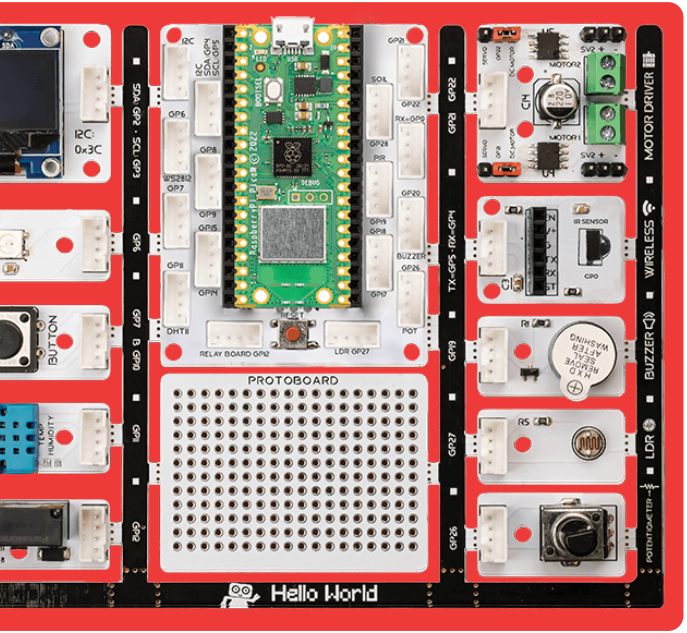
pico bricks

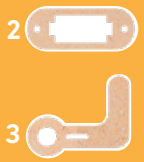
MicroBlocks MicroPython ARDUINO

Safe Box

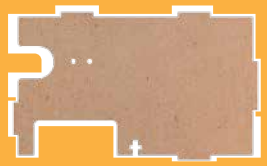
pico bricks



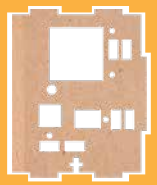

1


2

3


4


5


6


7

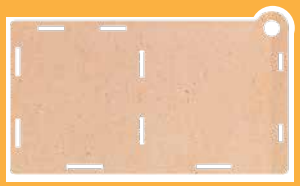

8


9


10


11


12


13


Servo Motor

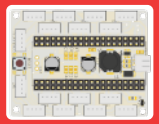

Potentiometer


OLED screen

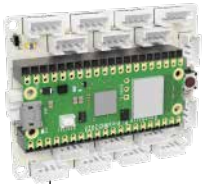

Motor Driver


Button and LED


LDR Sensor

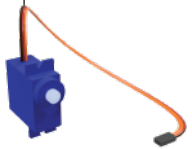

PicoBricks Main Board Module

Project contents



PicoBricks
Main Board Module

SG90 Servo Motor



PicoBricks OLED Module

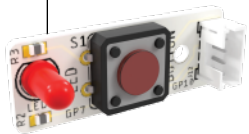
PicoBricks
Motor Driver Module

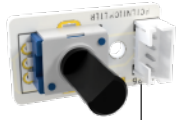
PicoBricks LDR Module

Connecting Cables 5pcs.

Triple Battery Holder

PicoBricks Button LED Module





PicoBricks Potentiometer Module

Micro USB Cable - 1.5 m

M3 Nuts 18pcs.

M3 x 12mm Screw 7pcs.

M3 x 10mm Screw 11pcs.

PicoBricks Safe Box

PicoBricks Safe Box is an educational project kit that aims to make a case that opens when the right password is entered by using the potentiometer and button, after combining the wooden parts and PicoBricks modules coming out of the PicoBricks Safe Box according to the installation steps, and that automatically locks with the LDR sensor inside the case after closing the door.

PicoBricks modules to use with wooden parts in the PicoBricks Safe Box kit: OLED screen, button&LED, motor driver, LDR sensor and potentiometer module are used.

Do You Know This?

The first saved use of encryption, also known as cryptography, was discovered in 1500 BC.

Why Do We Use Passwords?

The unique symbols produced in order to deliver existing content or a physical substance to the right person without any external factors and without deterioration on the quality of the sent susbstance or content are called passwords.

People sometimes use passwords to protect their physical objects and sometimes their written or visual content. The variety of symbols used in passwords is one of the qualities that increase the strength of the password. Likewise, using personal data that cannot be easily guessed by everyone in passwords will increase the strength of your password.

Unplugged 1 : Let's Draw

When and where was the last time you used a password?

Do You Know This?

What Is Cryptography?

It is a process that makes readable data incomprehensible to undesirable people.

For example, there is a number that corresponds to each letter and symbol on the keyboard. We can express the encrypted text we want to write in number by using this tabşe.

ASCII Table

dec	hex	oct	char	dec	hex	oct	char	dec	hex	oct	char	dec	hex	oct	char
0	0	000	NULL	32	20	040	space	64	40	100	@	96	60	140	`
1	1	001	SOH	33	21	041	!	65	41	101	A	97	61	141	a
2	2	002	STX	34	22	042	"	66	42	102	B	98	62	142	b
3	3	003	ETX	35	23	043	#	67	43	103	C	99	63	143	c
4	4	004	EOT	36	24	044	$	68	44	104	D	100	64	144	d
5	5	005	ENQ	37	25	045	%	69	45	105	E	101	65	145	e
6	6	006	ACK	38	26	046	&	70	46	106	F	102	66	146	f
7	7	007	BEL	39	27	047	'	71	47	107	G	103	67	147	g
8	8	010	BS	40	28	050	(	72	48	110	H	104	68	150	h
9	9	011	TAB	41	29	051	)	73	49	111	I	105	69	151	i
10	a	012	LF	42	2a	052	*	74	4a	112	J	106	6a	152	j
11	b	013	VT	43	2b	053	+	75	4b	113	K	107	6b	153	k
12	c	014	FF	44	2c	054	,	76	4c	114	L	108	6c	154	l
13	d	015	CR	45	2d	055	-	77	4d	115	M	109	6d	155	m
14	e	016	SO	46	2e	056	.	78	4e	116	N	110	6e	156	n
15	f	017	SI	47	2f	057	/	79	4f	117	O	111	6f	157	o
16	10	020	DLE	48	30	060	0	80	50	120	P	112	70	160	p
17	11	021	DC1	49	31	061	1	81	51	121	Q	113	71	161	q
18	12	022	DC2	50	32	062	2	82	52	122	R	114	72	162	r
19	13	023	DC3	51	33	063	3	83	53	123	S	115	73	163	s
20	14	024	DC4	52	34	064	4	84	54	124	T	116	74	164	t
21	15	025	NAK	53	35	065	5	85	55	125	U	117	75	165	u
22	16	026	SYN	54	36	066	6	86	56	126	V	118	76	166	v
23	17	027	ETB	55	37	067	7	87	57	127	W	119	77	167	w
24	18	030	CAN	56	38	070	8	88	58	130	X	120	78	170	x
25	19	031	EM	57	39	071	9	89	59	131	Y	121	79	171	y
26	1a	032	SUB	58	3a	072	:	90	5a	132	Z	122	7a	172	z
27	1b	033	ESC	59	3b	073	;	91	5b	133	[	123	7b	173	{
28	1c	034	FS	60	3c	074	<	92	5c	134	\	124	7c	174	|
29	1d	035	GS	61	3d	075	=	93	5d	135	]	125	7d	175	}
30	1e	036	RS	62	3e	076	>	94	5e	136	^	126	7e	176	~
31	1f	037	US	63	3f	077	?	95	5f	137	_	127	7f	177	DEL

- Let's together analyze a encrypted text by using ASCII character table.

80 - 105 - 99 - 111 - 66 - 114 - 105 - 99 - 107 - 115

P i c o B r i c k s

Unplugged 2 : ASCII Encryption

Convert the encrypted numbers below into a text by using the ASCII character table.

72-101-108-108-111-32-77-97-107-101-114

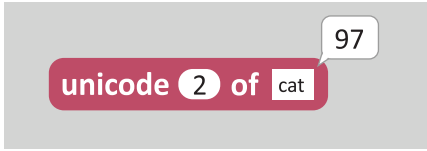
Do You Know This?

Do you know that using PicoBricks and MicroBlocks IDE you can automatically decryption a text that we have given as encrypted or you can automatically encrypt a text by using the ASCII character table?

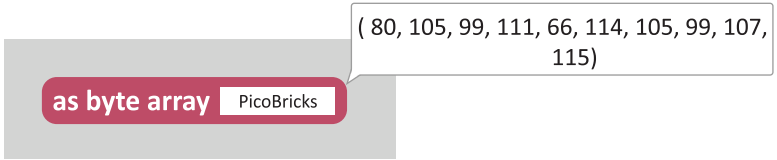
Firstly, open advanced blocks by making the following settings.

You will get the necessary blocks for ASCII code by opening the "Data" blocks.

