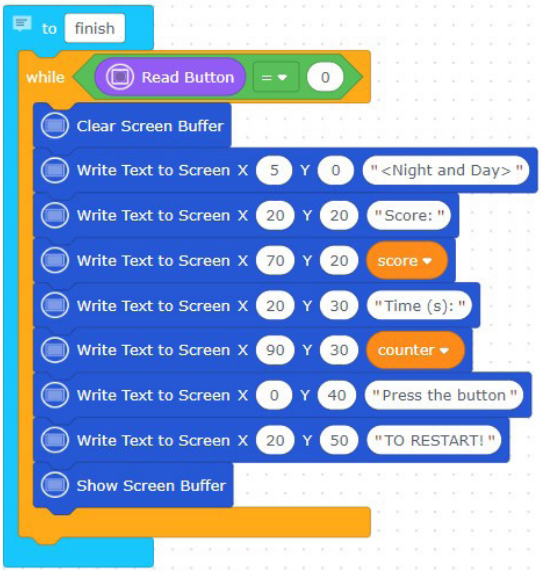
- By determining a function named as "start", let's determine the statement that will be written on the OLED screen until the button is pressed when the project starts.



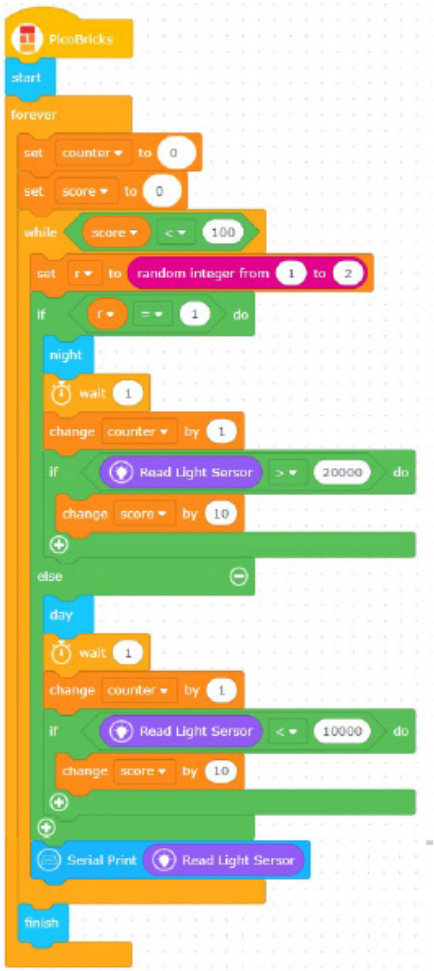
- Let's define two functions named as "night" and "day" for "NIGHT" and "DAY" statements that will be written on the OLED screen randomly.



- Define a function named as “finish” to determine the statements that will be written on the screen until the button is pressed when the game is over.



- Let’s create the code block structure by using the loop and condition structures that call the functions. Let’s create a structure that randomly generates numbers between 1 and 2, prints “NIGHT” on the screen when the number is 1, and prints “DAY” on the screen when the number is 2, measures the value of LDR according to the value and determines the score of the player.



2.16. Voice Controlled Robot Car

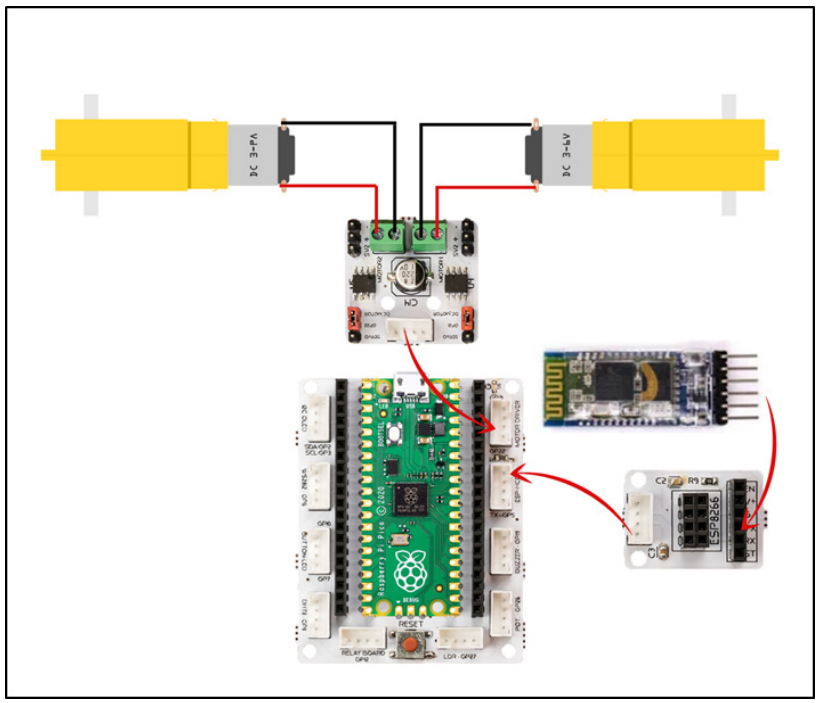
Developing and continuing to develop artificial intelligence applications recognize human characteristics, learn and try to behave like people. We can express artificial intelligence as software that can learn in its shortest form. Sometimes it learns the image, sometimes the sound, and sometimes by using the data it collects from the sensors. It does this thanks to the algorithms determined by the developers, and it helps in the decision-making processes in the areas it is used according to the results it has achieved. In short, artificial intelligence applications are now used in situations where the decision-making process needs to be done quickly and without errors. From the marketing field to the defense industry, from education to health, from economy to entertainment, artificial intelligence increases efficiency and reduces costs.

In this project we will do with PicoBricks, we will make a 2WD car that you can control by talking. PicoBricks allows you to communicate wirelessly with 2 6V DC motors and bluetooth.

2.16.1. Project Details and Algorithm

In the project, the robot car kit that comes out of the set will be assembled and controlled via mobile phone. The HC05 bluetooth module is a module that enables us to communicate wirelessly between PicoBricks and a mobile phone. Thanks to the mobile application installed on the mobile phone in the project, the commands sent from the phone will be transmitted to PicoBricks via the HC05 module and the robot car will move according to these data. We can direct the robot car with the forward, backward, right, left buttons from the mobile phone, as well as send data to PicoBricks with voice command. In the project, we will give voice commands to control the movements of the robot car.

2.16.2. Wiring Diagram



2.16.3. Project Proposal

In this project, we moved the robot car by giving voice commands via the mobile application we installed on the mobile phone. You can control the mechanism with voice commands or buttons by connecting the HC05 bluetooth module to the pan-tilt mechanism in the two-axis robot arm project. Likewise, you can try a mobile application where you can control the robot car in this project using buttons instead of voice commands, or you can develop a mobile application specific to your project with the MIT Appinventor editor.

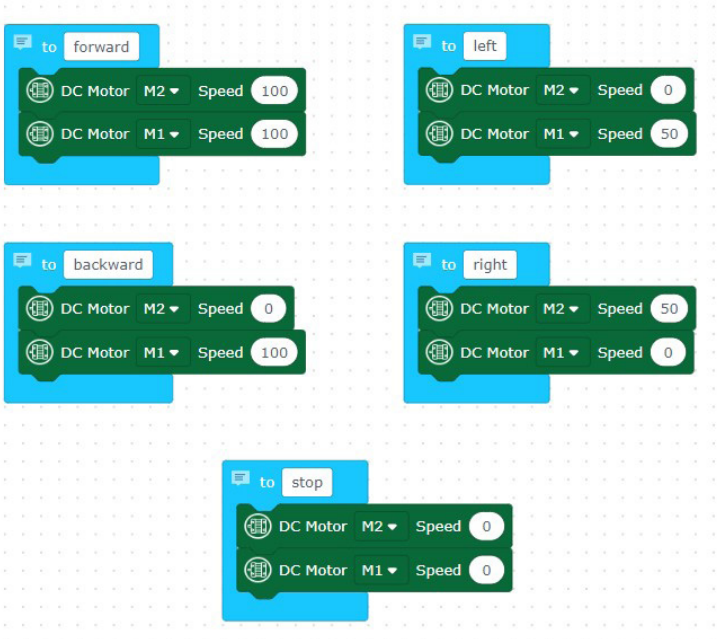
With the HC05 Bluetooth module, you can operate not only the motor driver and motor, but also other modules on PicoBricks. For example, you can light the RGB LED in any color you want through the mobile application, read the temperature and humidity values from the DHT11 module, the light values on the LDR sensor, and print texts on the OLED screen. There is a mobile application specially written for these processes with the MIT Appinventor editor, and ready-made codes written in Microblocks to automatically run the data coming from the application. You can run all these features by downloading and running the Microblocks file from the link below and by downloading the android apk file and installing it on your phone.

[Download Link]

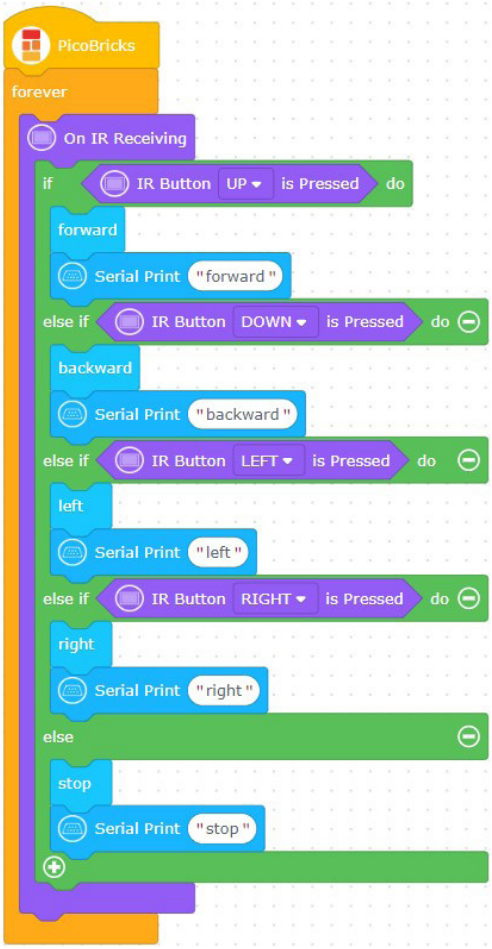
2.16.4. Coding the Project with PicoBricks IDE (V1.2 IR)

Firstly, let's start the project by creating the necessary functions for the code.

- Let's define five functions named as "forward", "left", "backward", "right" and "stop" to determine the movements of the car when IR remote buttons are pressed.



- Let's create code blocks that move the robot car by calling these five functions that are created when the keys of the IR control are pressed, according to the pressed key.



2.17. Two Axis Robot Arm

Robot arms have replaced human power in the industrial field. In factories, robotic arms undertake the tasks of carrying and turning loads of weights and sizes that cannot be carried by a human. Being able to be positioned with a precision of one thousandth of a millimeter is above the sensitivity that a human hand can exhibit. When you watch the production videos of automobile factories, you will see how vital the robot arms are. The reason why they are called robots is that they can be programmed to do the same work with endless repetitions. The reason why it is called an arm is because it has an articulated structure like our arms. How many different directions a robot arm has the ability to rotate and move is expressed as axes. Robot arms are also used for carving and shaping aluminum and various metals. These devices, which are referred to as 7-axis CNC Routers, can shape metals like a sculptor shapes mud.

According to the purpose of use in robot arms, stepper motor and servo motors, which are a kind of electric motor, are used. PicoBricks allows you to make projects with servo motors.

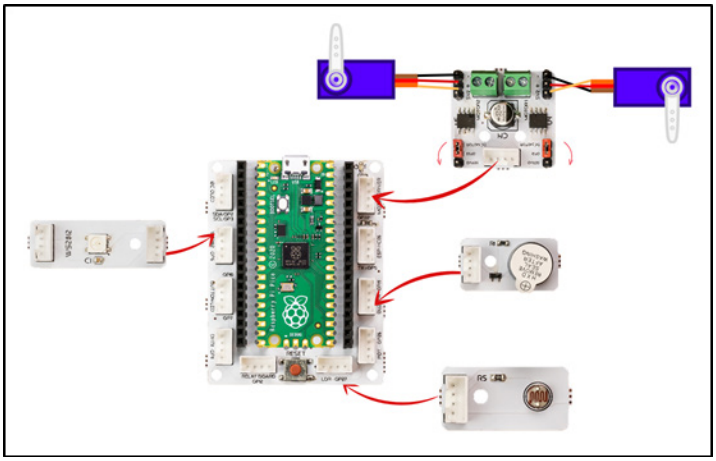
2.17.1. Project Details and Algorithm

In preparation for the installation, we will first write and upload the codes to set the servo motors to 0 degrees. When an object is placed on the LDR sensor, the robot arm will bend down and close its open gripper. After the gripper is closed, the robot arm will rise again. As a result of each movement of the robot arm, a short beep will be heard from the buzzer. The RGB LED will glow red when an object is placed on the LDR sensor. When the object is held by the robot arm and lifted into the air, the RGB LED will turn green.

Servo motor movements are very fast. In order to slow down the movement, we will code the servo motors with a total of 90 degrees of movement, 2 degrees each at 30 millisecond intervals. We're not going to do this for the gripper to close.

In order for the servo to perform its holding and releasing function, print and assemble the necessary parts from the 3D printer from the **link here.**

2.17.2. Wiring Diagram

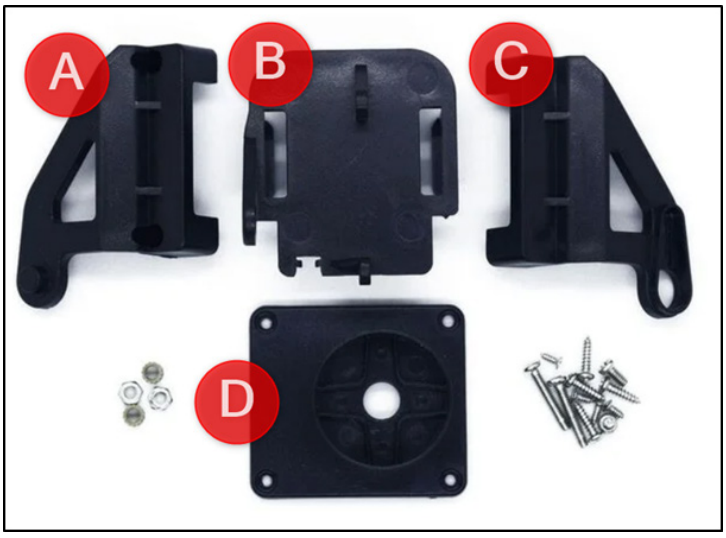


2.17.3. Project Proposal

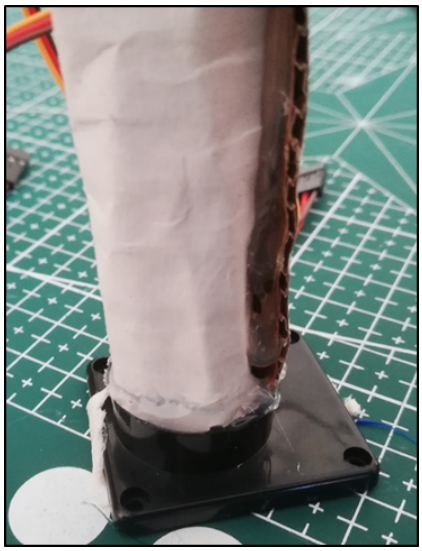
By adding the HC05 module to the 2 axis robot arm project, you can develop it by controlling it from your mobile phone with the mobile application.

2.17.4. Construction Stages of the Poject

Prepare the parts of the Pan-Tilt kit to prepare the project. Carry your 3D printed parts, waste cardboard pieces, hot silicone glue and scissors with you.



1. First of all, we will prepare the fixed arm of the robot arm. Make an 8 cm high cardboard cylinder into the rounded part of part D. Place it on the D part and stick it with silicone.



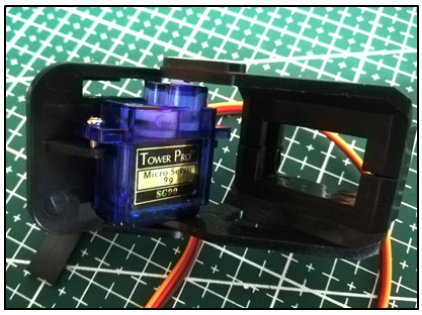
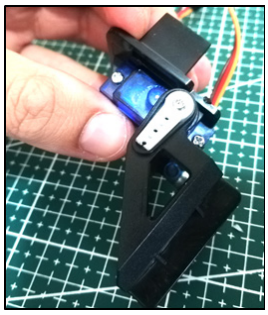
2. Place the head that came out of the servo motor package on the C part by shortening it a little. Fix with the smallest screws from the Pan Tilt kit.