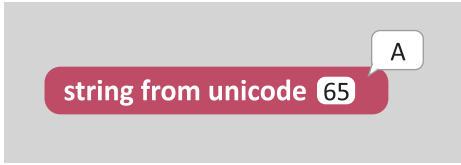
- The following code block is used to find the ASCII equivalent of the characters in a textual expression you've written. We see that the ASCII equivalent of the letter "a", which is the second character in the text "cat", is the number 97.



- The following code block is used to get the numerical equivalent of a textual expressionyou've written according to the ASCII character table. For example, the code block below gives the ASCII equivalent of the text "PicoBricks".



- The following code block is used to find the character equivalent of a number in the ASCII table. For example, the character equivalent of the number "65" in the ASCII table is "A".

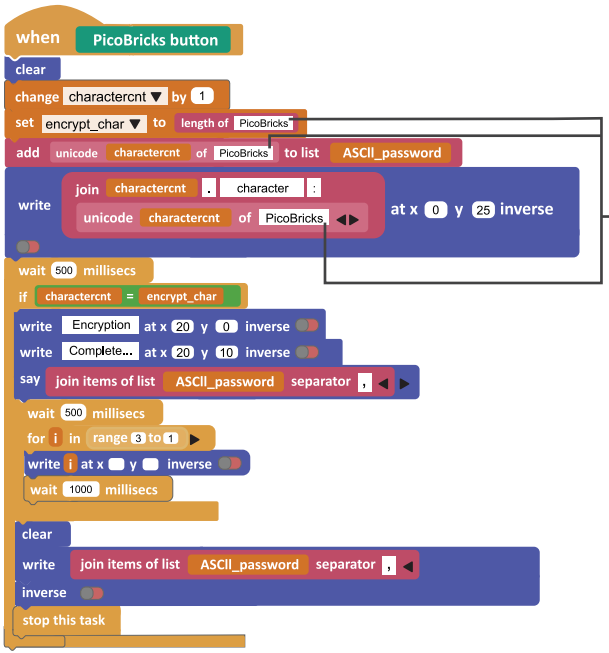


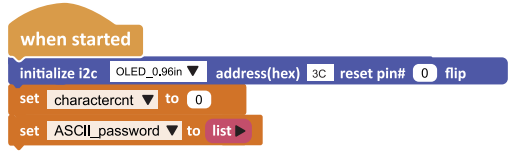
MicroBlocks Activity 1: ASCII Encryption With PicoBricks

You can use the following MicroBlocks code to encrypt any text you want with PicoBricks by using the ASCII table.

After specifying the text you want to encrypt in the code, each tiem you click the PicoBricks button, you will see the equivalent in ASCII table of letters in the text you want to encrypt in order. After seeing the numerical equivalent of all the characters in the text you want to encrypt in the ASCII table, the encrypted version of the text will appear on both the PicoBricks OLED screen and the MicroBlocks blocks after a 3-second countdown.

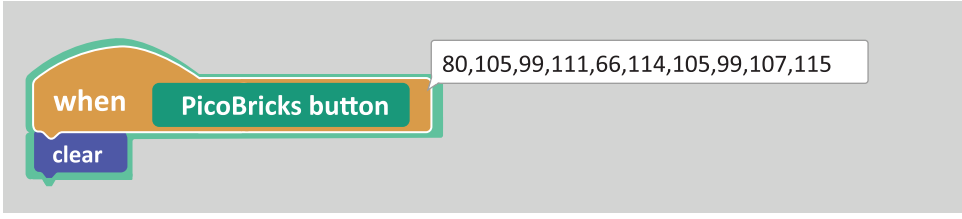
MicroBlocks Code of The ASCII Encryption





Type the text you want to encrypt in these fields.

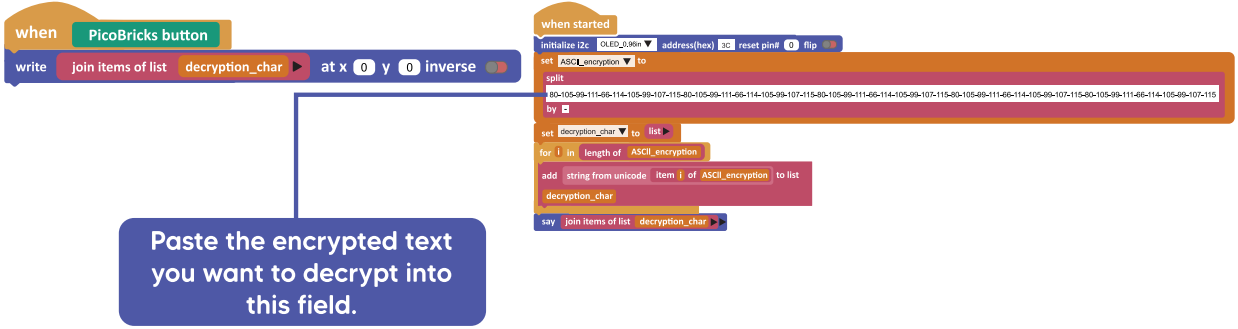
- The encrypted version of the text you wrote by using the ASCII table will also appear on the blocks.



MicroBlocks Activity 2: ASCII Decryption With PicoBricks

With PicoBricks, we can decrypt a given encrypted text by using the ASCII table. For this, you can create the code blocks below and print the encrypted text you specified in the code on the OLED screen of PicoBricks.

MicroBlocks Code of The ASCII Decryption

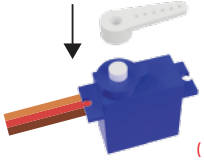


Unpulugged 3 : Setup

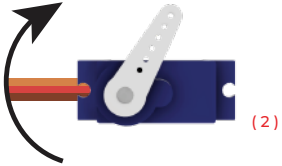
Let’s complete the setup by combining the wooden parts from the PicoBricks Safe Box set.

Servo Motor Calibration

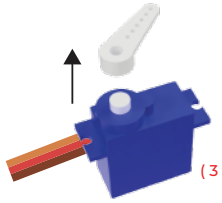
Before starting the assembly, you have to manually calibrate the angles of the servo motors. Otherwise, Servo Motors won't be working properly.



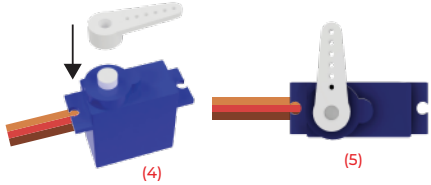
Attach the servo horn to the servo motor (1)



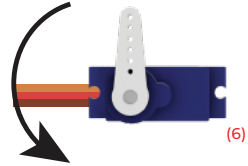
Then slowly turn the servo horn clockwise until it stops. It is not a problem if the servo horn is not the same as the angle shown in the image above. The important thing here is that you have hit the last angle of the servo. (2)



Remove the servo horn from the servo motor (3)



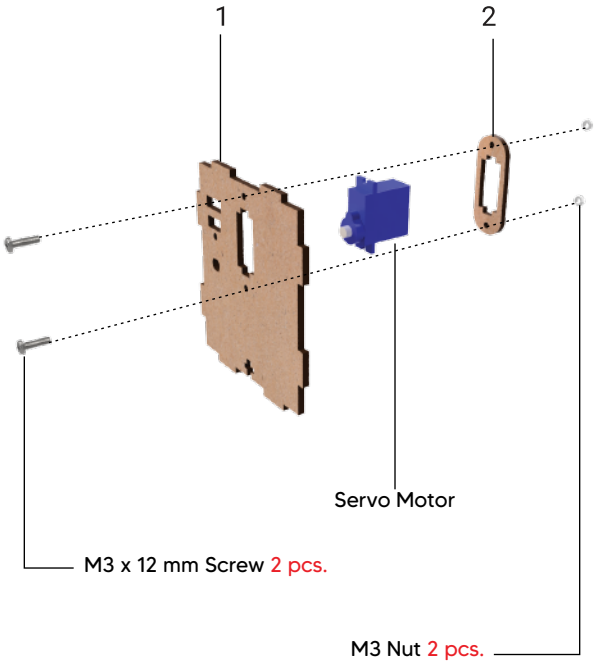
Reattach (4) and reposition the servo horn perpendicular to the servo motor as shown. (5)

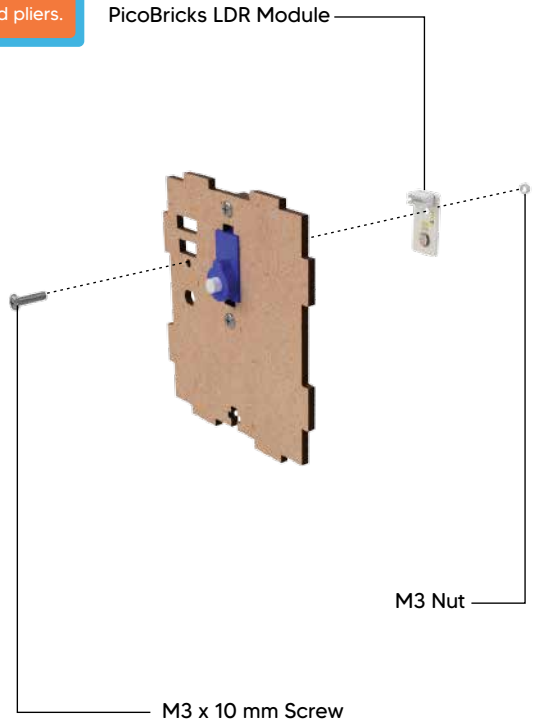


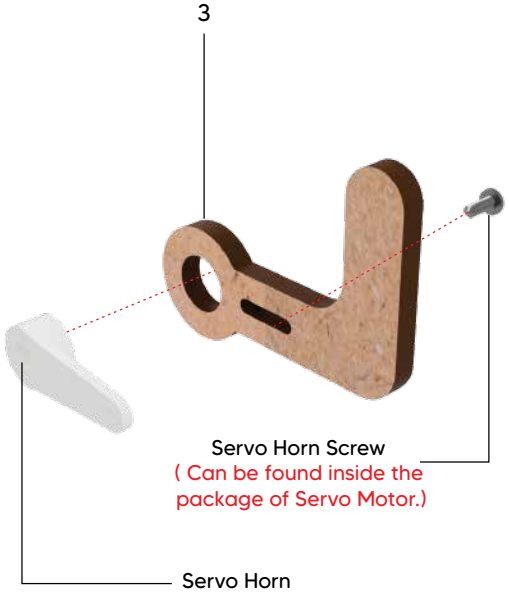
Slowly turn the servo horn counterclockwise (6) until it is parallel with the servo motor, as seen in the image.(7)

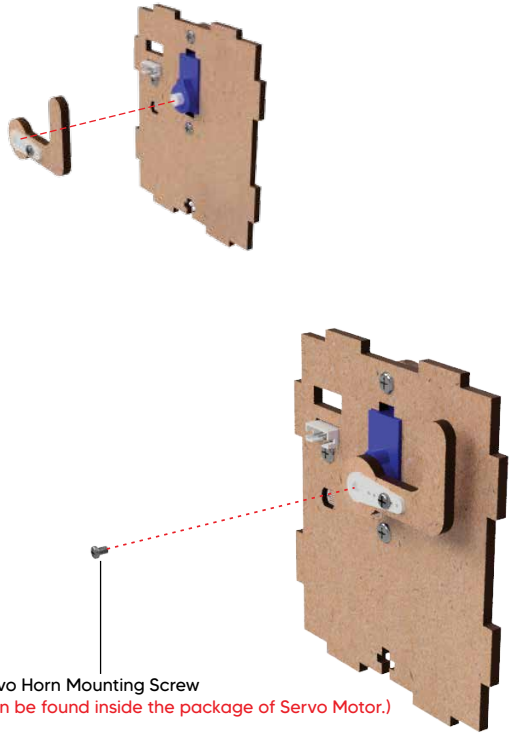
When this step is finished, it means that the servo motor is in the center position. It is important that you apply this process to other servo motors in the set. Afer processing the other motors, remove the servo horn and set aside for assembly.

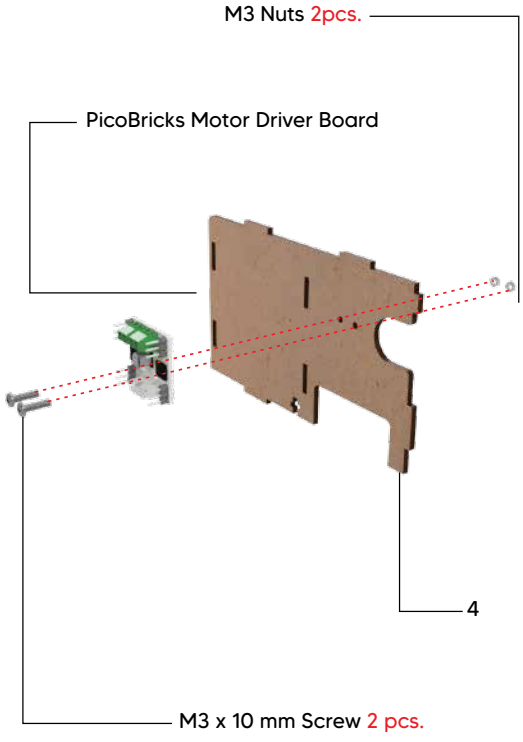
In this step, you will need a screwdriver and pliers.

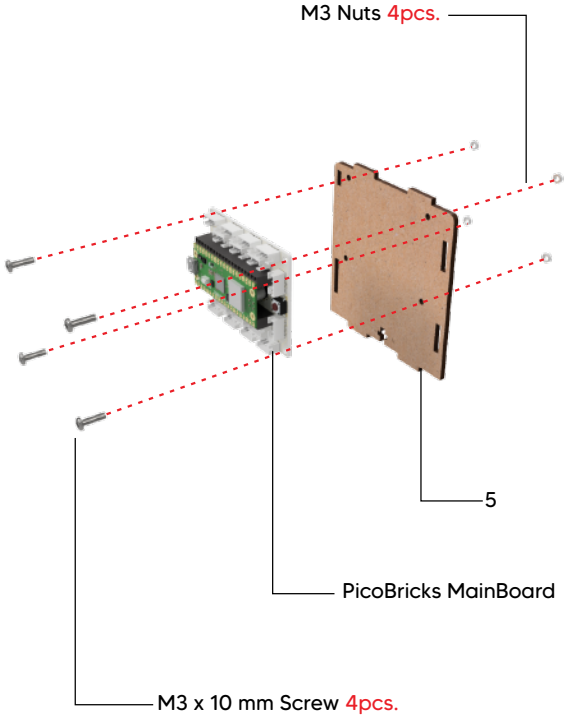


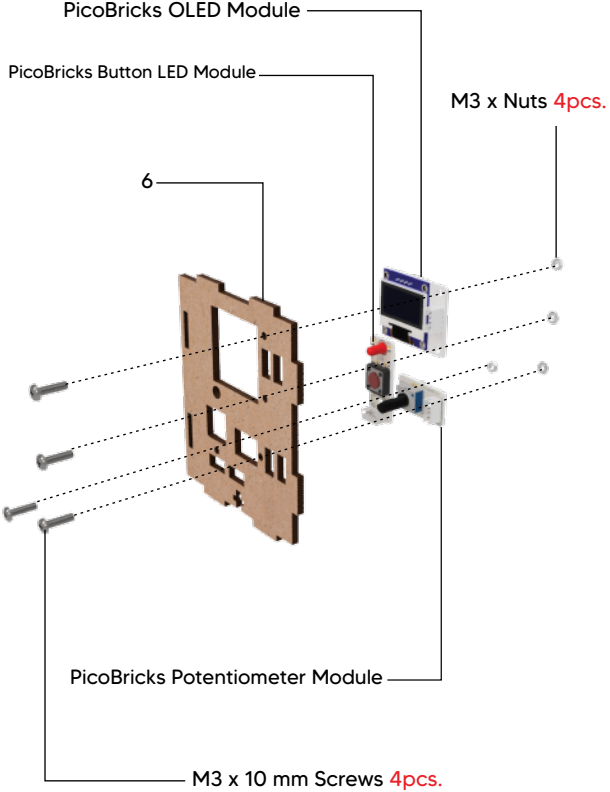


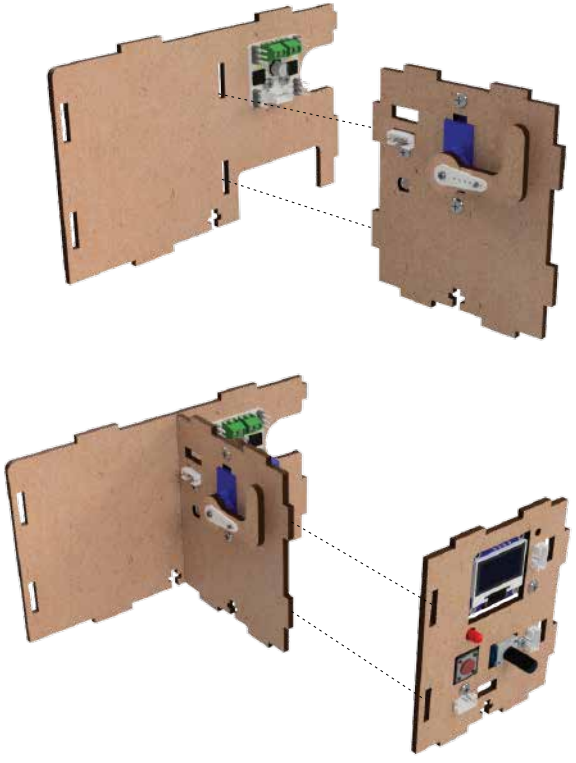

3
Servo Horn Screw
(Can be found inside the
package of Servo Motor.)
Servo Horn


Servo Horn Mounting Screw
(Can be found inside the package of Servo Motor.)