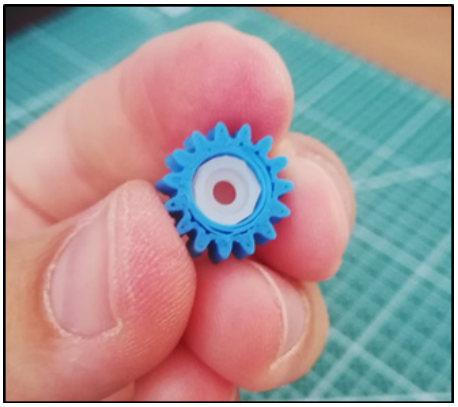
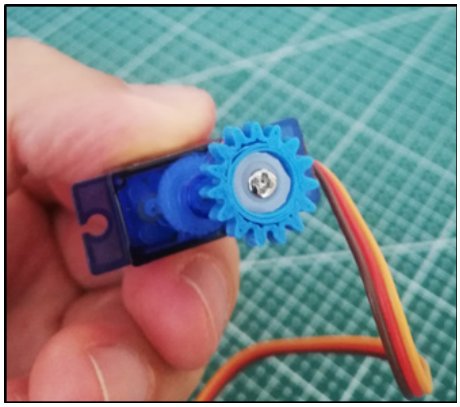
3. Fix parts A and C together with 2 pointed screws.



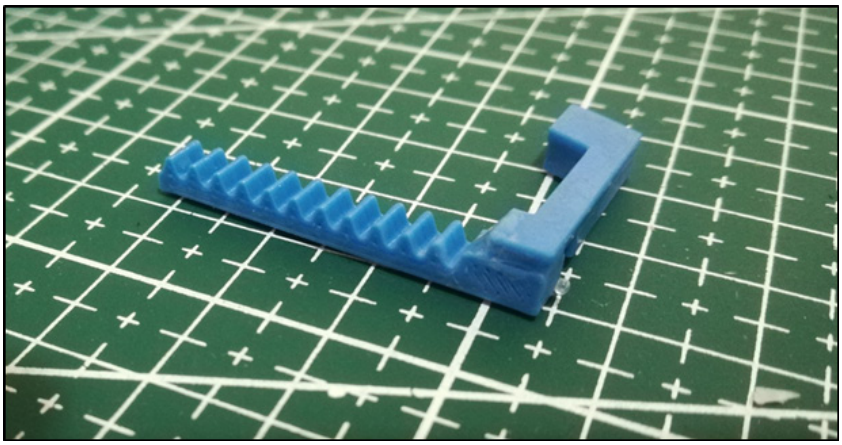
4. Internally attach the servo motor to part C. Then place the servo motor on part B and screw it.

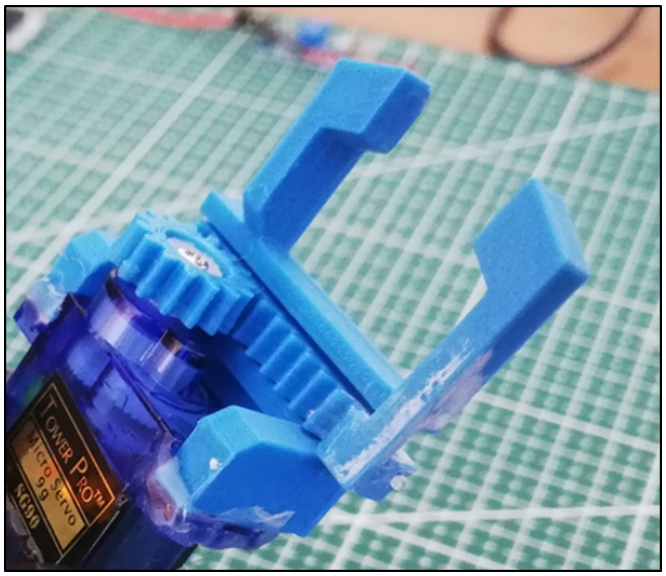
5. For the holder, cut one of the servo motor heads in the middle of the gear part that you printed on the 3D printer and place it into the gear. Then screw it to the servo motor.

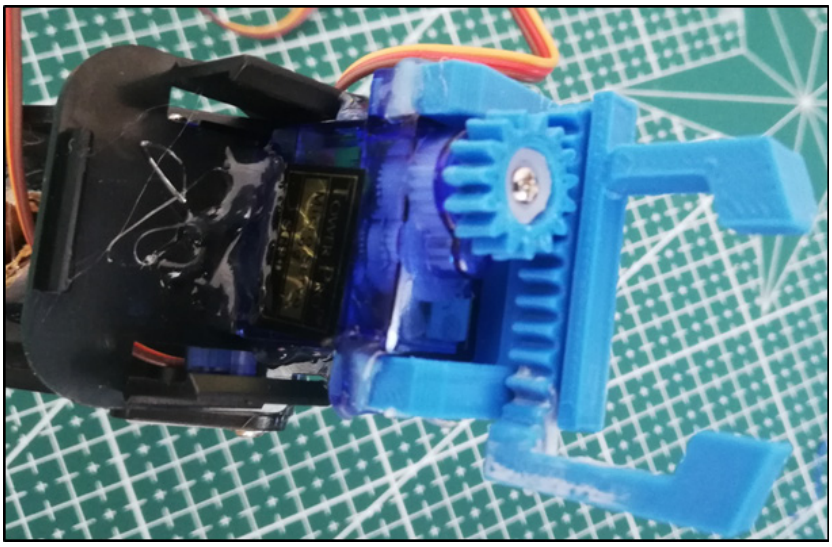
6. Adhere together the 3D printed Linear gear and the handle with strong adhesive.



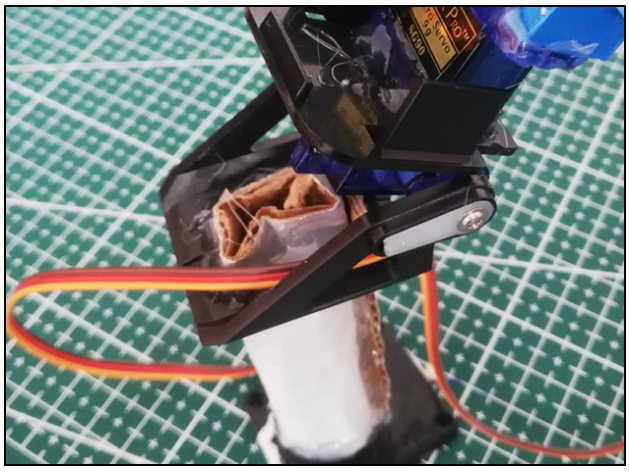
7. Place the servo in the 3D print holder and fix it. You can do this with hot silicone or by screwing. When placing the servo gear on the linear gear, make sure it is fully open.



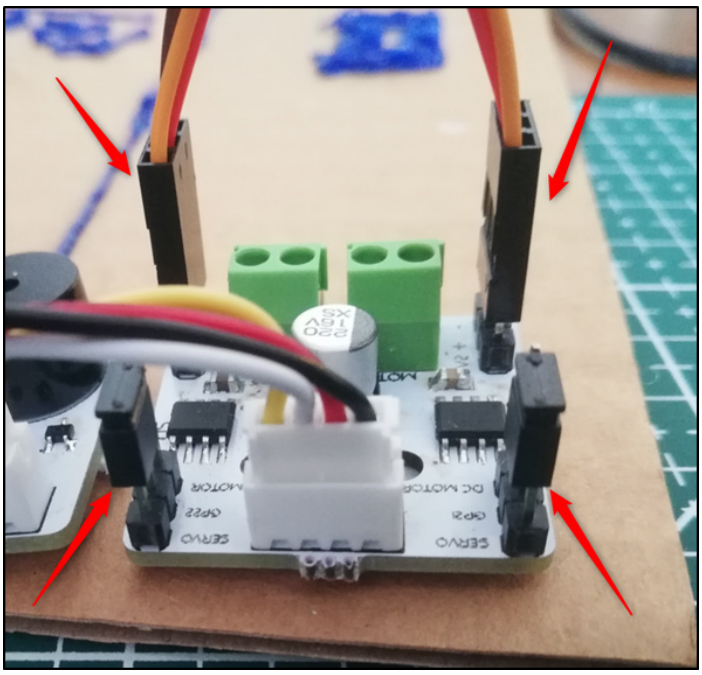
8. Stick the holding servo system to part B with silicone.



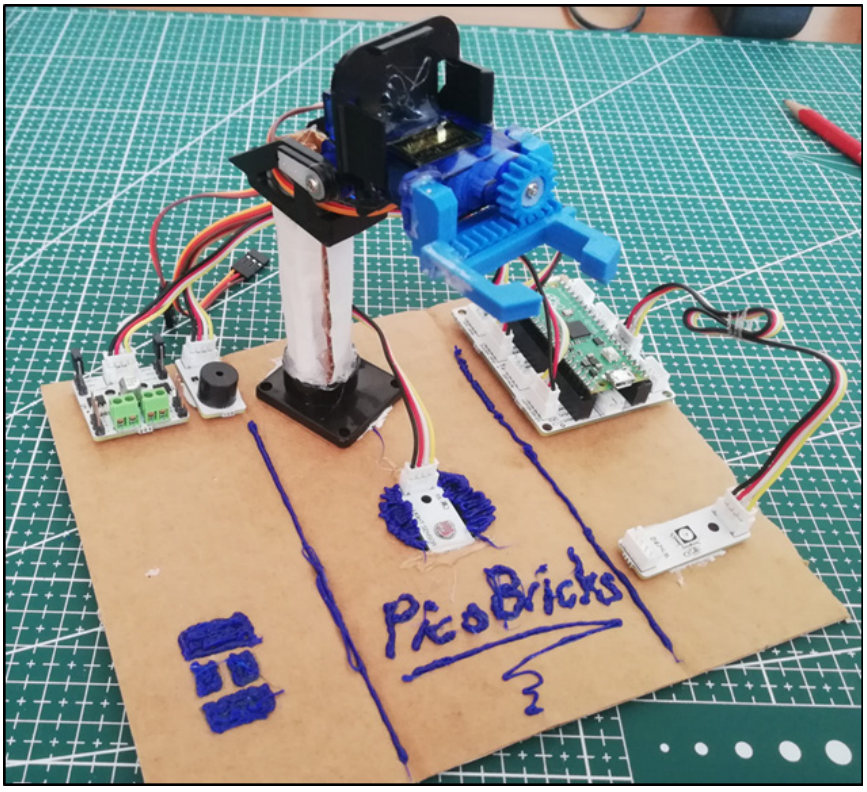
9. Pass the piece we prepared in step 3 over the cylinder we prepared from cardboard in the first step and fix it with silicone.



10. Put the motor drive jumpers on the Servo pins. Connect the cable of the holding servo to the GPIO21 and the cable of the tilting servo to the GPIO22.



11. Place the motor driver, buzzer, LDR and RGB LED module on a platform and place the robot arm on the platform accordingly. With the 3D Pen printer, you can customize your project as you wish.

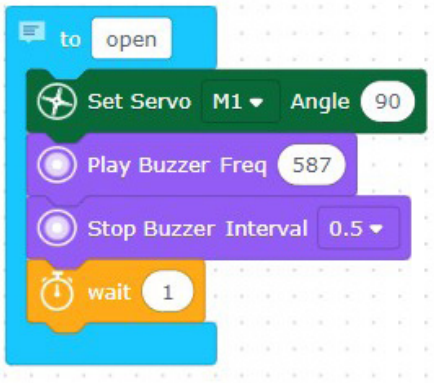


12. You can operate the Robot arm if you feed Picobricks with USB or 3 pen batteries from the power jack on the Picoboard.

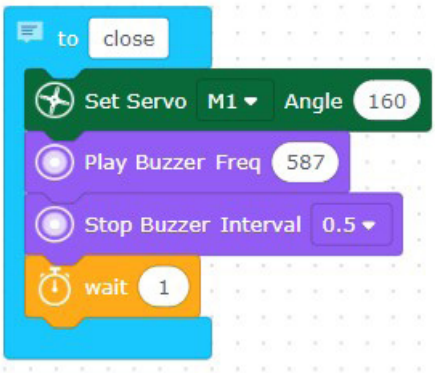
2.17.5. Coding the Project with PicoBricks IDE

Firstly, let's start the project by creating the necessary functions for the code.

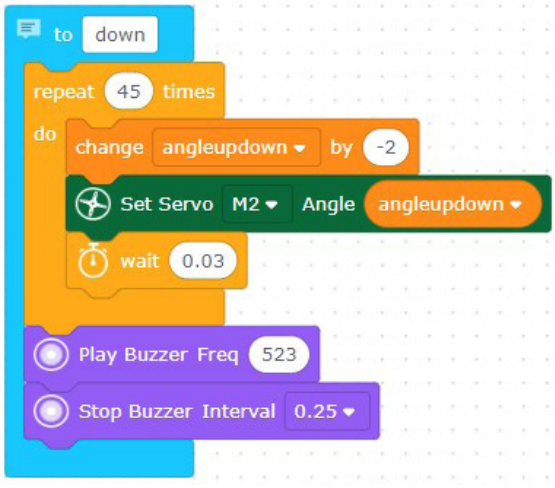
- Let's define the code blocks that, set the position of the servo motor to open the grippers of the robot arm into the "open" function.



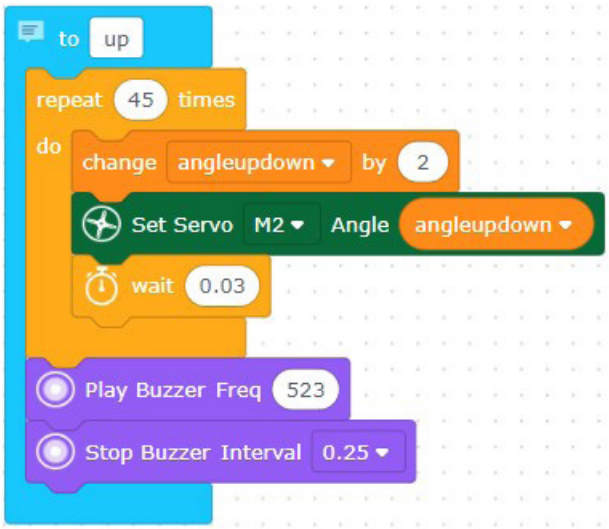
- Let's define the code blocks that, set the position of the servo motor to close the grippers of the robot arm into the "close" function.



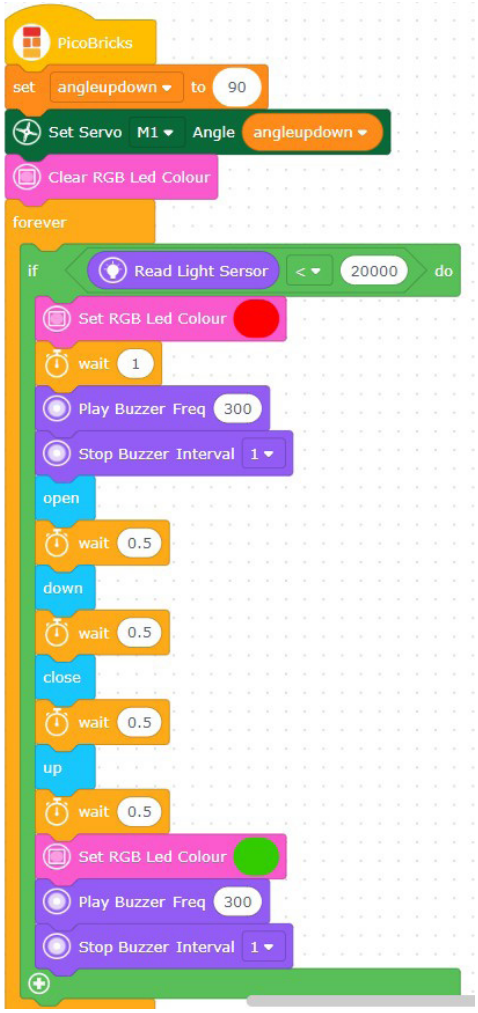
- Let's define the code blocks that, set the position of the servo motor to lean robot arm down into the "down" function.



- Let's define the code blocks that, set the position of the servo motor to set the position of the robot arm is up into the "up" function.



Let's call the functions that we created by using the loop and condition structures. Create the code blocks that, call "open", "down", "close" and "up" functions in 0.5 seconds interval when the LDR sensor detects a value that is less than 20000.