M3 Nuts 2pcs.
PicoBricks Motor Driver Board
4
M3 x 10 mm Screw 2 pcs.

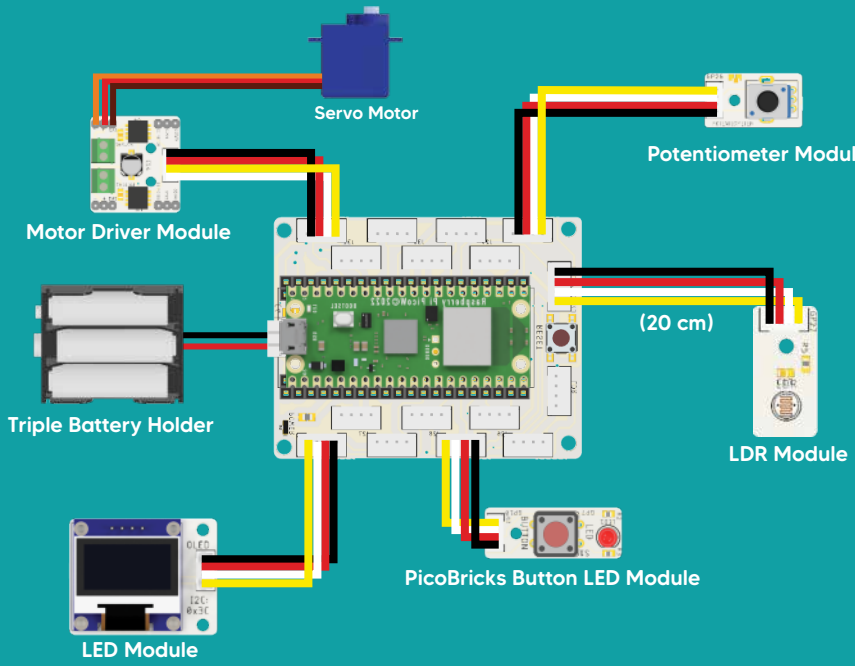

M3 Nuts 4pcs.
5
PicoBricks MainBoard
M3 x 10 mm Screw 4pcs.

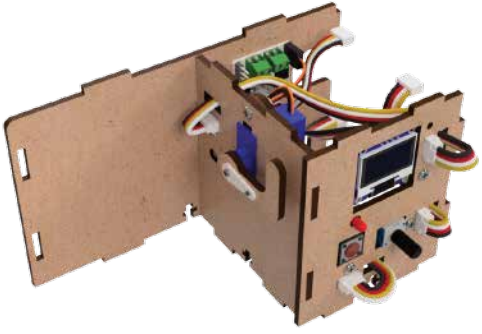

PicoBricks OLED Module
PicoBricks Button LED Module
M3 x Nuts 4pcs.
6
PicoBricks Potentiometer Module
M3 x 10 mm Screws 4pcs.

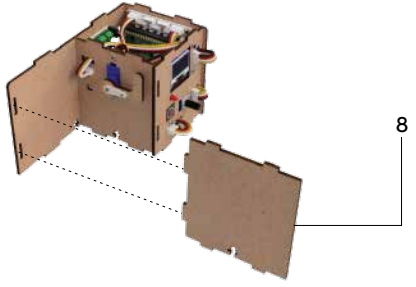


Setting Up The Circuit

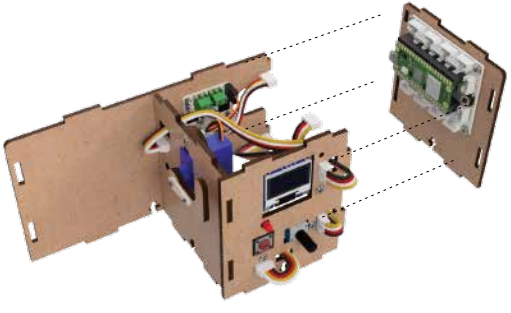
Let’s get to know the circuit elements of Safe Box that we completed the setup and make the circuit setup with PicoBricks modules.

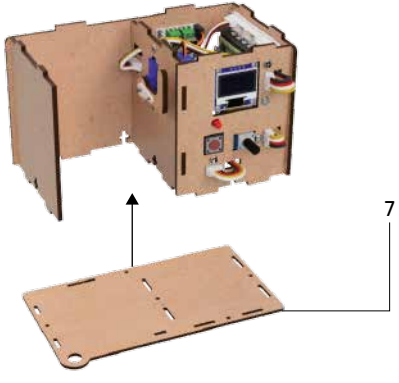


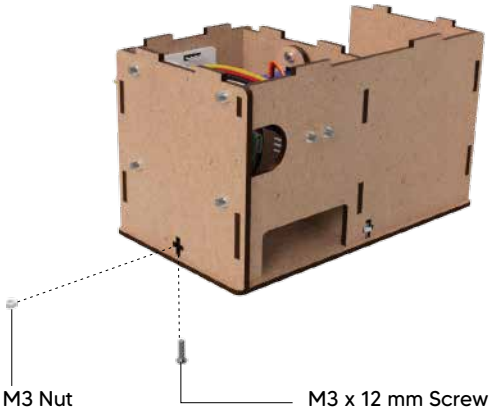


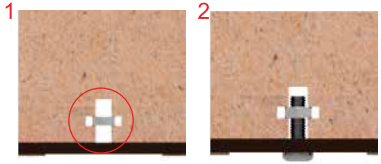


Make sure all cables are ready to be socketed to MainBoard. You can go back to page 18 for circuit diagram.

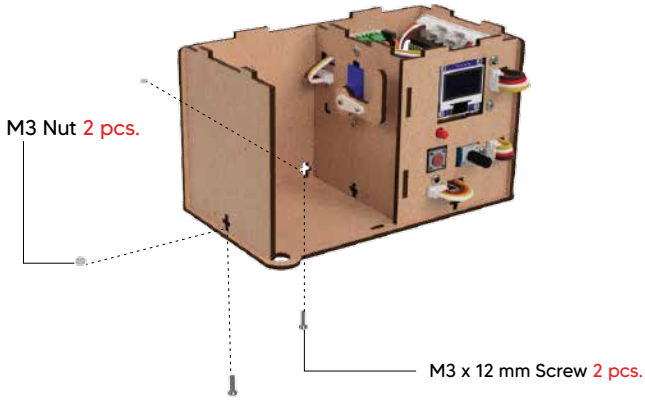


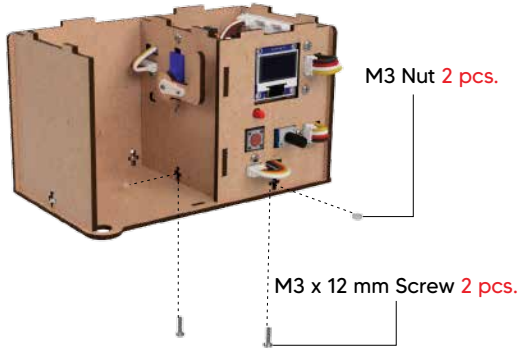


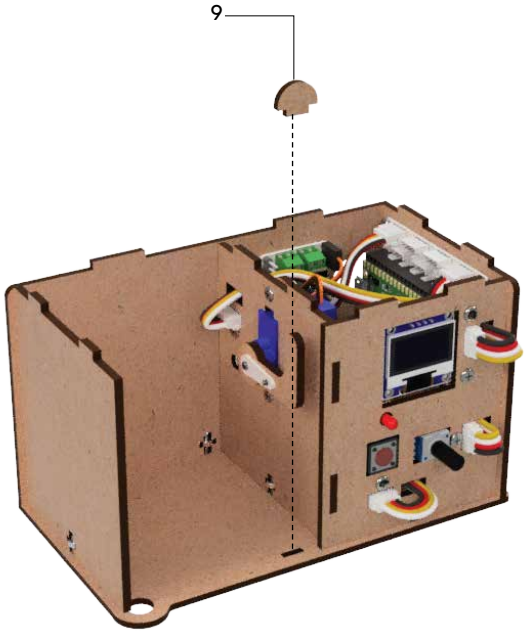


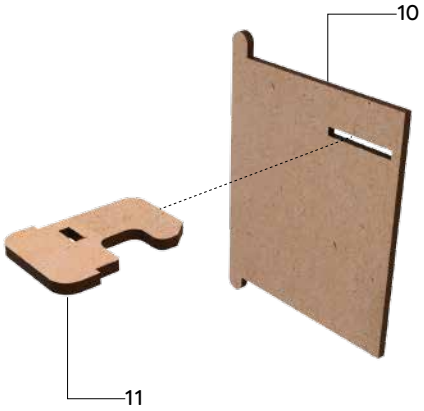


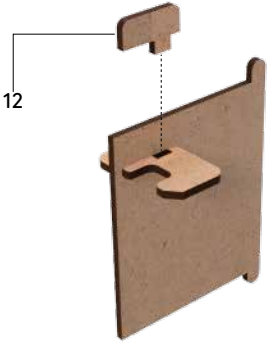
For T Slot joint setup first, place the M3 Nut into the Nest (1) Then tighten the Joint with using M3 x 12 mm Screw.(2)

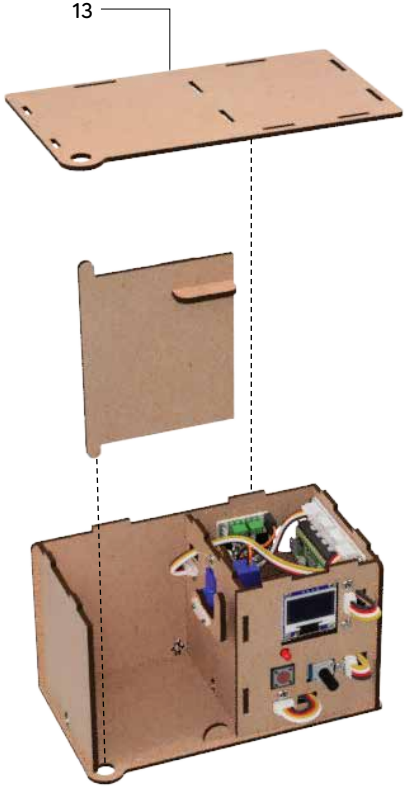




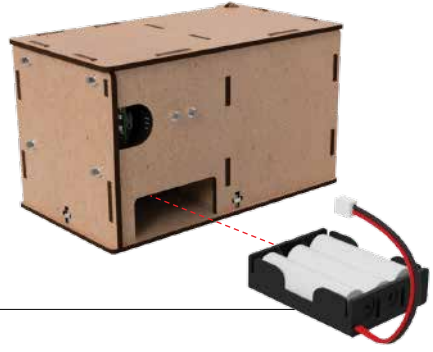

9


10
11


12



Triple Battery Holder



The Assembly is finished and you can move on the coding steps.

Do you know that you can make encrypting by using “Mors Code”?

Unplugged 4: Let’s Paint

After putting all the pieces together, let’s paint the Safe Box in an original way.

Useful Information

You can choose acrylic paint in the painting process. Acrylic paint is a water-based, non-toxic and quick-drying paint that can be used in many applications. The surfaces we will paint are made of materials not harmful to human health.

Let's Get To Know The Circuit Elements

Potentiometer: It is an input sensor that we can change the resistance value applied to the circuit with physical intervention.

OLED Screen: It is a module that we can get textual, visual or animation-type outputs.

Button & LED: It is the circuit element that enables a process to start and end by applying pressure on it. LED: It is the circuit element that gives red light output.

Motor Driver: It is a circuit element that adjusts the speed and frequency of the circuit elements.

Servo Motor: It is used to provide movement in mechanical projects that require angular movement.

LDR Sensor: It is a circuit element that detects the light amount of the environment.

Code of The Project

Let's print the MicroBlocks, Thonny and Arduino code of our project.

MicroBlocks

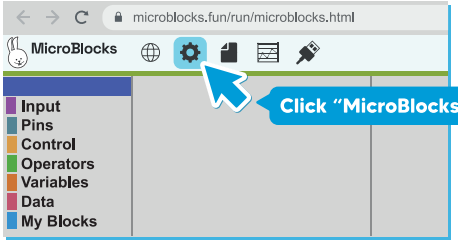
How to Use MicroBlocks?

A Let's open https://microblocks.fun/ in the browser.

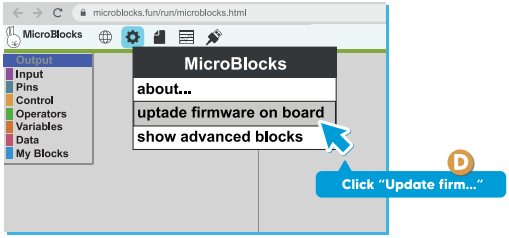
rbt.ist/jjy

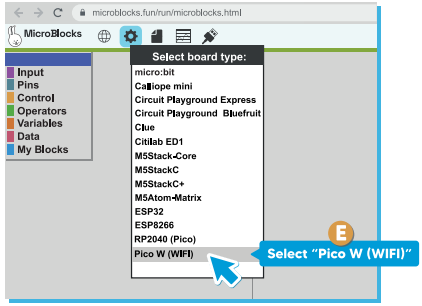
Scan Code

B Click "RUN" button

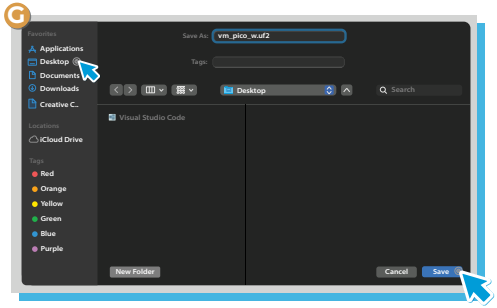


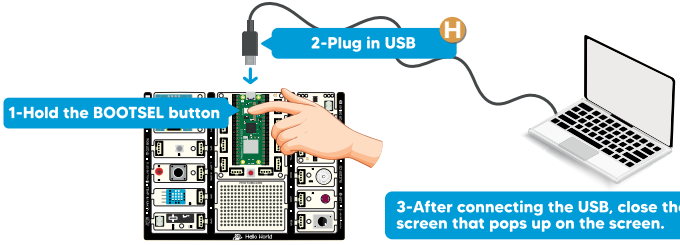
C Click "MicroBlocks" button


microblocks.fun/run/microblocks.html
MicroBlocks
Output
Input
Pins
Control
Operators
Variables
Data
My Blocks
MicroBlocks
about...
uptade firmware on board
show advanced blocks
D
Click "Update firm..."


microblocks.fun/run/microblocks.html
MicroBlocks
Input
Pins
Control
Operators
Variables
Data
My Blocks
Select board type:
micro:bit
Calliope mini
Circuit Playground Express
Circuit Playground Bluefruit
Clue
Citilab ED1
M5Stack-Core
M5StackC
M5StackC+
M5Atom-Matrix
ESP32
ESP8266
RP2040 (Pico)
Pico W (WIFI)
E
Select "Pico W (WIFI)"

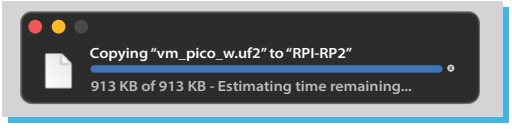

F
Firmware Install
Connect USB cable while holding down the white BOOTSEL button before proceeding.
You will be asked to save the firmware file.
Select RPI-RP2 as the destination drive, then click save.
OK

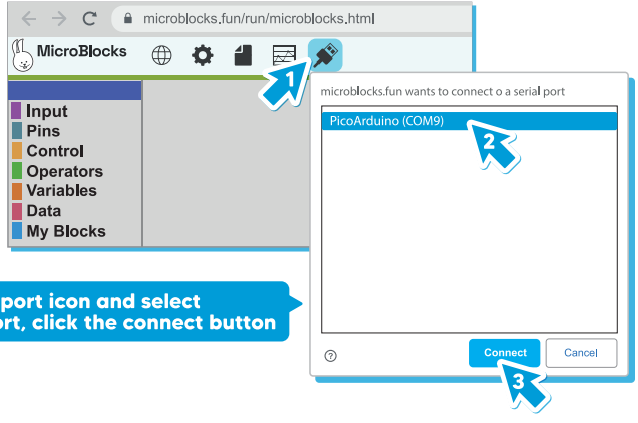

G
Favorites
Applications
Desktop
Documents
Downloads
Creative C...
Locations
iCloud Drive
Tags
Red
Orange
Yellow
Green
Blue
Purple
Save As: vm_pico_w.uf2
Tags:
Desktop
Search
Visual Studio Code
New Folder
Cancel
Save


2-Plug in USB
H
1-Hold the BOOTSEL button
3-After connecting the USB, close the RPI-RP2 (E:) screen that pops up on the screen.

I

Drag the "vm_pico_w.uf2" file into the "RPI-RP2" driver.