2.18. Smart House

Workplaces, factories, homes and even animal shelters... There are different electronic systems that can be used to protect our living spaces against intruders. These systems are produced and marketed as home and workplace security systems. There are systems where the images produced by security cameras are processed and interpreted, as well as security systems that detect the human body and its movements with sensors and take action. Security systems are set up like a kind of alarm clock and give audible and visual warnings when an unidentified activity is detected in the specified time zone. It notifies the business or the home owner, and it can also make automatic notifications to the security units.

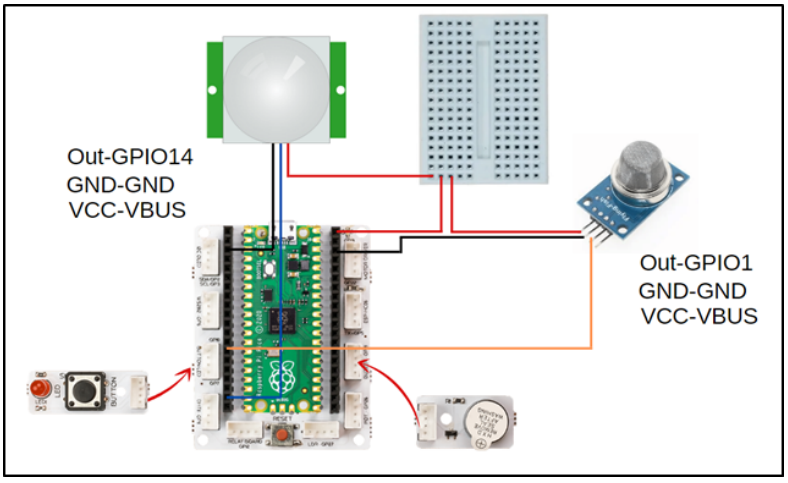
Gas leakage, fire etc. in such cases, gas sensors are used in homes and workplaces to prevent poisoning. In a negative situation, people living in the environment are warned by giving a loud alarm.

We will prepare a model smart home project with PicoBricks using the HC-SR501 and MQ-2 gas sensor. This sensor HC-SR501, also known as PIR sensor, detects motion by capturing the changes of infrared waves reflected by the human body.

2.18.1. Project Details and Algorithm

When the HC-SR501 PIR sensor detects motion, it gives digital output for 3 seconds. We will use a Picoboard, buzzer and button LED module in the project. All parts must be in the model. When Picobricks starts, the button must be pressed to activate the alarm system. After pressing the button, we must wait 3 seconds for the hand to be pulled out of the model. At the end of 3 seconds, the red LED lights up and the alarm system is activated. When the alarm system detects a movement, the red LED will start to flash and the buzzer will sound the alarm. To mute it, Picobricks must be restarted.The MQ-2 sensor is always on. When it detects a toxic gas, it will notify you with a buzzer and red LED.

2.18.2. Wiring Diagram

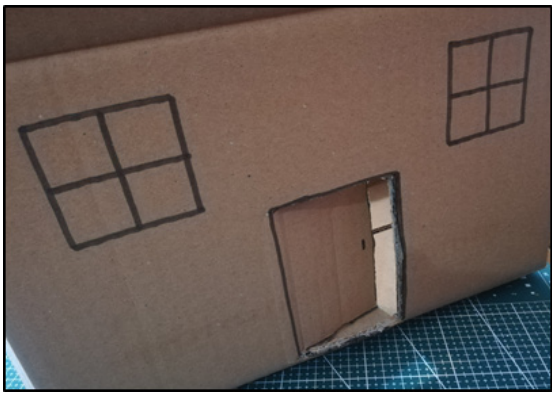


2.18.3. Project Proposal

After making the 25th project, the smart greenhouse, by adding the ESP8266 mode to the burglar alarm project, you can send a notification to the home owner's phone when a thief enters to the home, and turn the project into an IOT project. You can install fire extinguishing pipes on the ceiling of the house with a submersible pump, so that when there is a fire in the house, you can automatically extinguish it.

2.18.4. Construction Stages of the Project

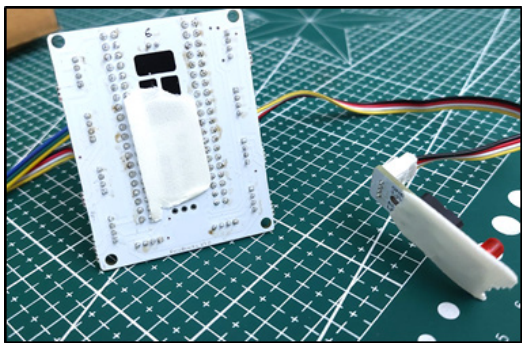
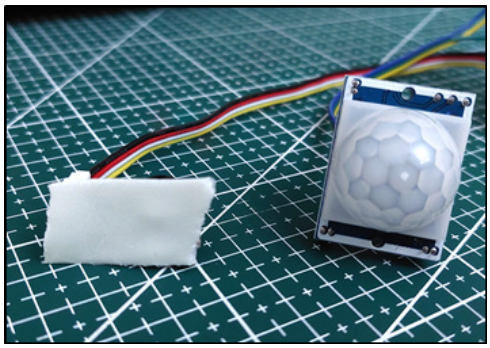
To run the project, you have to turn a cardboard box into a model house. You will need scissors, pencils, tape, glue, and a utility knife. Draw windows and doors on the box with a pencil. Cut the door section with a utility knife.

You can use another cardboard to make the roof part.

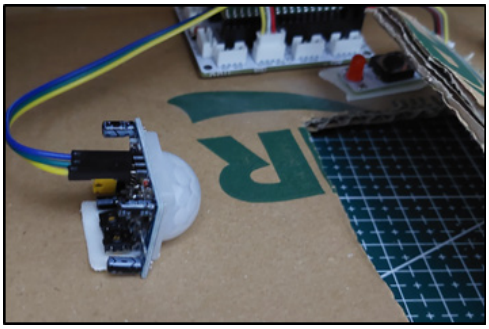
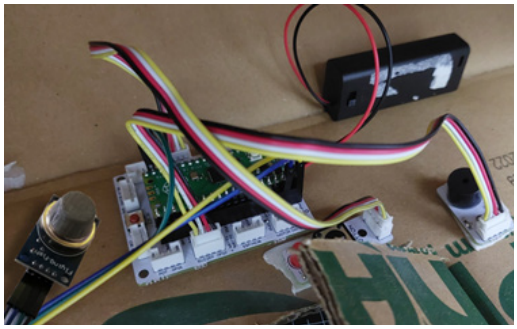



Stick double-sided foam tape under the picobricks pieces.

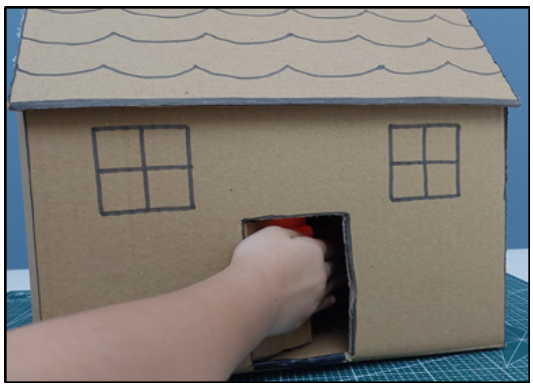
Place pieces of Picobricks inside the model house. Position the PIR sensor to see the

door directly from the inside. Stick the button module just above the door from the inside.

When you connect the battery case to Picoboard and open it, the codes will start to run. 3 seconds after pressing the button, the alarm system will be activated and the red LED will turn on. As soon as you put your hand in the door, the buzzer will start to sound.



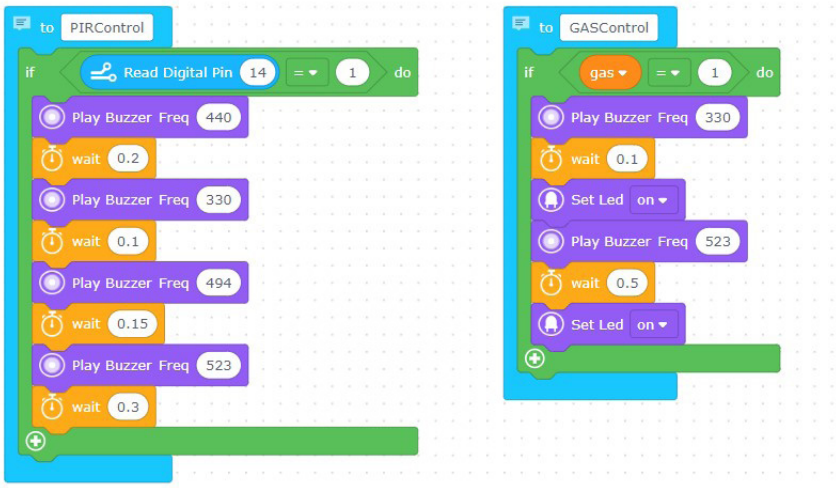



When you hold the lighter gas inside the house, the alarm system is expected to be activated again.

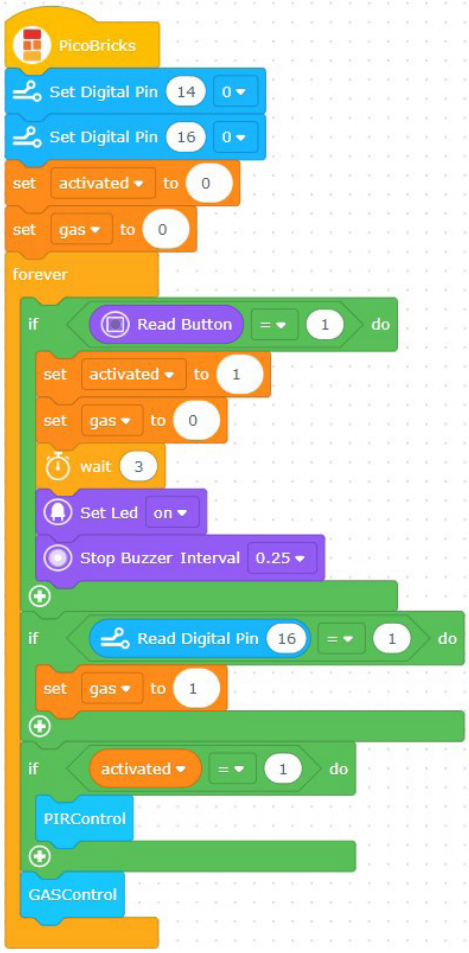
2.18.5. Coding the Project with PicoBricks IDE

Firstly, let's start the project by creating the necessary functions for the code.

- Let's create two functions named as "PIRControl" and "GASControl" that give the warning by controlling the values from the PIR sensor and the GAS sensor.



- Call these functions by using the condition and loop structures. Connect PIR sensor to GPIO 14 and GAS sensor to GPIO 16 digital pins. Create the code blocks call the "GASControl" and "PIRControl" functions, if the values from Gas and PIR sensors are "1".



2.19. Piggy Bank

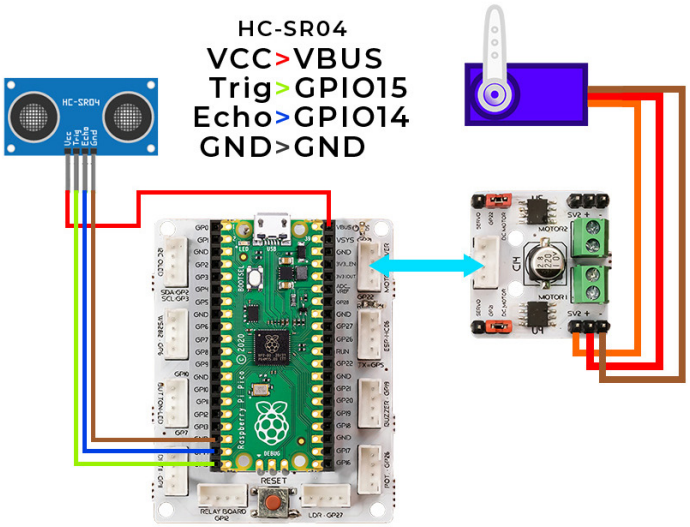
Ultrasonic sensors are sensors that show electrical change by being affected by sound waves. These sensors send sound waves at a frequency that our ears cannot detect and produce distance information by calculating the return time of the reflected sound waves. We, the programmers, develop projects by making sense of the measured distance and the changes in distance. Parking sensors in the front and back of the cars are the places where ultrasonic sensors are most common in daily life. Do you know the creature that finds its way in nature with this method? Because bats are blind, they find their way through the reflections of the sounds they make. [Ses dalgalarının gösterildiği bir görsel olabilir]

Many of us like to save money. It is a very nice feeling that the money we save little by little is useful when needed. In this project, you will make yourself a very enjoyable and cute piggy bank. You will use the servo motor and ultrasonic distance sensor while making the piggy bank.

2.19.1. Project Details and Algorithm

HC-SR04 ultrasonic distance sensor and SG90 servo motor will be used in this project. When the user leaves money in the hopper of the piggy bank, the distance sensor will detect the proximity and send it to the Picobricks. According to this information, Picobricks will operate a servo motor and raise the arm, throw the money into the piggy bank and the arm will go down again.

2.19.2. Wiring Diagram



2.19.3. Project Proposal

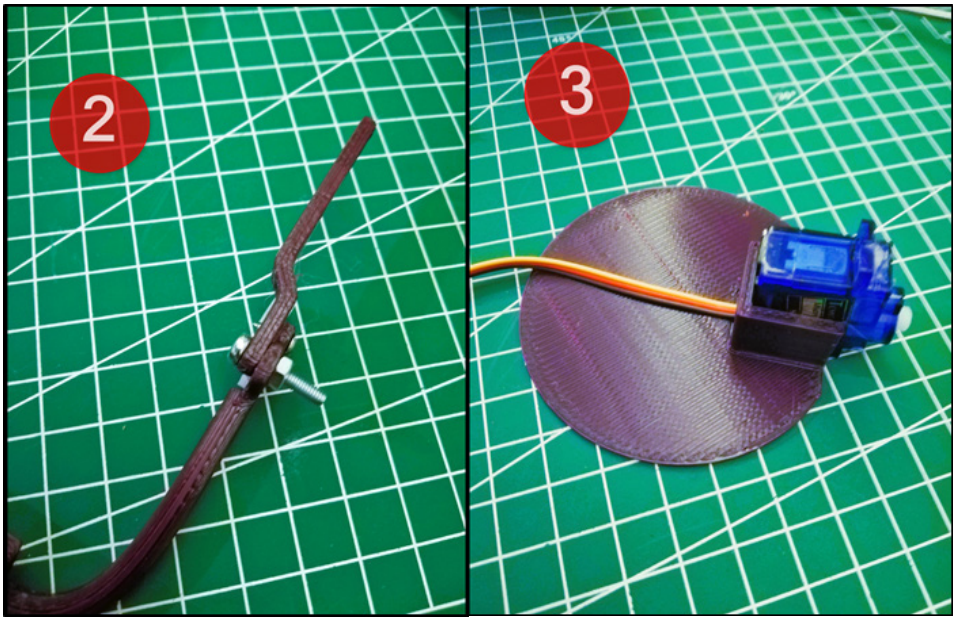
By adding an RGB LED module to the glutton piggy bank project, you can make the light turn on in the color you want every time a coin is thrown, you can add a buzzer and make a sound every time a coin is thrown. You can also print the number of coin flips on the screen by adding an OLED screen.

2.19.4. Construsction Stages of the Project

You can access the original files and construction stages of the project by clicking here. Unlike the project in this link, we will use the HC-SR04 ultrasonic distance sensor. You can download the updated 3D drawing files according to the HC-SR04 ultrasonic distance sensor from this link and get 3D printing.