K

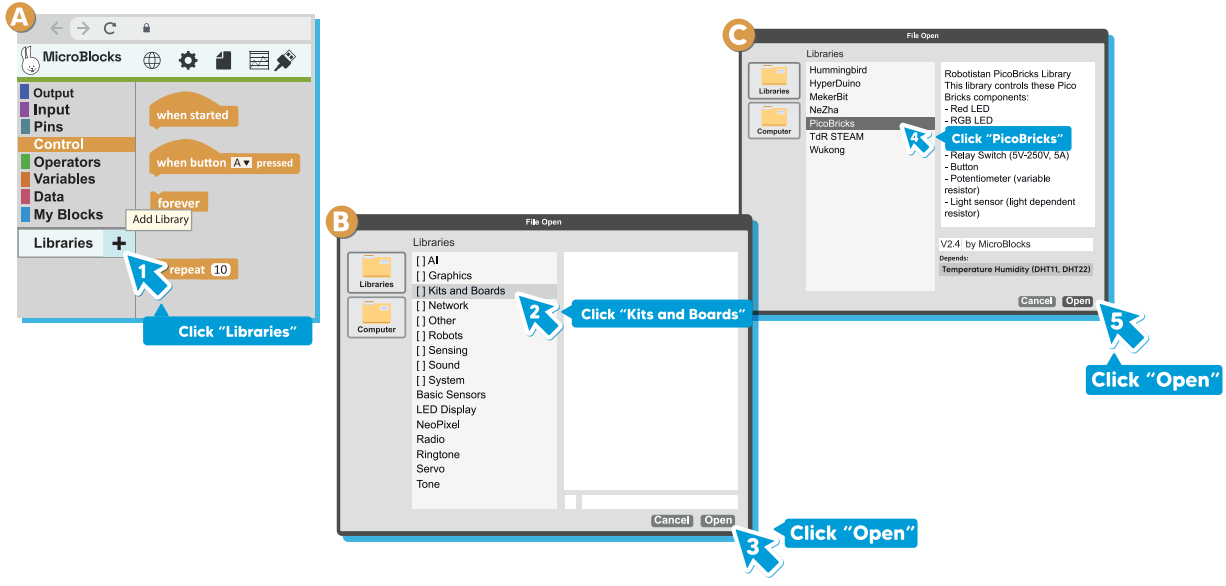
Click on the USB port icon and select the Picobricks port, click the connect button

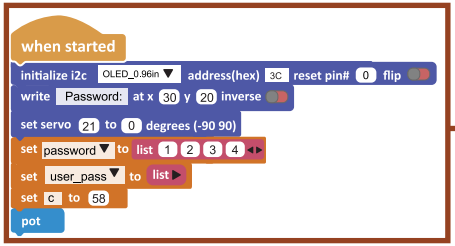
J

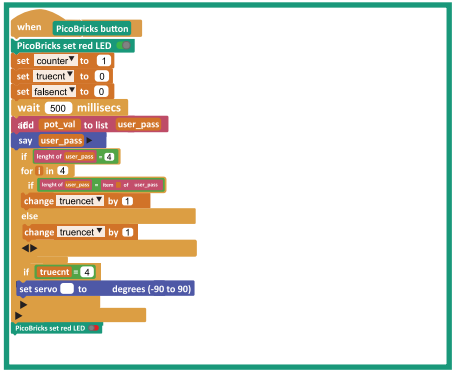
If the icon is green, it is connected

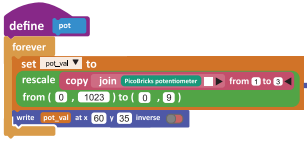
MicroBlocks Code of The Project

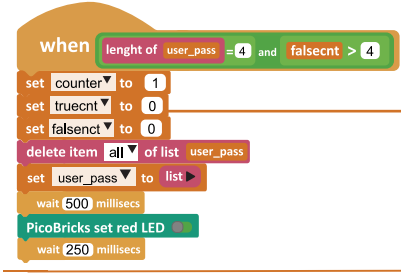
After opening a new project on MicroBlocks IDE, let's install the PicoBricks library by following the steps below.

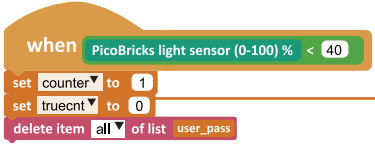












Let's determine the correct password and print the password determined by the Potentiometer on the OLED screen.

Function that continuously prints the value of the potentiometer to the screen.

Actions to be taken if wrong password is entered.

When the button is pressed, it assigns the value of the potentiometer to a list. When the number of elements of the list is 4, it is, compared to the password list and unlocks if it is true.

LDR detects when the door is closed from the light level and locks the door.

pico bricks IDE

A

Let's open https://picobricks.com/ in the brower.

picobricks.com

rbt.ist/ide

B

Click "IDE" button.

Educational + Be Our Ambassador IDE

PicoJR

A simple and visual-based coding platform for kids who are new to robotic coding

Go

PicoBlockly

Easily code PicoBricks with block coding and see python code at the same time.

Go

PicoPY

Do high-level coding with text-based python.

Go

Simulator

With PicoBricks Simulator you can experience using PicoBricks.

Go

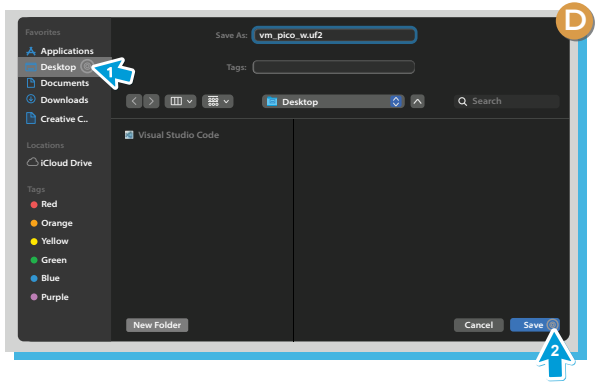
C

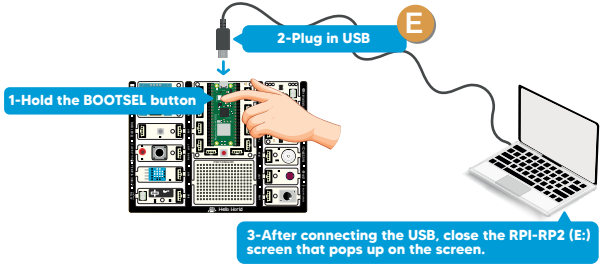
Click "PicoBlockly" button.

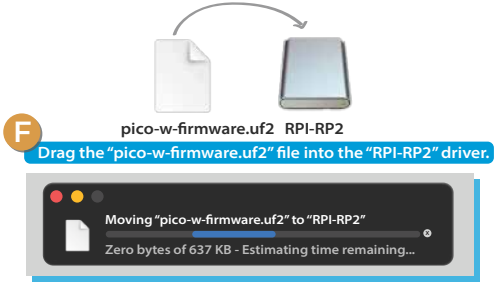

Block
PicoBricks
PicoBricks Simulator
A

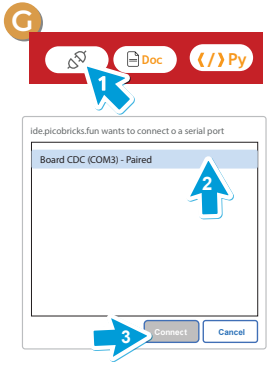

Block
PicoBricks
PicoBricks Simulator
Upload Firmware
Upload Library
Upload Code For Mobile
B
Click “Upload Firmware” button.

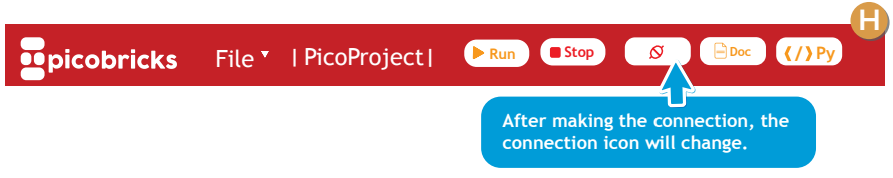

C
Select “Pico W (Wi-Fi)”.


D
Save As: vm_pico_w.uf2
Tags:
Favorites
Applications
Desktop
Documents
Downloads
Creative C..
Locations
iCloud Drive
Tags
Red
Orange
Yellow
Green
Blue
Purple
Visual Studio Code
New Folder
Cancel
Save
1
2

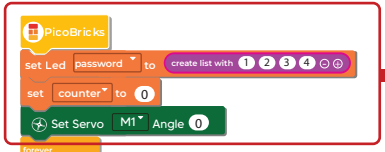

E
2-Plug in USB
1-Hold the BOOTSEL button
3-After connecting the USB, close the RPI-RP2 (E:) screen that pops up on the screen.


pico-w-firmware.uf2
RPI-RP2
F
Drag the "pico-w-firmware.uf2" file into the "RPI-RP2" driver.
Moving "pico-w-firmware.uf2" to "RPI-RP2"
Zero bytes of 637 KB - Estimating time remaining...

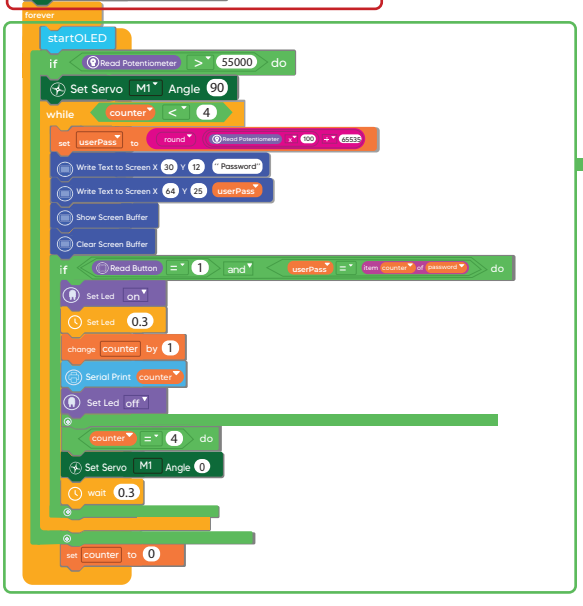

G
Doc
Py
1
ide.picobricks.fun wants to connect to a serial port
Board CDC (COM3) - Paired
2
3
Connect
Cancel


H
picobricks
File
| PicoProject |
Run
Stop
Doc
Py
After making the connection, the connection icon will change.

PicoBricks IDE Code of The Project

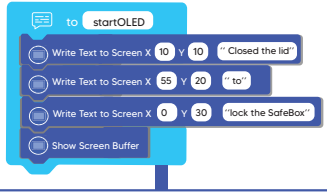


Let's define each character of the correct password to the elements of the list by creating a list called "password".
Let's determine its initial value by creating a variable called "counter" and set the servo motor to the initial position.



It is a loop and condition structure that runs until the correct password is entered after the lid of the SafeBox is closed.

When the lid is closed, the password screen is reset.



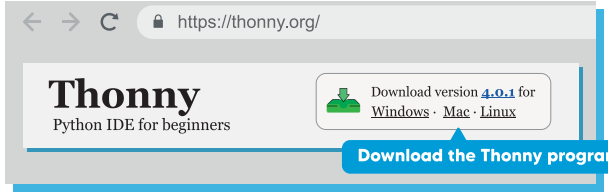
It is the function that determines the texts to be written on the OLED screen when the lid of SafeBox is closed.

How to Use MicroPython?

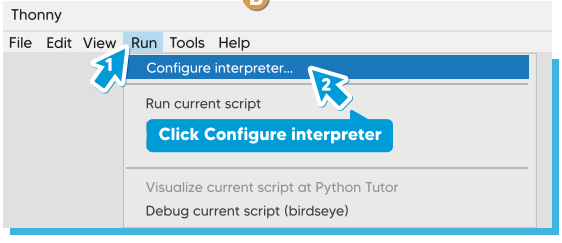
A Let's open thonny.org in the browser.

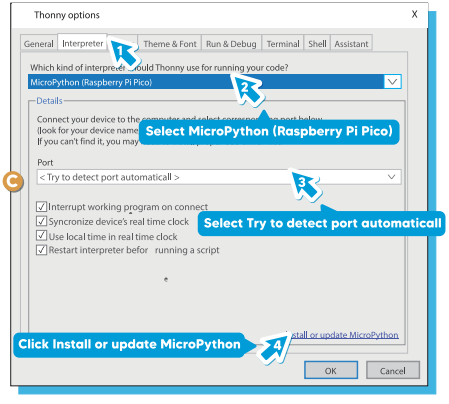
rbt.ist/tho

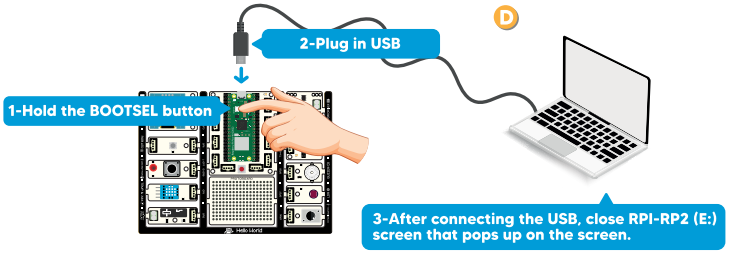
Scan Code

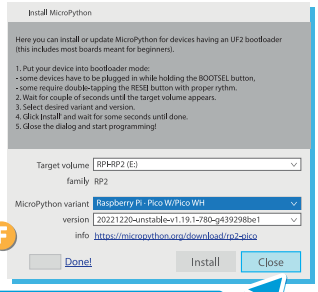


B

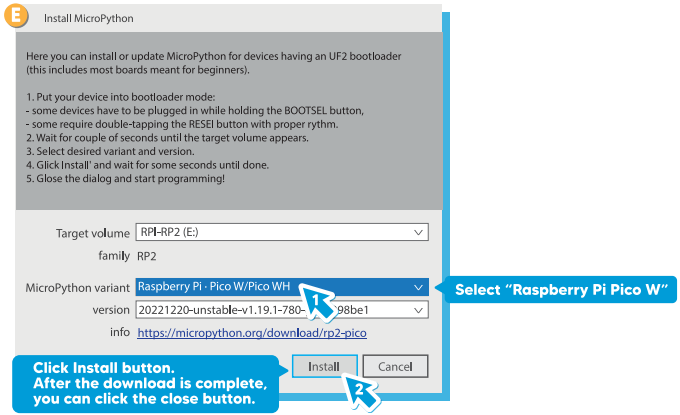






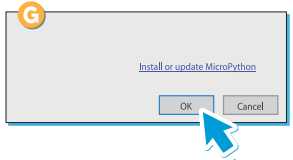


After the download is complete, you can click the close button.



Select “Raspberry Pi Pico W”

Click Install button.
After the download is complete, you can click the close button.



Click “OK” button to complete the installation process.

You are ready now.
Let’s start coding
with MicroPython

```
from machine import Pin, I2C, ADC, Timer, PMW  #to acces the hardware picobricks
from picobricks import SSD1306_I12C  #oled library
import utime #time library

WIDTH  = 128
HEIGHT = 64
#define the widht and height values

LIGHT_THRESHOLD = 55000
OPEN_POSITION = 1920
CLOSED_POSITION = 5500
servo =PMW(Pin(21,Pin.OUT))
servo. freq (50)
servo.duty_u16(CLOSED_POSITION)
l
button = Pin(10, Pin.IN)

ldr = ADC (Pin(27))

led = Pin(7, Pin.OUT)

sda=machine.Pin (4)
scl=machine.Pin (5)
#we define sda and scl pins for inter-path communication
i2c=machine.I2C (0,sda=sda, scl=scl, freq=1000000) #determine the frequency values

oled = SSD1306_I2C(128, 64, I2C)
pot= ADC(Pin(26))

#Define the correct password
correct_password =[1, 2, 3, 4,]
```

Let's define the necessary libraries and variables and then set the correct password in the code.

```
#Define a list to store the password
password = [0, 0, 0, 0]
oled.text("Safe Box" ,30,12)
oled.text(" Press the button: ,0,45)
oled.show()
utime.sleep(0.1)
oled.fill(0)
digit = int ((pot.read_u16()*10)/65536)
oldDigit = 0
lock_state = 0    #0 is locked, 1 is unlocked    count=int((pot.read_u16()*10)/65536)
buttonReleased = 1
passIndex = 0
def lockTheSafe()
    oled. fill (0)
    oled. text ("Locking..." ,30,12)
    oled.show()
    utime.sleep (0.3)
    servo.duty_u168CLOSED_POSITTION)
    oled.fill(0)
    oledDigit = 0
def lockTheSafe():
    oled.fill(0)
    oled.text ,8"Opening...", 30,20)
    oled.show()
    utime.sleep (0.3)
    dervo.duty_u16(OPEN_POSITION)
    utime.sleep,(5)
def passwordCheck (definedPassword, enteredPassword):
    for i in range (len(definedPassword)):
        if definedPassword [i]    = enteredPassword [i]:
          return False
    return True

digitCounter = 1
```

Let's determine the expressions to be written on the OLED screen and define the functions that allow us to unlock, lock and run the potentiometer continuously for the code.

```
while True:
    if lock_state:
        if ldr.read_u16() > LIGHT_THRESHOLD:
            lockTheSafe()
            lock_state = 0
            utime.sleep (2)
    else:
            digit = int((pot.read_u16()*10)/65536)
            if digit != oldDigit:
            oldDigit = digit
            oled.fill(0)
            oled.text("Password",30,10)
            oled.hline(25, 40, 9, 0xffff)
            oled.hline(50, 40, 9, 0xffff)
            oled.hline(75, 40, 9, 0xffff)
            oled.hline(100, 40, 9, 0xffff)
            for c in range (digitCounter-1):
                oled.text(str(password[c]),25*(c+1),30)

            oled.text(str(digit),25*digitCounter,30)
            oled.show()
            print("button RELEASED")
    if button.value() == 0 and buttonReleased == 0:      .# button released (for latch detection)
            print("button RELEASED")
            buttonReleased = 1
                    led.value(0)
    if button.value() == 1 and buttonReleased == 1:          # button pressed (for latch detection)
        print("button preseed")
        buttonReleased = 0
        led.value(1)                    "
        utime.sleep(0.3)
        password[passIndex] = digit
        digitCounter += 1
        oldDigit = 0
        if passIndex >= 3:
        passIndex = 0
        digitCounter = 1
        if (passwordCheck(correct_password, password)):
            unlockTheSafe()
            lock_state = 1
        else:
            oled.fill(0)
            oled.text("Try Again",30,20)
            oled.show()
            utime.sleep(1.5)
        else:
            passIndex += 1
```

Let's create the code lines that print the password digits determined by the potentiometer to the OLED screen.