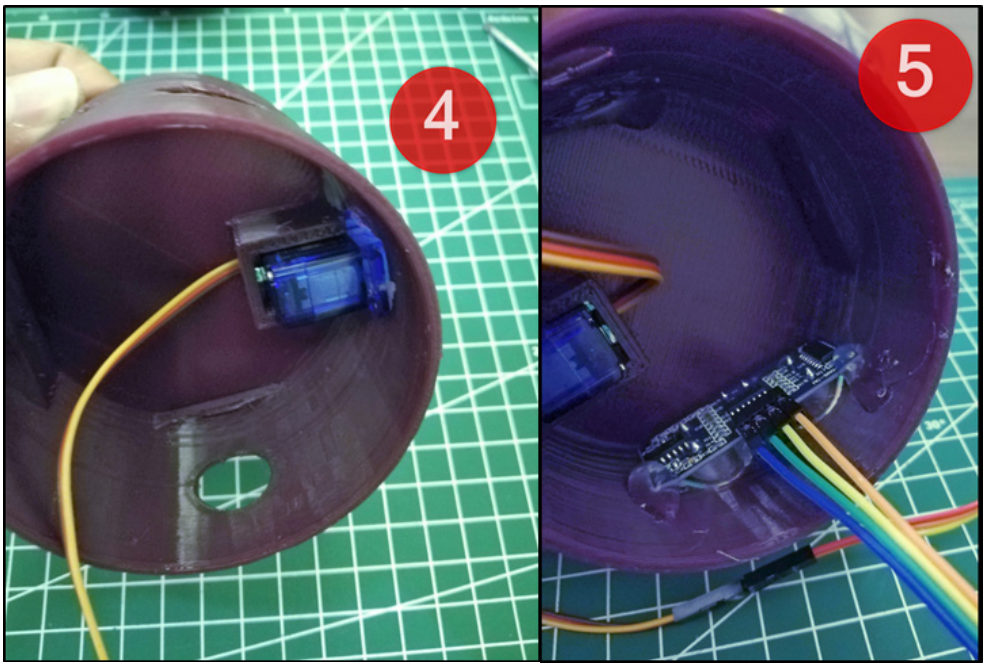


1: Fix the plastic apparatus of the servo motor to the piggy bank arm with 2 screws.



2: Fix the second part of the piggy bank arm with the M3 screw and nut to the first part where the hopper is.

3: Pass the servo motor cable and place it in its slot.



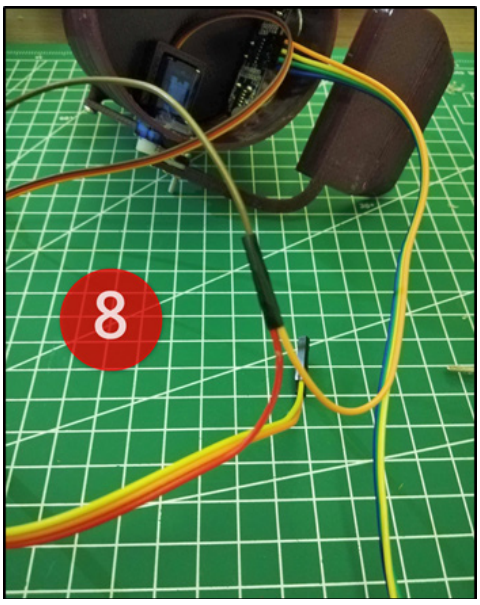
4: Place the servo motor and its housing on the body of the piggy bank. You can use hot glue here.

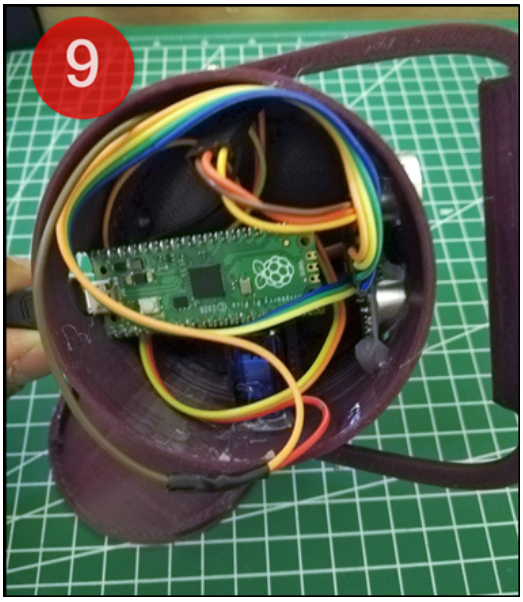
5: Place the ultrasonic distance sensor in the piggy bank body and fix it with hot glue.



6: Attach the piggy bank arm to the servo motor and fix it to the top cover with M3 screws.

7: Fix the piggy bank arm to the body with M2 screw.





8: Plug the cables of the servo motor and ultrasonic distance sensor and connect the power cables.

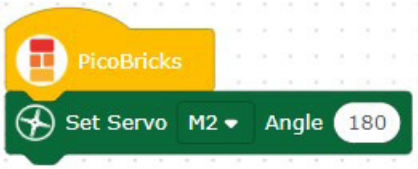
9: According to the circuit diagram, connect the cables of the servo motor and ultrasonic distance sensor to the pico.



10: Plug Pico's USB cable and reassemble the cables and attach the bottom cover. That is all.

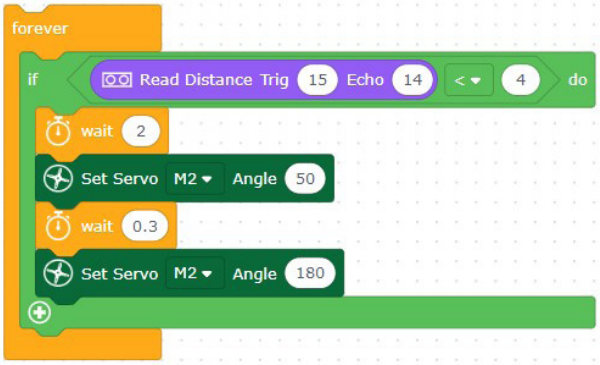
2.19.5. Coding the Project with PicoBricks IDE

Let's start the project by set the position of the servo motor to the beginning position.

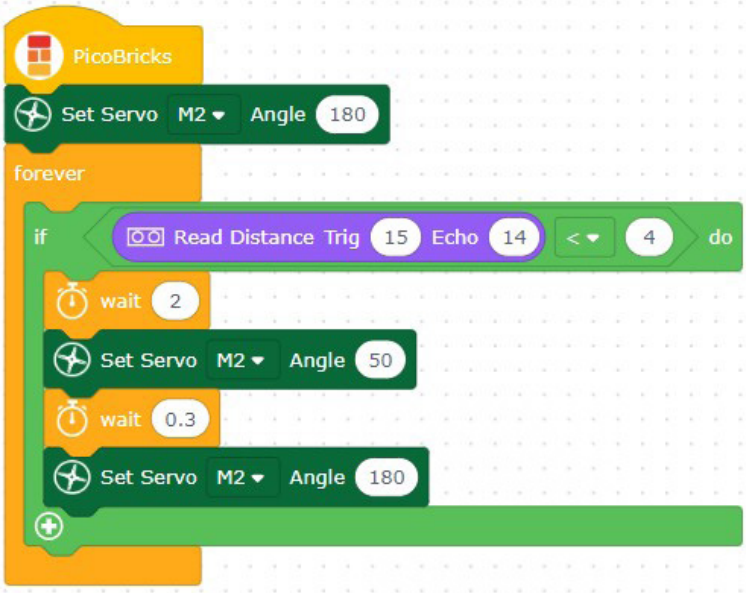


Let's specify the condition structure that run consistently when the project starts. Create the code blocks that,

- wait for two seconds,
- set the angle of the servo motor-2 to 50 degrees and
- set the servo motor to the first position after waiting for 300 milliseconds, if the value from the distance sensor is less than 4.



The code blocks are ready!



2.20. Automatic Trash Bin

The Covid 19 pandemic has changed people's daily routines in many areas. In many areas such as cleaning, working, shopping and social life, people were introduced to a series of new rules that they had to comply with. Covid-19 has LED to the development of new business areas as well as some products to stand out. At a time when hand hygiene was very important, no one wanted to touch the lid of the trash can to throw away their garbage.

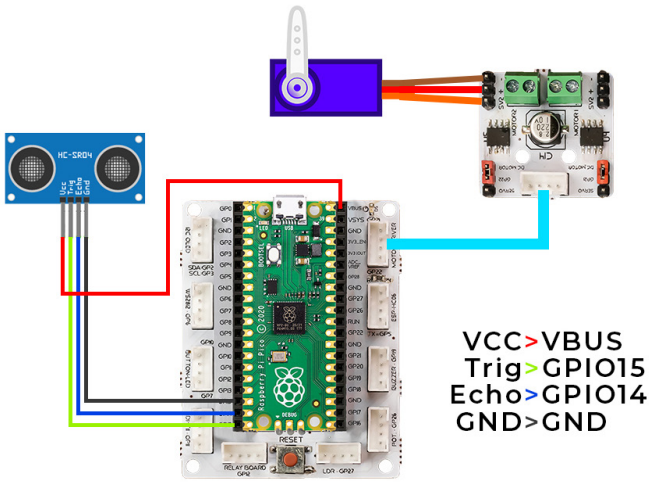
When approached, the lids of which open automatically and when it is full, the trash bins, which make bags ready to be thrown away, found buyers at prices far above their cost. In addition, automatic disinfectant machines provided contactless hygiene by pouring a certain amount of liquid into our palms when we held them under our hands. Automatic disinfectant machines took place on the shelves at prices well above their cost. These two products have similarities in terms of working system. In automatic disinfectant machines, a pump with an electric motor directly transfers the liquid, and some models have devices based on the pumping system with the power of the servo motor. In automatic trash bins, a servo motor that opens the lid was used, and infrared or ultrasonic sensors were used to detect hand movement.

In this project, you will make a mobile and automatic stylish trash bin for your room using an ultrasonic sensor and servo motor with PicoBricks.

2.20.1. Project Details and Algorithm

HC-SR04 ultrasonic distance sensor and SG90 servo motor will be used in this project. When the user puts his hand in front of the lid of the trash can, the distance sensor will detect the proximity and send it to the Picobricks. According to this information, Picobricks will open the lid of the garbage can by running a servo motor and will lower it again after a short while.

2.20.2. Wiring Diagram



2.20.3. Coding the Project with PicoBricks IDE

Let's start the project by setting the servo motor to the beginning position.

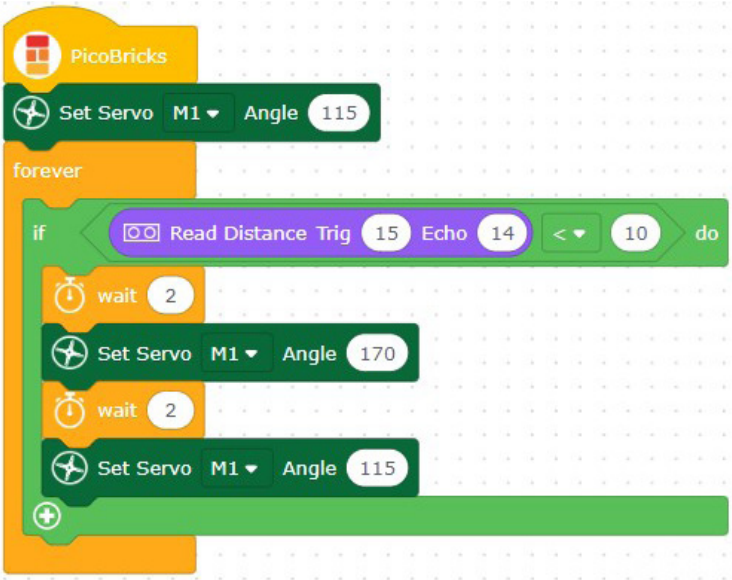


When the project starts, let's determine the condition structure running consistently. Let's create the code blocks that,

- wait for two seconds,
- set the angle of the servo motor-1 to 170 degrees and
- set the servo motor to the first position after waiting for two seconds, if the value from the distance sensor is less than 10.



The code blocks are ready!



2.21. Digital Ruler

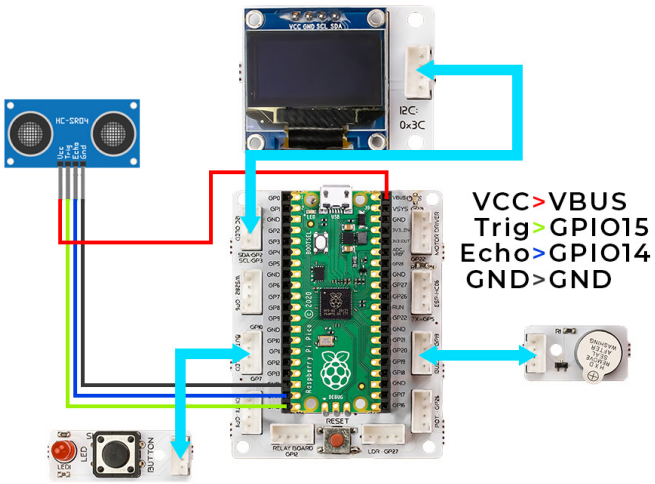
Many tools are used to measure length. At the beginning of these tools are rulers. Our measuring instrument differs according to the place and size to measure. Tape measures are used in construction and architecture, and calipers are used for small objects that require millimeter precision. In addition, if it is desired to measure an area that needs both large and precise measurement, distance meters working with laser and infrared systems are used. Ultrasonography devices used in the health sector also work with the same logic, but convert their measurements into visuals.

In our project, we will use PicoBricks and an ultrasonic sensor to prepare a digital ruler that will display the distance value on the OLED screen when the button is pressed. Ultrasonic sensors detect distance according to the return times of the sound waves they emit.

2.21.1. Project Details and Algorithm

When Picobricks starts, instructions are displayed on the OLED screen. After the user presses the button, 20 measurements are made at 50 millisecond intervals for 1 second and the average is taken. The red LED stays on during the measurement, and the red LED turns off when the measurement is complete. The average value is added to the distance from the tip of the sensor to the back of the box. The last distance value is displayed on the OLED display.

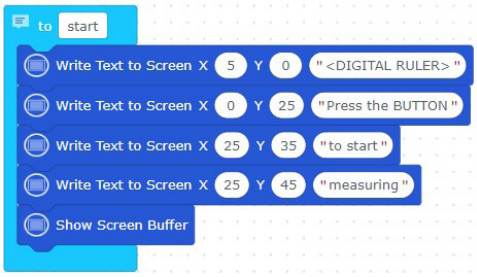
2.21.2. Wiring Diagram



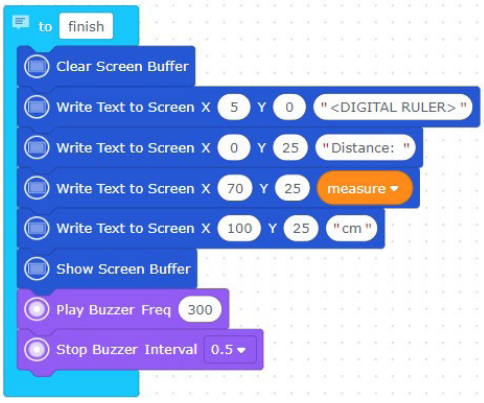
2.21.3. Coding the Project with PicoBricks IDE

Firstly, let's start the project by creating the necessary functions for the code.

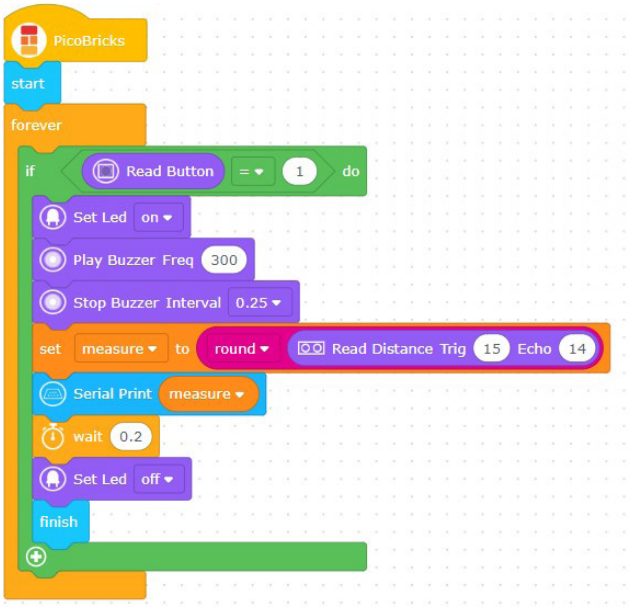
- By defining the function named as "start", let's determine the statements that will be printed on the OLED screen when the project starts.



- Let's determine the operations to be needed after the measurement is made, by defining a function named as "finish".



- Let's call the functions we made by using the loop and condition structures. Make the measurement every time the button is pressed and print the value of the measurement on the OLED screen.



2.22. Air Piano

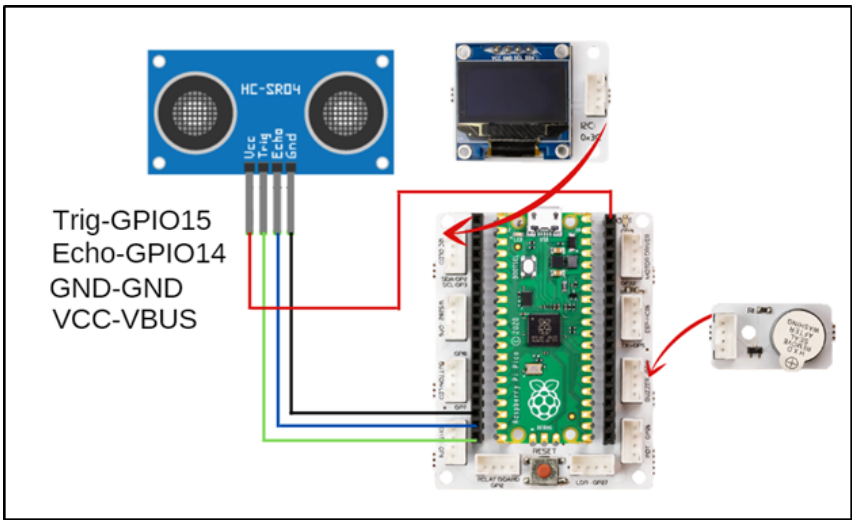
With the development of electronic technology, musical instruments that are difficult to produce, expensive and producing high-quality sound have been digitized. Pianos are one of these instruments. Each key of digital pianos produces electrical signals at a different frequency. Thus, it can play 88 different notes from its speakers. Factors such as the delay time of the keys of digital instruments, the quality of the speaker, the resolution of the sound have appeared as the factors affecting the quality. In electric guitars, vibrations in strings are digitized instead of keys. On the other hand, In wind instruments, the notes played can be converted into electrical signals and recorded thanks to the high-resolution microphones plugged into the sound output. This development in electronic technology has facilitated access to high-cost musical instruments, music education has gained a wider variety and spread to a wider audience.

In this project we will make a simple piano that can play 8 notes with PicoBricks. The speaker of this piano will be the buzzer. The ultrasonic sensor will act as the keys of the piano.

2.22.1. Project Details and Algorithm

In this project, we will make a piano application using the HC-SR04 Ultrasonic distance sensor and the buzzer module on PicoBricks. We will make the buzzer play different notes according to the values coming from the distance sensor, and we will create melodies by moving our hand closer to the sensor and away from it. In addition, we will instantly print the distance played note information on the OLED screen.

2.22.2. Wiring Diagram



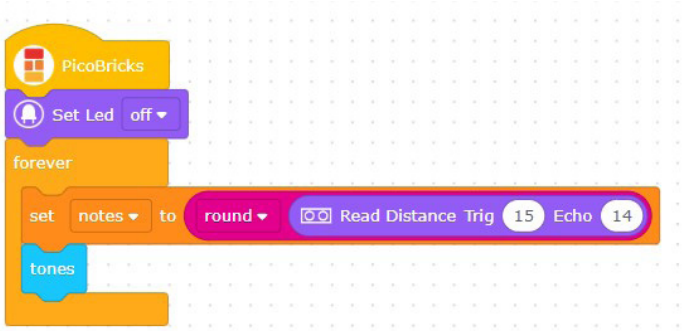
2.22.3. Coding the Project with PicoBricks IDE

Firstly, let's start the project by creating the necessary functions.

- Let's determine the notes to be in the piano project by creating a function named as "tones".



- Let's play the notes according to the distance value with buzzer, by using "Forever" loop block.



2.23. Maze Solver Robot

Coding education is as old as the history of programming languages. Today, different products are used to popularize coding education and make it exciting and fun. The first of these is educational robots. Preparing and coding robots improves children's engineering and coding skills. Robotics competitions are organized by institutions and organizations to popularize coding education and encourage teachers and students. One of these competitions is the Maze Solver Robot competitions. These robots firstly learn the destination by wandering around the maze and return to the starting point. Then, when they start the labyrinth again, they try to reach their destination in the shortest way possible. Robots use distance sensors while learning about the maze. Infrared or ultrasonic sensors are used in these robots.

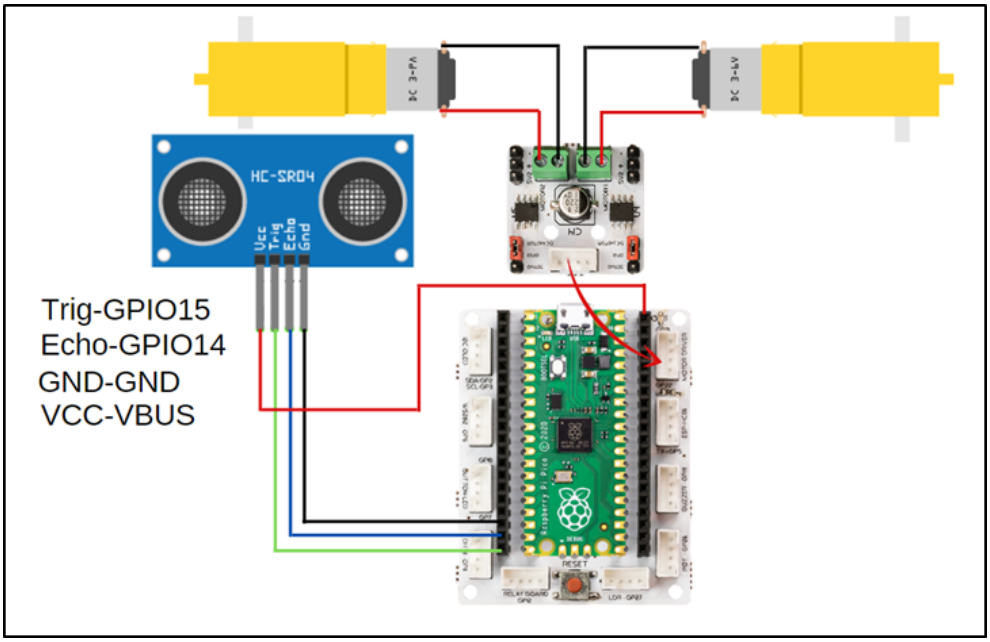
Smart robot vacuums used in homes and workplaces also work with logic close to the algorithms of maze-solver robots. Thanks to their algorithms that constantly check and map the obstacles, they try to do it completely and without crashing. Most of the smart vacuums are equipped with LIDAR and infrared sensors, which make high-precision laser measurements for distance measurement and obstacle detection.

In this project, we will make a simple robot with PicoBricks that you can prepare for maze solver robot competitions.

2.23.1. Project Details and Algorithm

In the maze solving robot project, we will use the 2WD robot car kit that comes out of the set. We will use the HC-SR04 ultrasonic distance sensor so that the robot can detect the distance in front of it and decide its movements on its own. In the maze, the robot will detect the distance in front of the car and move forward if it is empty. If the distance is less than 5 cm, the car will turn right, measure the distance again, if the distance on the right is greater than 5 cm, it will continue on its way, if it is less, it will turn left and move forward. In this way, by turning right and left, we will enable the vehicle to move forward and exit the maze through the empty roads in the maze.

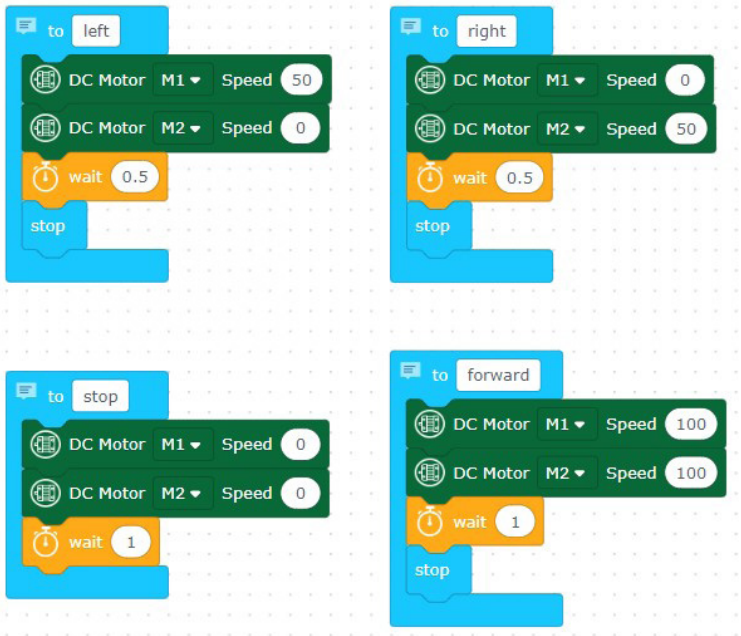
2.23.2. Wiring Diagram



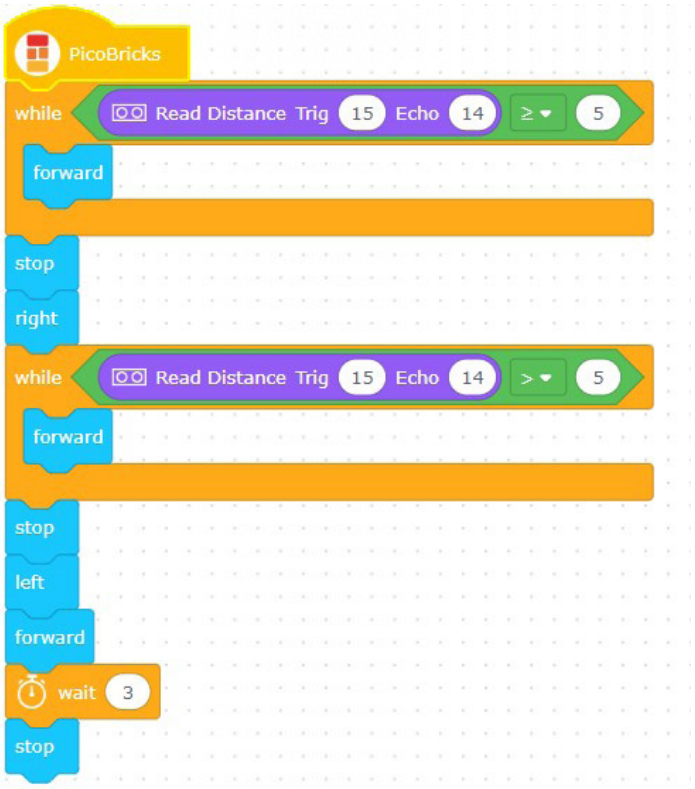
2.23.3. Coding the Project with PicoBricks IDE

Firstly, let's start the project by creating the necessary functions.

- Let's determine the movements of the robot by defining four functions named as "left", "right", "forward" and "stop".



- Let's run the functions we have determined in the loop and condition structures determined according to the path of the labyrinth.



2.24. Smart Greenhouse

The rapid changes in climate due to the effect of global warming cause a decrease in productivity in agricultural activities. In the 1500s, Daniel Barbaro built the first known greenhouse in history. Greenhouses are suitable environments for growing plants that can provide controllable air, water, heat and light conditions. In greenhouses, heaters are used to balance the heat, electric water motors for irrigation, fans are used to regulate humidity and to provide pollination. With the development of technology, the producer can follow the status of the greenhouse with his phone from anywhere and can do the work that needs to be done. The general name of this technology is Internet of Things (IOT).

Special sensors are used to measure temperature, humidity and oxygen content in greenhouses. In addition, special sensors measuring soil moisture are used to decide on irrigation. Electronically controlled drip irrigation systems are used to increase irrigation efficiency.

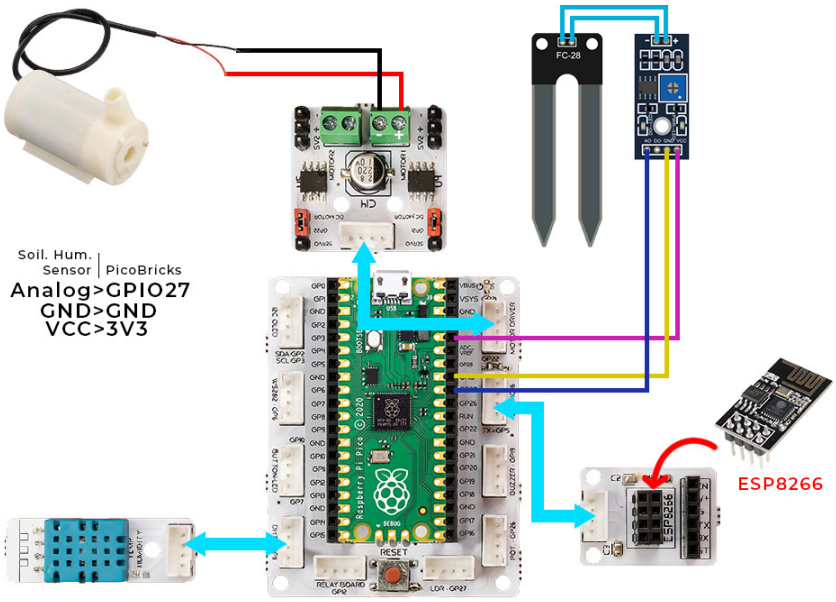
In this project, we will prepare a simple greenhouse with IOT technology and PicoBricks. We will use PicoBricks with the ESP8266 wifi module in this greenhouse. In this way, we will turn the greenhouse into an object that we can track over the Internet.

2.25.1. Project Details and Algorithm

The greenhouse model you will prepare will include a soil moisture sensor, and a DHT11 temperature and humidity sensor hanging from the top. A submersible pump will be placed in the water tank outside the model, and the hose coming out of the end of the pump will go to the ground in the greenhouse. Picoboard will be placed in a suitable place outside the greenhouse model.

When Picobricks starts, it starts to broadcast wifi thanks to the ESP8266 wifi module. When we enter the IP address of Esp8266 from the smart phone connected to the same network, we encounter the web page where we will control the Greenhouse. Here we can see the temperature and humidity values. If we wish, we can start the irrigation process by giving the irrigation command.

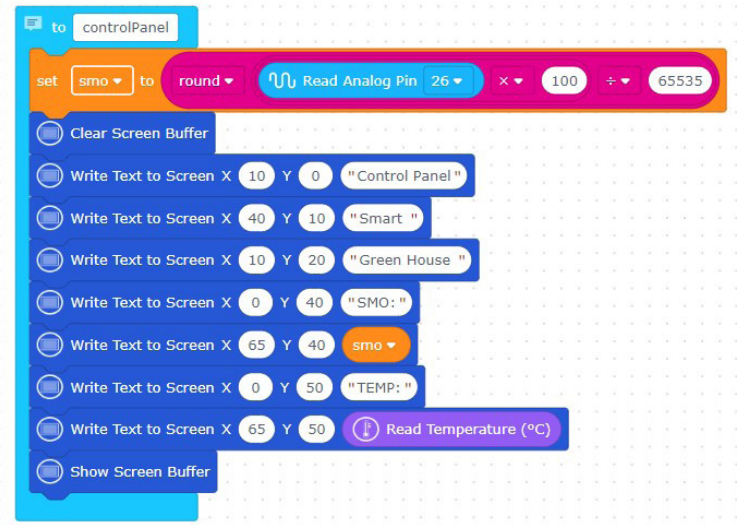
2.24.2. Wiring Diagram



2.24.3. Coding the Project with PicoBricks IDE

Firstly, let's start the project by creating the necessary functions.

- Let's create the function named as "controlPanel" that print the values of the PicoBricks modules used in the project.



■ Let's continue creating the project by using the Wi-Fi blocks in PicoBricks IDE. Enter the ID and password of your Wi-Fi network to communicate Raspberry Pi Pico over the Wi-Fi network.



■ By using the Wi-Fi blocks, continue the communication process over the IP address. In this step, let's paste the HTML code into the HTML block in PicoBricks IDE by creating a HTML page.

HTML Code:

```
<!DOCTYPE html>
  <html>
    <head>
      <meta name="viewport" content="width=device-width, initial-scale=1">
      <link rel="icon" href="data:,">
      <title>Smart Green House</title>
    </head>
    <style>
      body{margin:15%;}
      h1{color:#7EA03E}
      .buttonw{width:300px; height:200px; background-color:#D89008; font-size:50px; color:white; border-radius:10px;cursor: pointer;}
    </style>
    <body>
      <center>
        <h1>Smart Green House Watering Panel</h1>
        <form><button class="buttonw" name="watering" value="on" type="submit">Watering</button></form>
        </center>
    </body>
  </html>
```



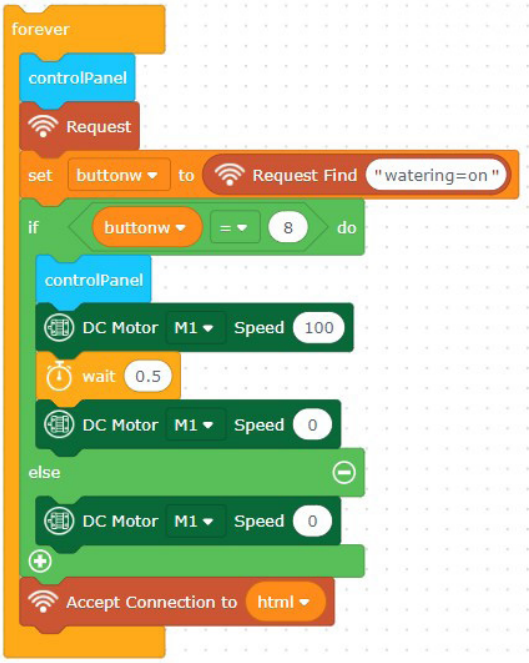
Paste the given HTML code in this area.