Let's check the set password. If it is correct, let's open the door, if it is wrong let's print “try again” text. If the amount of light in the environment is low, that is, if the door is closed, let's lock the door.

Arduino IDE

How to Use Arduino IDE?

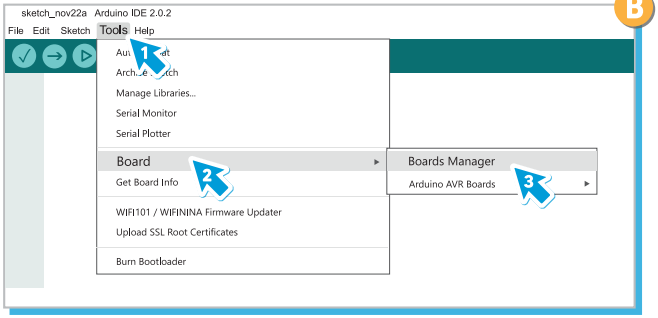
A Let's open https://www.arduino.cc/en/software in the browser.

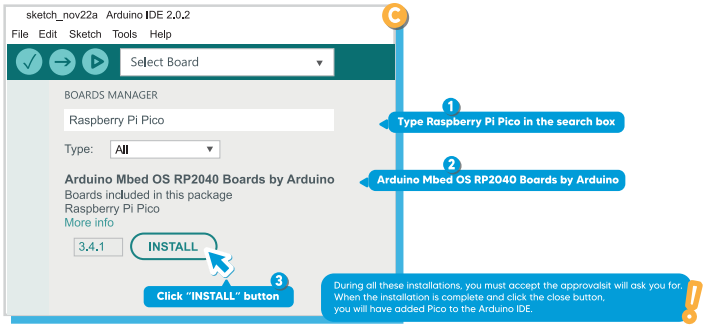
rbt.ist/ardu

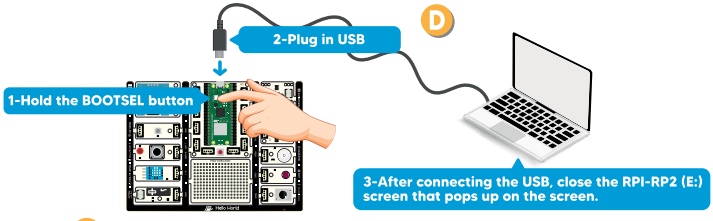
Scan Code

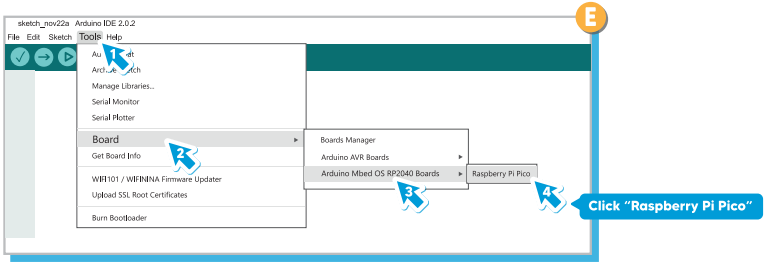
Download the Arduino IDE program by selecting your system

B









You are ready now.
Let's start coding with Arduino

safebox

```
#include <Wire.h>
#include "ACROBOTIC_SSD1306.h" // v1.0.0
#include <Servo.h>

#define pot 26
#define led 7
#define ldr 27
#define button1 10

char correct_password[] = {1, 1, 1, 1};
char password[] = {0, 0, 0, 0};
int oldDigit = 0;
int lock_state = 0;// 0 is locked, 1 is unlocked    count=int((pot.read_u16()*10)/65536)
int buttonReleased = 1;
int passIndex = 0;
int digit = (analogRead(pot) * 9 / 1023);
Servo servo;

int LIGHT_THRESHOLD = 500 ;
int OPEN_POSITION = 150 ;
int CLOSED_POSITION = 0 ;
int digitCounter = 1;

void lockTheSafe() {
  oled.clearDisplay();
  oled.setTextXY(2, 2);
  oled.putString("Locking...");
  delay(300);
  servo.write(CLOSED_POSITION);
  oldDigit = 0;
  oled.clearDisplay();
}
void unlockTheSafe() {
  oled.clearDisplay();
  oled.setTextXY(2, 2);
  oled.putString("Opening...");
  delay(300);
  servo.write(OPEN_POSITION);
  delay(5000);
  oled.clearDisplay();
}
```

Let's define the required libraries, pins and functions.

1

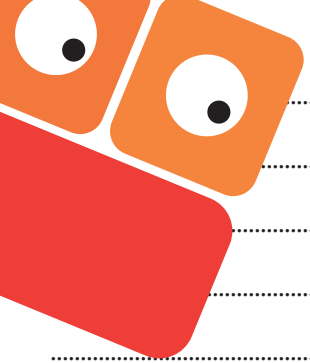
```
bool passwordCheck(char* definedPassword, char* enteredPassword) {
  for (int i = 0; i < 4 ; i++) {
    if (definedPassword[i] != enteredPassword[i])
      return false ;
  }
  return true ;
}


void setup() {
  servo.attach(21);
  Serial.begin(115200);
  servo.write(CLOSED_POSITION);
  Wire.begin();
  oled.init();
  oled.clearDisplay();
  pinMode(pot, INPUT);
  pinMode(ldr, INPUT);
  pinMode(button1, INPUT);
  pinMode(led, OUTPUT);
  oled.setTextXY(0, 4);
  oled.putString("Safe Box");
}

void loop() {
  if (lock_state) {
    if (analogRead(ldr) > LIGHT_THRESHOLD) {
      lockTheSafe();
      lock_state = 0;
      delay(2000);
    }
  }
  else {
    int digit = (analogRead(pot) * 9 / 1023);
    if (digit != oldDigit) {
      oldDigit = digit;
      oled.setTextXY(2, 4);
      oled.putString("Password");
      oled.setTextXY(4, 7);
      oled.putString(String(digit));
    }
```

2

```
  if (digitalRead(button1) == 0 and buttonReleased == 0) {
    Serial.println("Button RELEASED.");
    buttonReleased = 1;
    digitalWrite(led, LOW);
  }
  if (digitalRead(button1) == 1 and buttonReleased == 1) {
    Serial.println("Button PRESSED.");
    buttonReleased = 0;
    digitalWrite(led, HIGH);
    delay(300);
    password[passIndex] = digit;
    digitCounter += 1;
    oldDigit = 0;
    if (passIndex >= 3) {
      passIndex = 0;
      digitCounter = 1;
      if (passwordCheck(&correct_password[0], &password[0])) {
        unlockTheSafe();
        lock_state = 1;
      }
      else {
        oled.clearDisplay();
        oled.setTextXY(2, 2);
        oled.putString("Try Again");
        delay(1500);
        oled.clearDisplay();
      }
    }
    else {
      passIndex += 1;
    }
  }
 }
}
```

Let's define the function that checks whether the password is correct or not and print the value of the potentiometer to the screen continuously.

Let's check the password value determined by the potentiometer when the button is pressed. Let's open the door when the number of correct passwords exceeds 3. If the LDR value is low, that is, the door is closed, let's lock the door.



pico bricks

picobricks.com

GitHub

RobotistanINC
Selim GAYRETLI (Content) - Mehmet Suat MORKAN (Editor) - Elanur TOKALAK (Designer) - Sercan OKAY (Industrial Design)
3 Germany Drive STE 4 #1430 Wilmington, DE 19804
hello@robotistan.com