Smart Green House Watering Panel

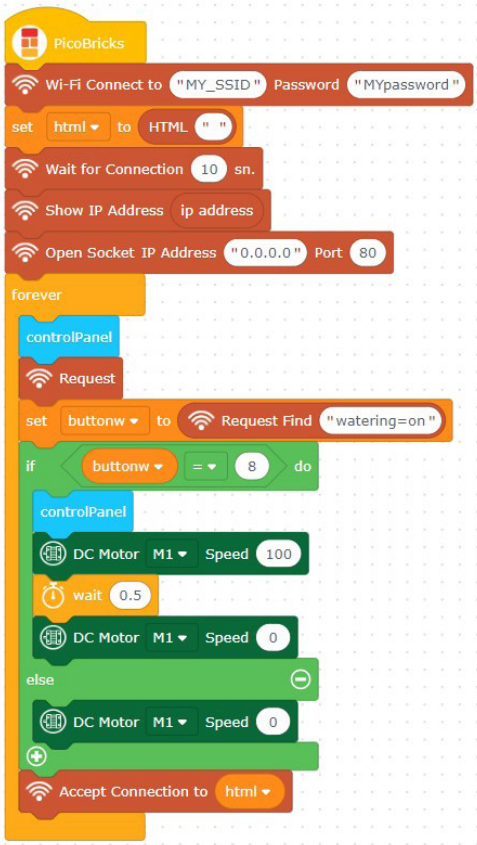
Watering

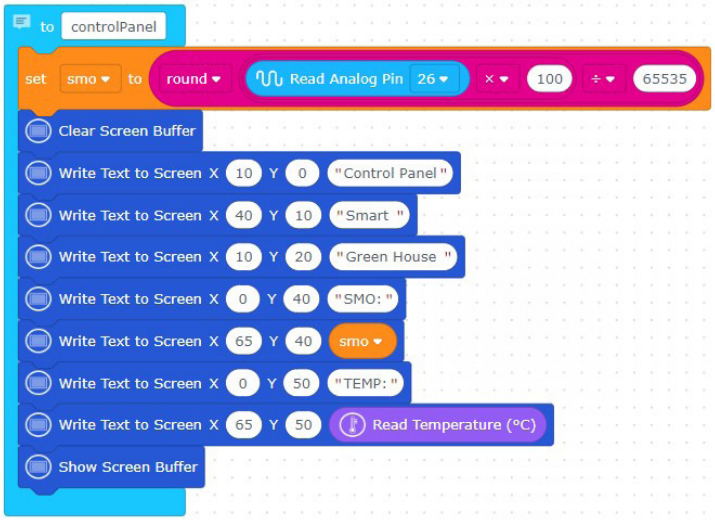
By clicking the button, you can irrigate the greenhouse.

■ Let's complete the project by doing the necessary processes in the condition and loop structures for the project. In this step, let's create the code blocks that provides to irrigate the greenhouse by clicking the button in the WEB page.



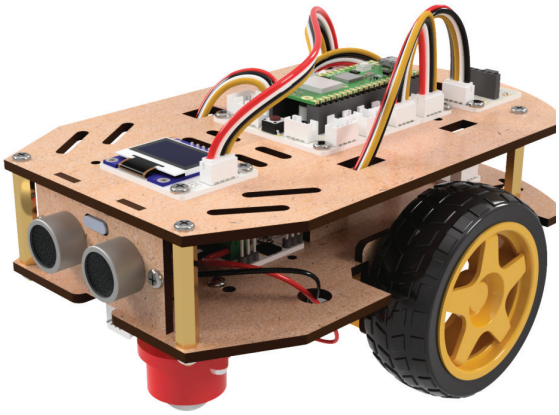
The projects is completed!

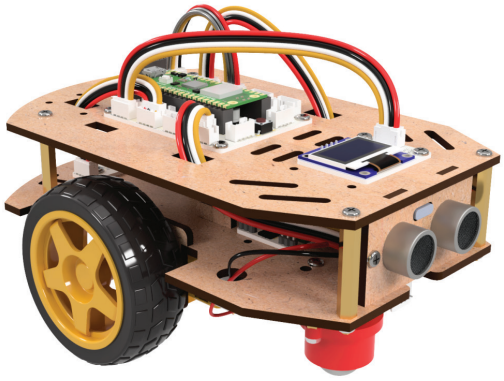




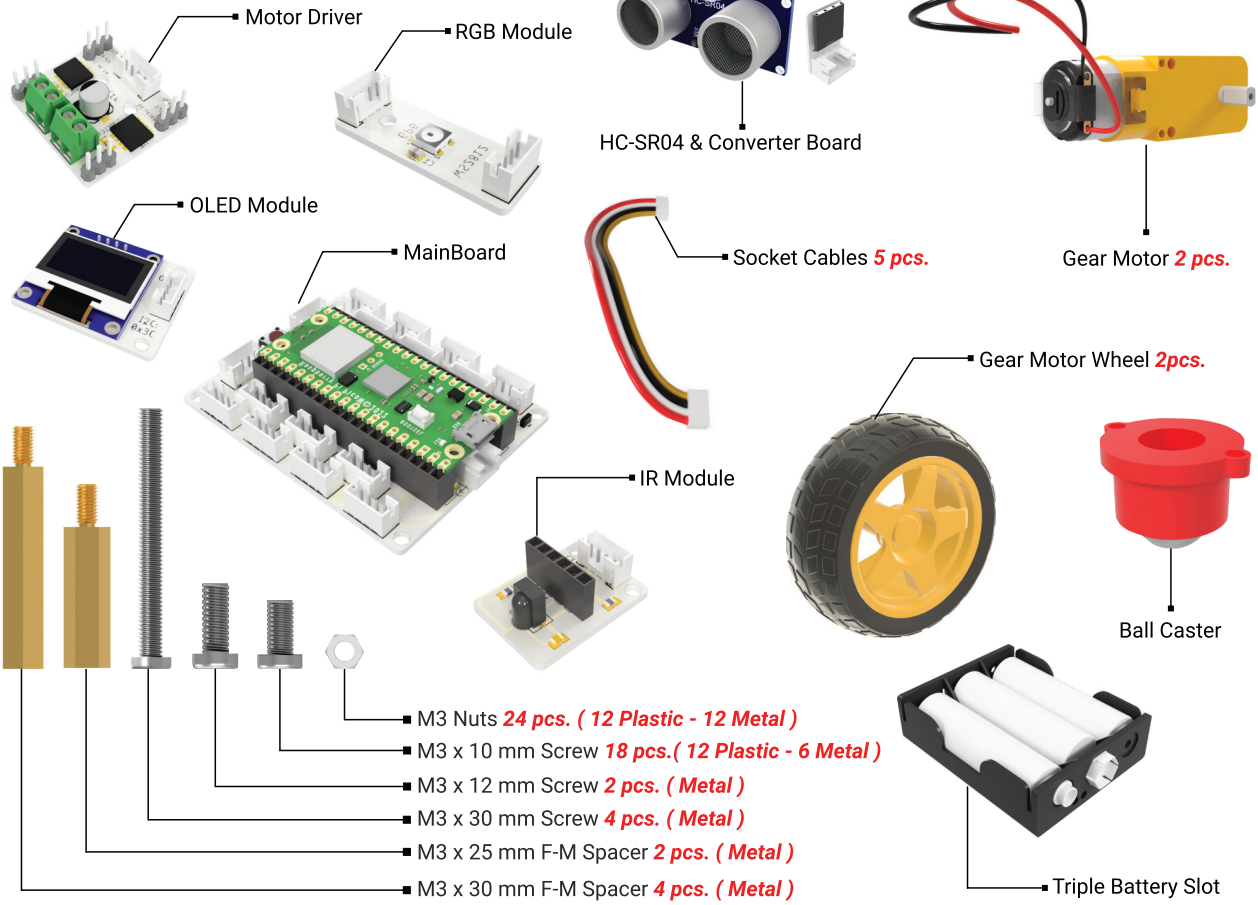
3. RESOURCES

PicoBricks Robot Car Setup Guide

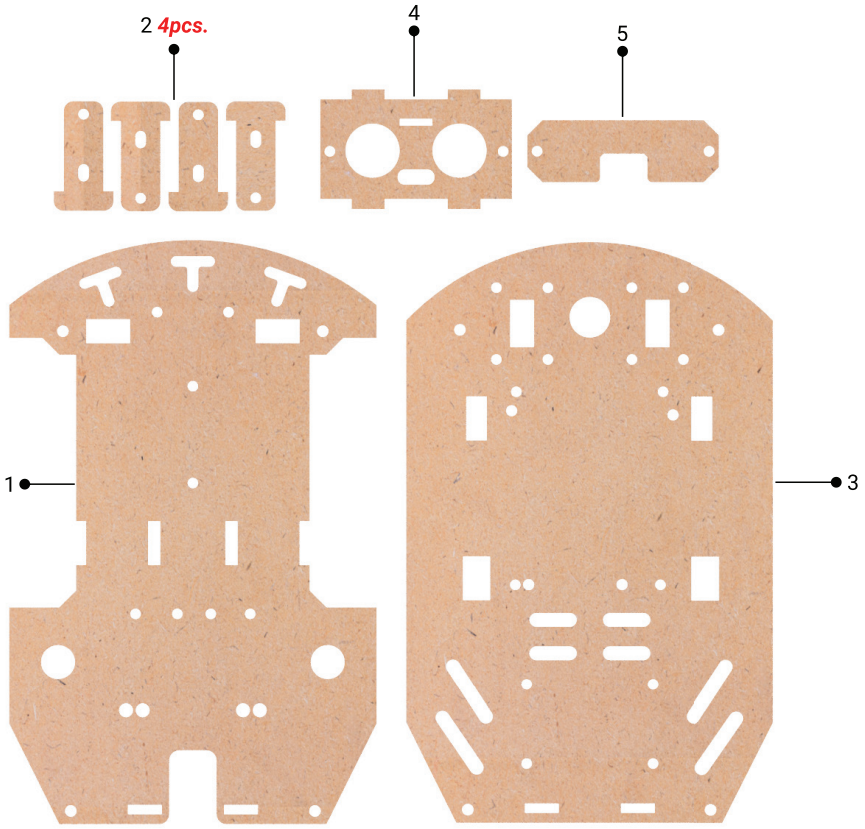


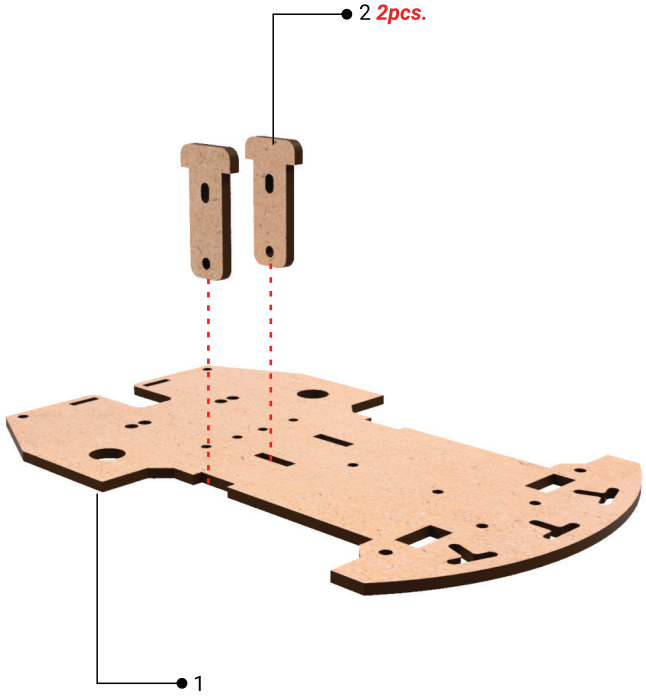


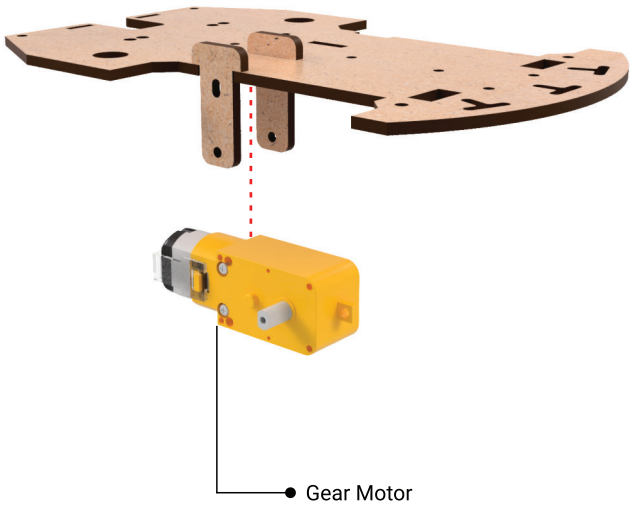
Material List

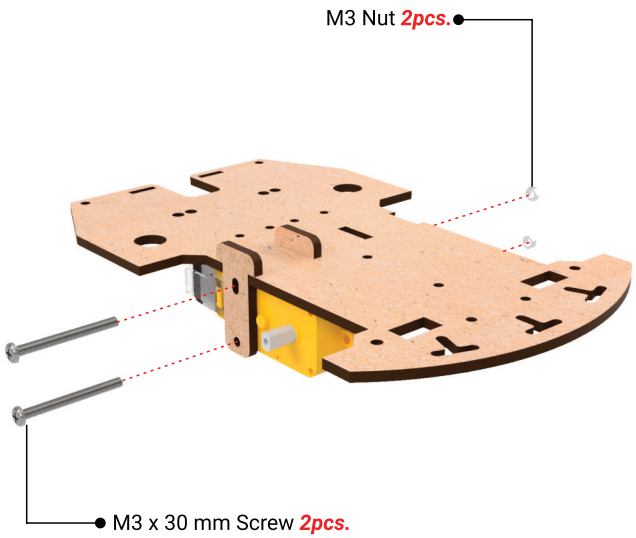


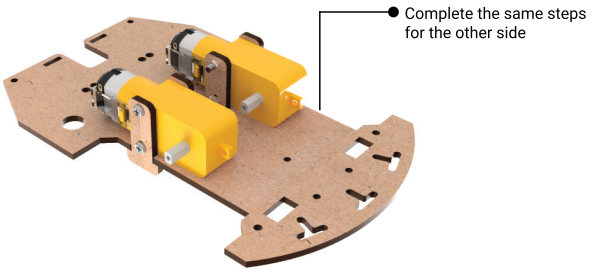
Wooden Parts

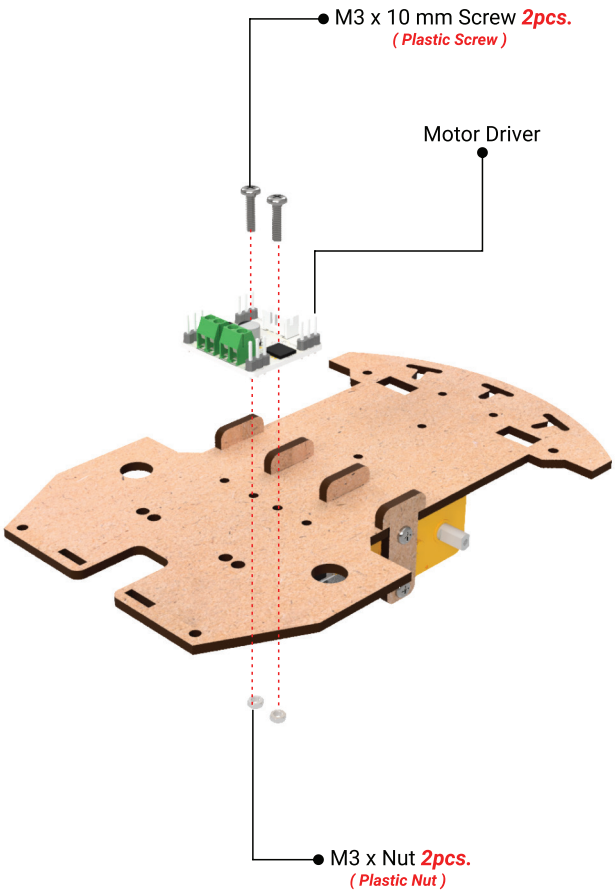


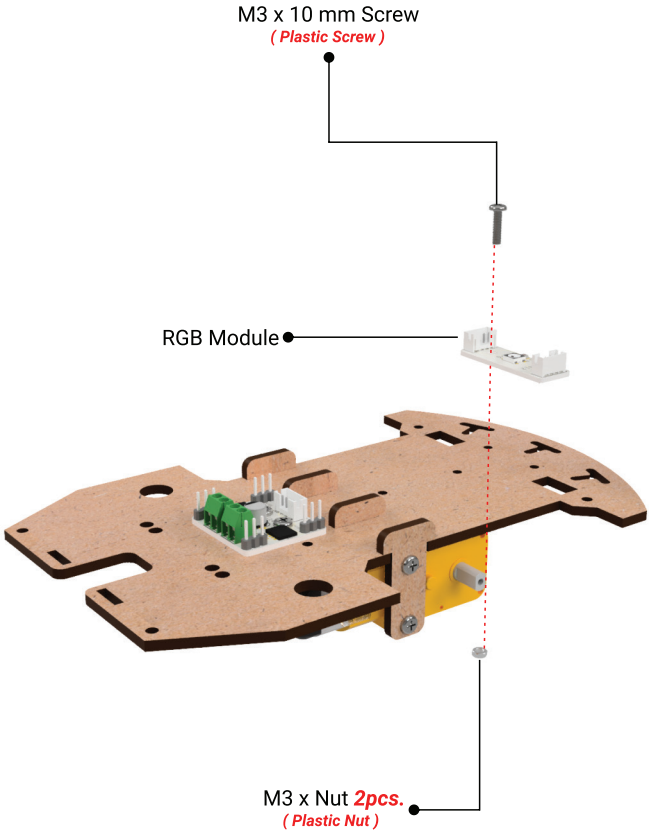


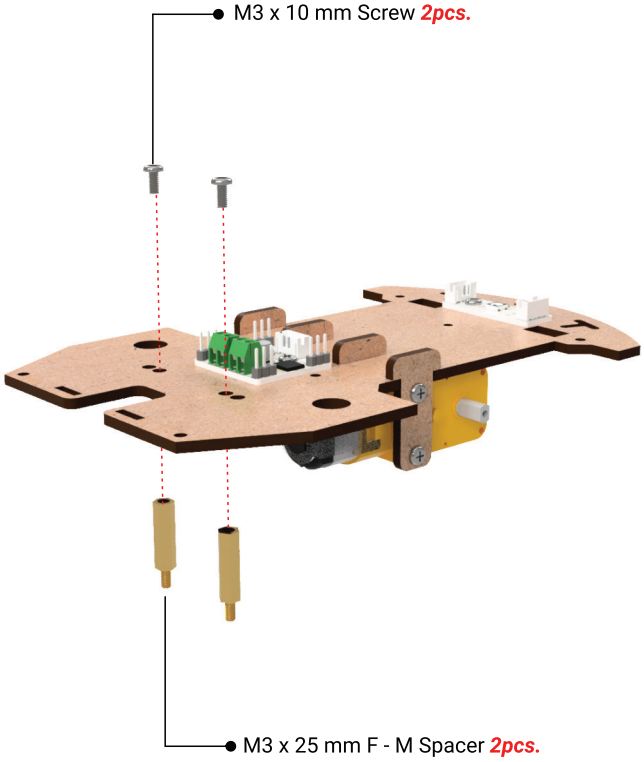


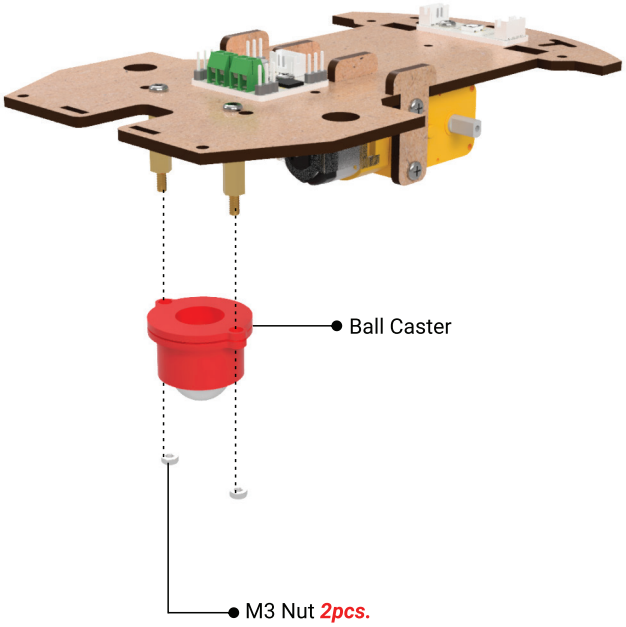

M3 Nut 2pcs.
M3 x 30 mm Screw 2pcs.

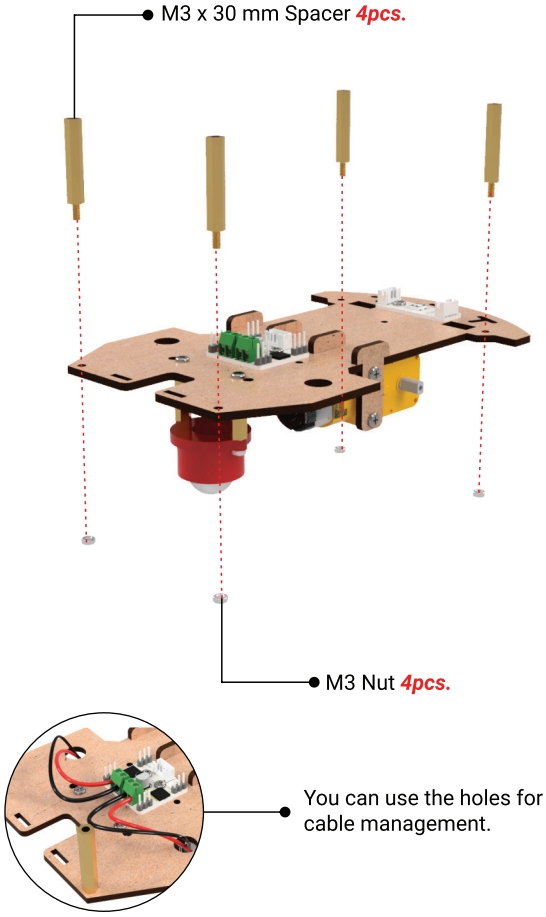

Complete the same steps
for the other side

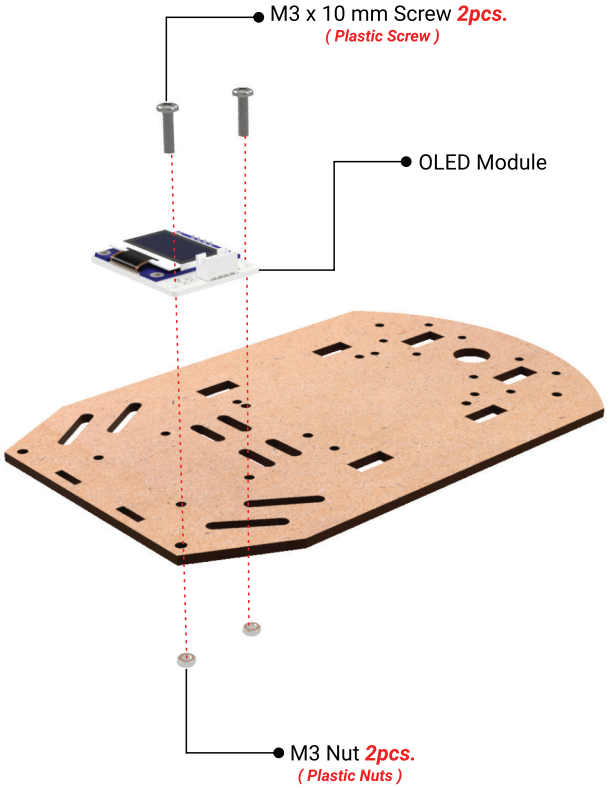

M3 x 10 mm Screw 2pcs.
(Plastic Screw)
Motor Driver
M3 x Nut 2pcs.
(Plastic Nut)

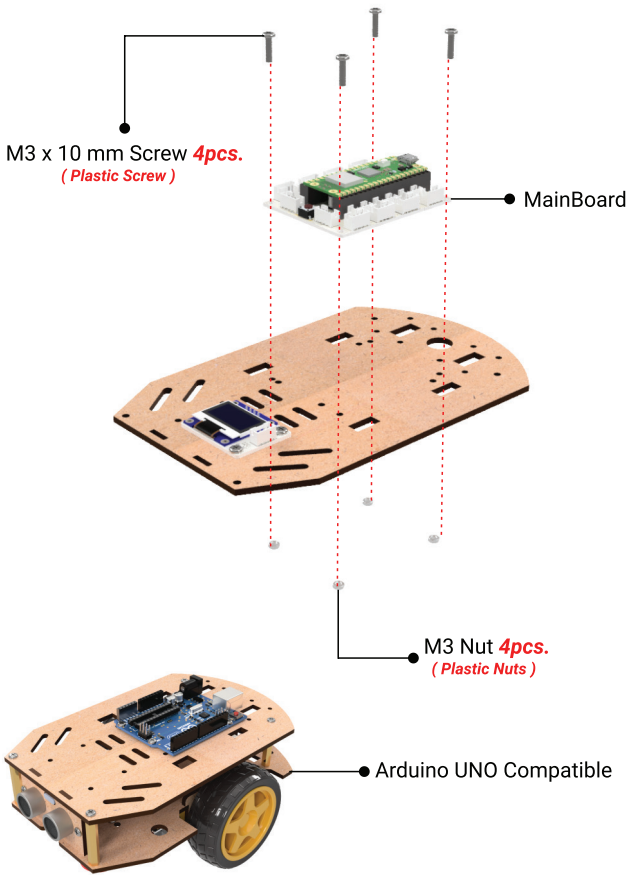

M3 x 10 mm Screw
(Plastic Screw)
RGB Module
M3 x Nut 2pcs.
(Plastic Nut)

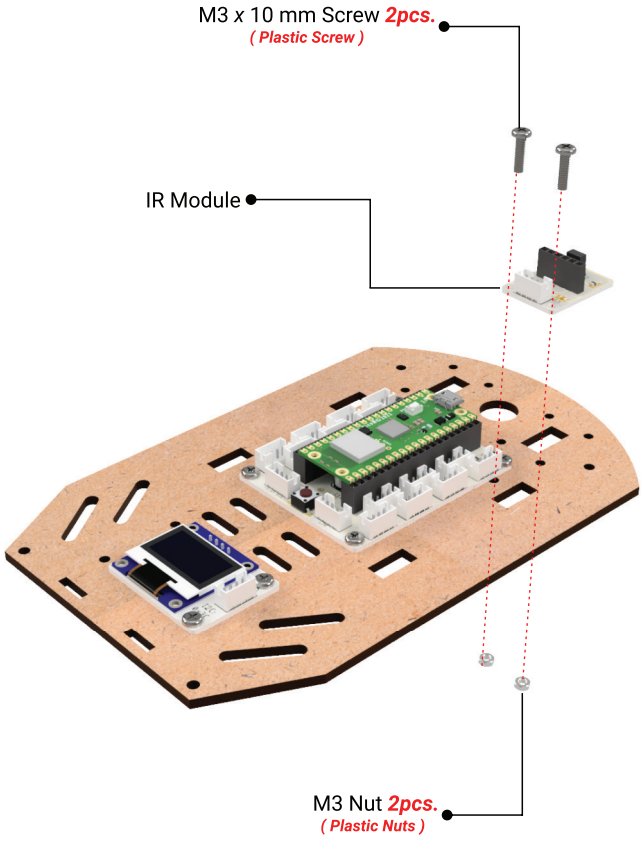

M3 x 10 mm Screw 2pcs.
M3 x 25 mm F - M Spacer 2pcs.

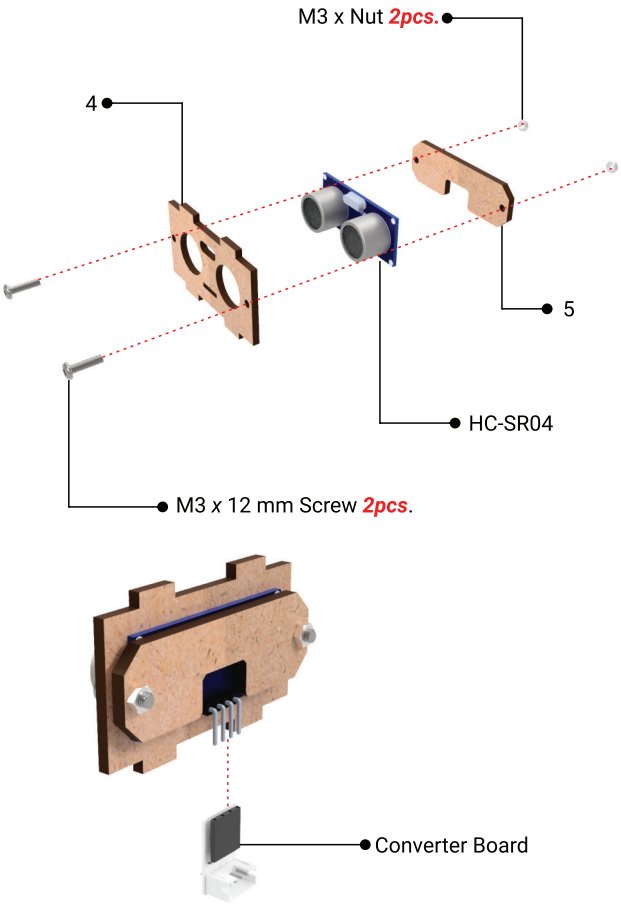

Ball Caster
M3 Nut 2pcs.

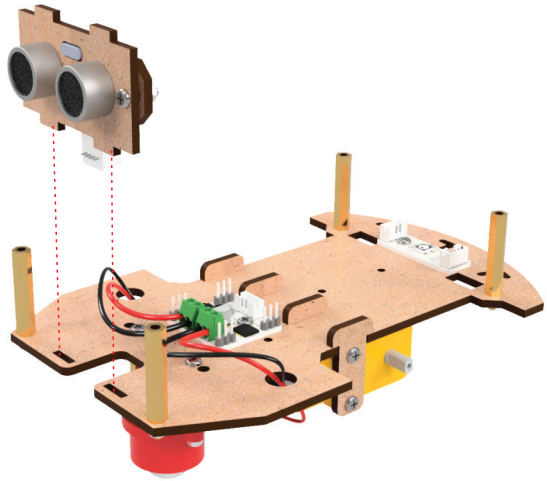

M3 x 30 mm Spacer 4pcs.
M3 Nut 4pcs.
You can use the holes for cable management.

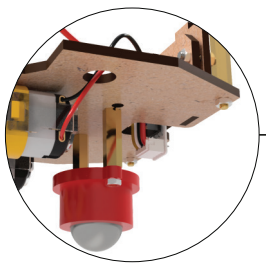

M3 x 10 mm Screw 2pcs.
(Plastic Screw)
OLED Module
M3 Nut 2pcs.
(Plastic Nuts)

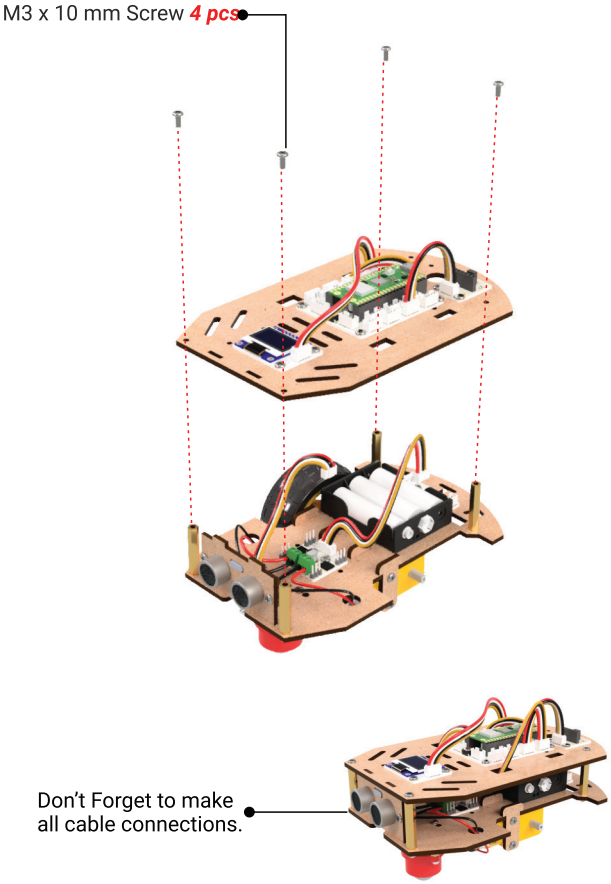

M3 x 10 mm Screw 4pcs.
(Plastic Screw)
MainBoard
M3 Nut 4pcs.
(Plastic Nuts)
Arduino UNO Compatible

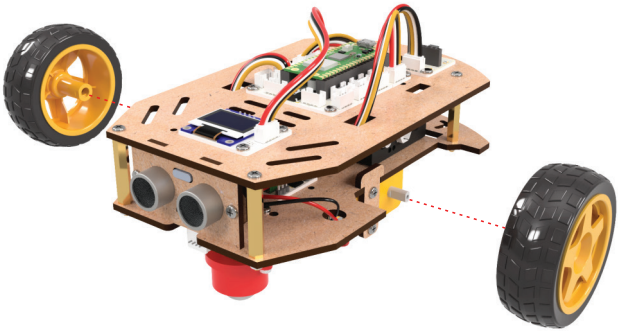

M3 x 10 mm Screw 2pcs.
(Plastic Screw)
IR Module
M3 Nut 2pcs.
(Plastic Nuts)

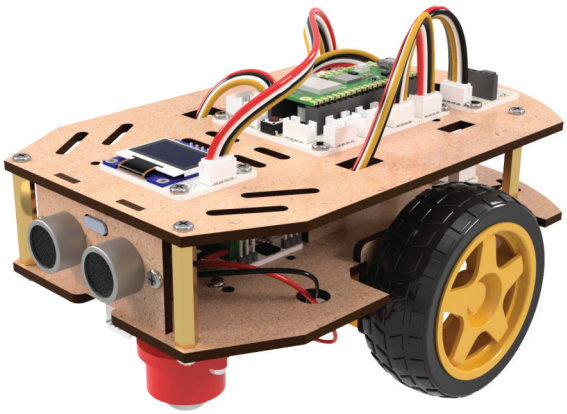

M3 x Nut 2pcs.
4
5
HC-SR04
M3 x 12 mm Screw 2pcs.
Converter Board

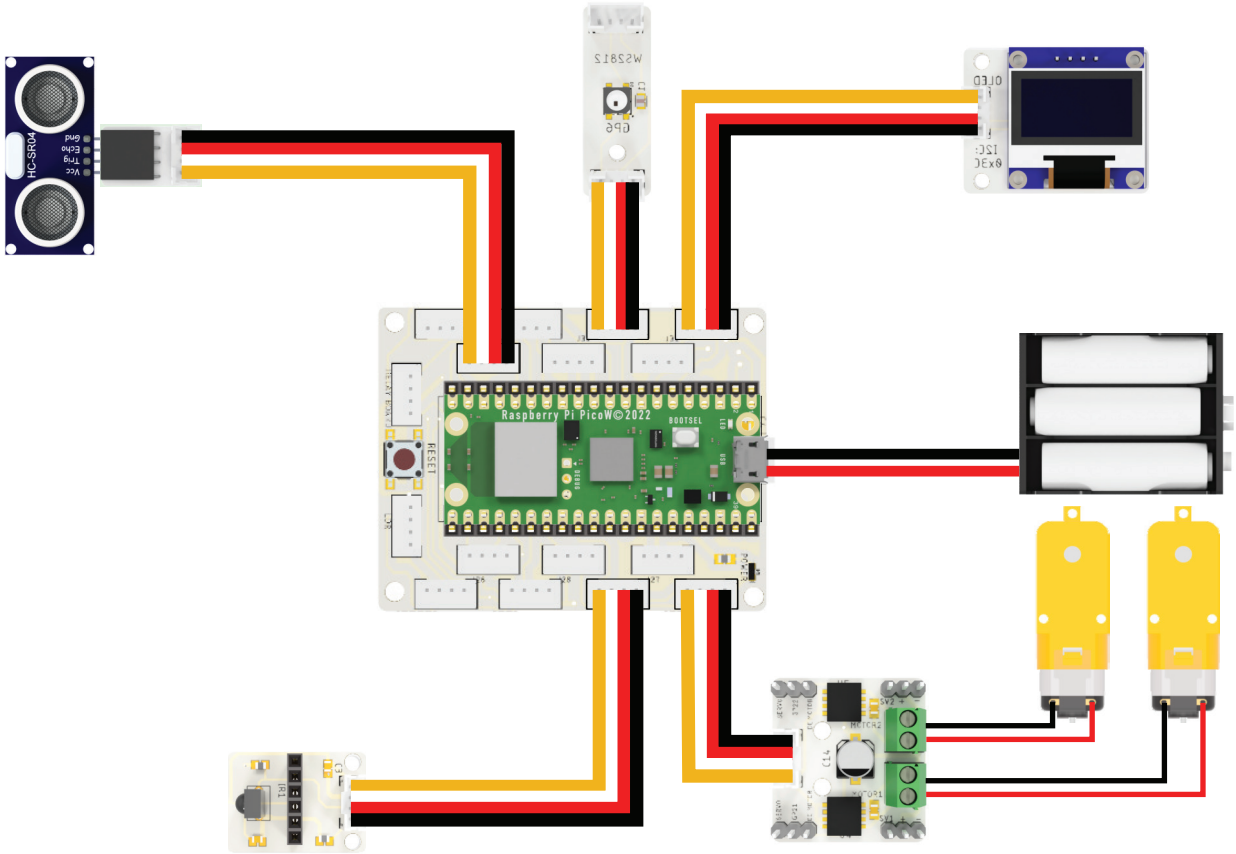



You can use Holes for
cable management.


M3 x 10 mm Screw 4 pcs
Don't Forget to make
all cable connections.



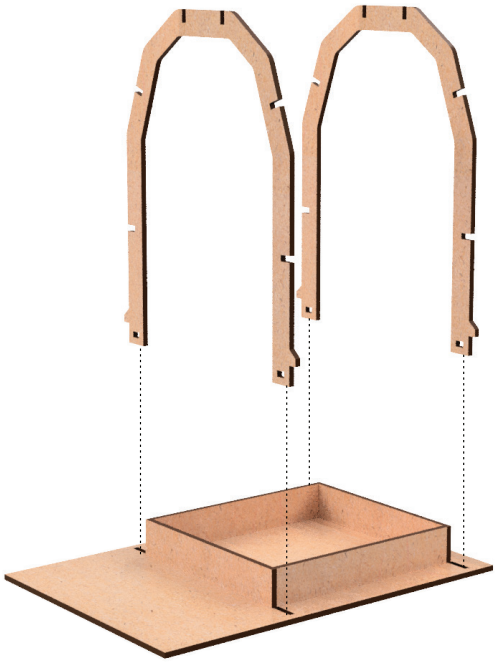


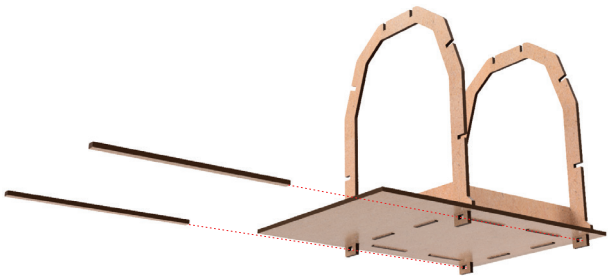

HC-SR04
Vcc
Trig
Echo
Gnd
Raspberry Pi PicoW©2022
BOOTSEL
RESET

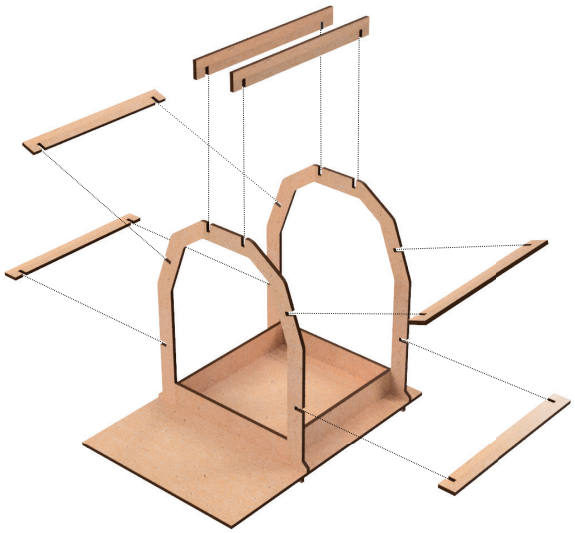
PicoBricks Smart Green House Setup Guide



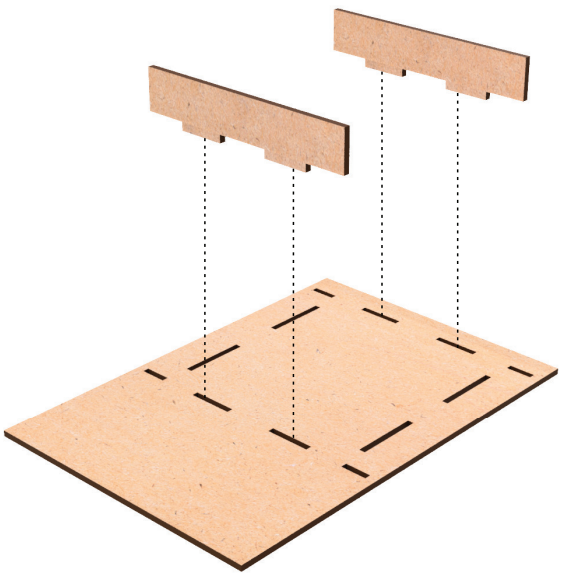


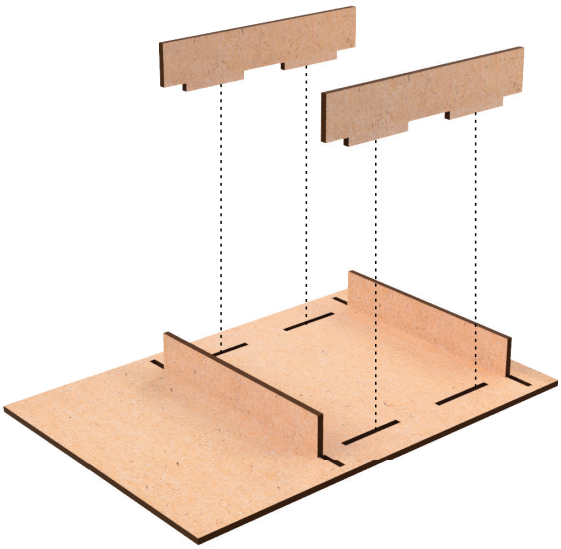


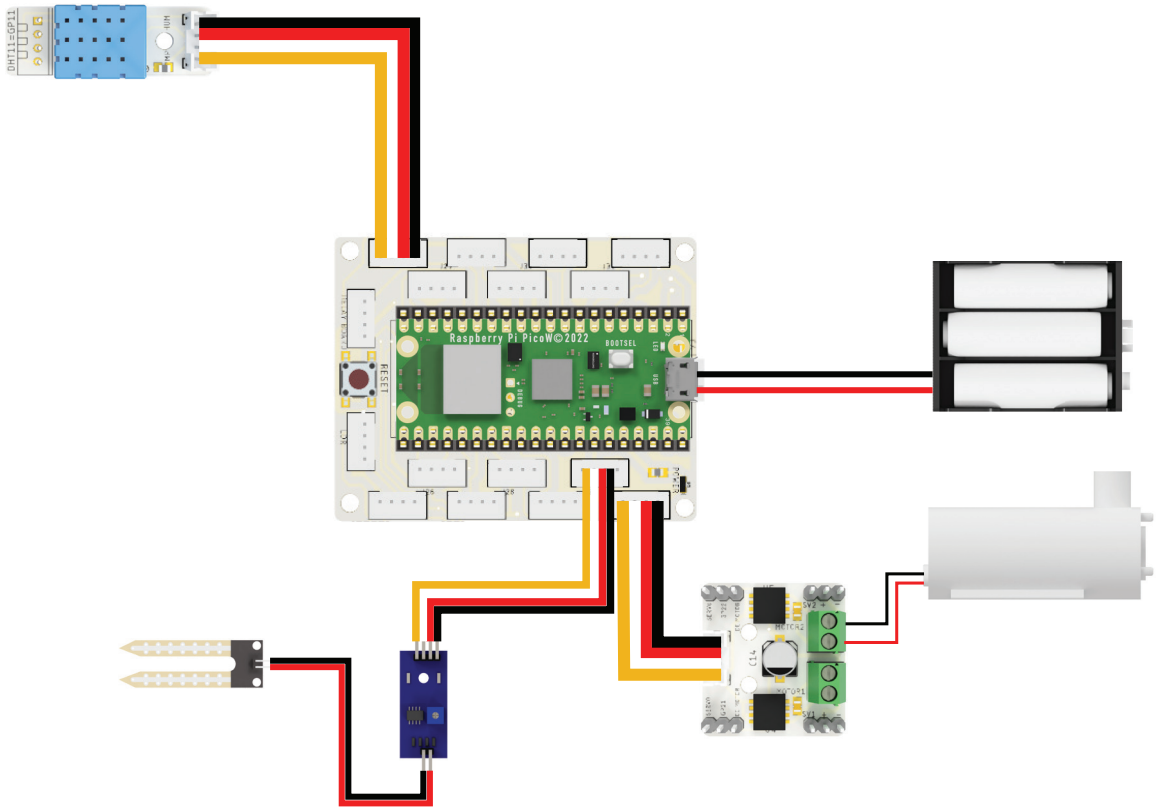











DHT11-GP11
Raspberry Pi Pico W©2022
BOOTSEL

3D MODELS

Two Axis Robot Arm Project 3D Parts

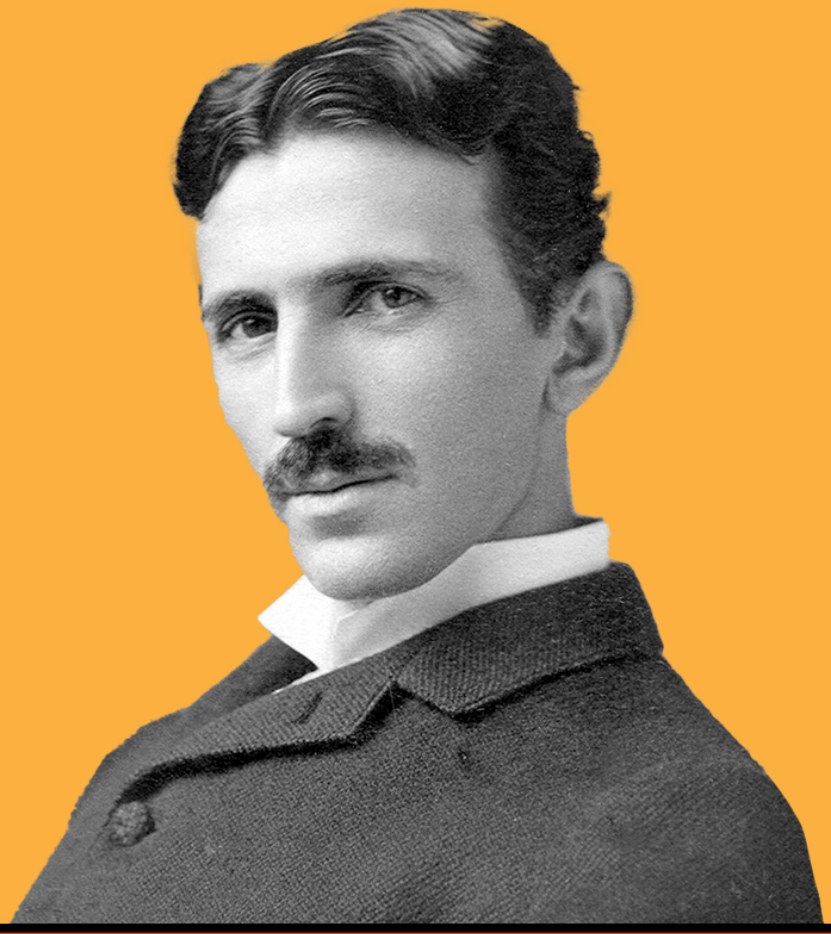
Piggy Bank Project 3D Parts

Maze Solving Robot Project 3D Parts

Greenhouse Control Android App(.apk)

Sera Control MITAppInventor 2 Project File

Voice Controlled Robot Car Project Android App (.apk)



"If you want to find the secrets of the universe, think in terms of energy, frequency and vibration."

Nikola Tesla

pico bricks

Join the community

community.robotistan.com

PicoBricks GitHub

github.com/Robotistan/PicoBricks

picobricks.com