

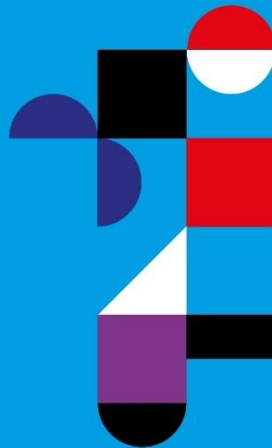


WORLD ROBOT OLYMPIAD™



KMATRIX

FUTURE INNOVATORS SET



sponsored by



Table of Contents

1. Introduction	1
2. MATRIX Mini R4 (MA000)	2
3. RGBLED.....	13
4. OLED.....	14
5. Buzzer	15
6. User Buttons	16
7. IMU	17
8. DC Motors	19
9. DC Motors with Encoder.....	20
10. RC Servo	23
11. Gray Scale Sensor.....	24
12. Potentiometer	27
13. Soil Moisture Sensor	28
14. Water Level Sensor	29
15. Miniature Switch.....	30
16. PIR Sensor	32
17. Temperature Sensor (one wire)	33
18. Temperature and Humidity Sensor.....	34
19. Laser Sensor.....	35
20. Colour Sensor.....	37
21. Gesture Sensor.....	39
22. M-Vision Camera.....	40
23. Packing List	47
24. Additional Details and References	49

1. Introduction

In an exciting collaboration, MATRIX Robotics System and the World Robot Olympiad (WRO) have developed a specialized robotics set crafted for the Future Innovators category. This advanced set features the MATRIX Mini R4 controller, powered by the Arduino UNO R4 Wi-Fi, providing seamless connectivity for enhanced projects. Paired with MATRIXBlock software, a new graphical programming tool based on Scratch, students can transition smoothly from block-based coding to C++, aided by a serial port monitor for easy data debugging. The set includes MATRIX sensors and the MATRIX M-Vision Camera, providing students with the tools to create smart, interactive robotics projects that bring ideas to life. These components enable robots to recognize objects, track movement, and respond to their environment, making them ideal for exploring AI and IoT connectivity. Perfect for both beginners and experienced coders, the set offers a hands-on way to learn about smart technologies, automation, and more. Whether for classroom learning or WRO Future Innovators competition, this versatile set empowers students to unleash their creativity and build the robots of tomorrow!

1.1 Overview

- Arduino® Powered
- Third Party Sensor Support
- Compatible with Easy Assembly
- Dual Programming Platforms
- Industrial Standards
- Ideal for Competitions



2. MATRIX Mini R4 (MA000)

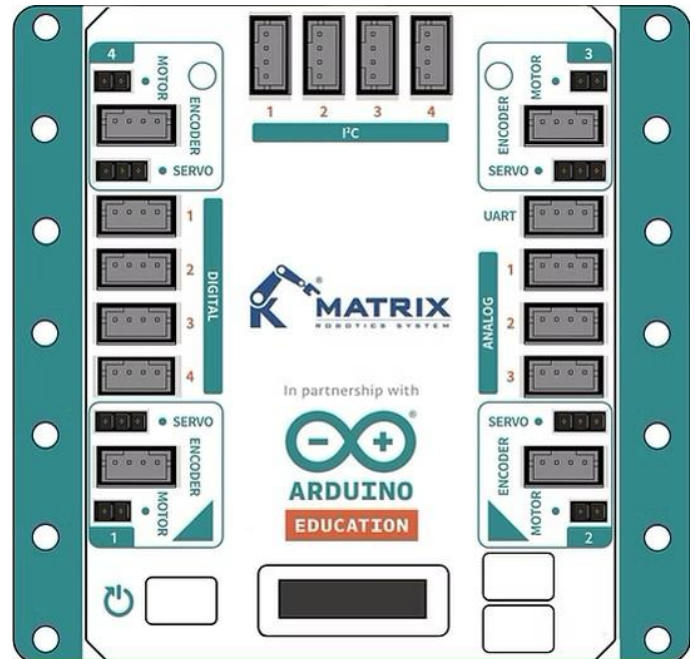
MATRIX Mini R4 is an Arduino UNO R4 WiFi based robot controller. With the MATRIX building system, you can make tons of projects. From basic tracking car to omni-directional mobile platform, you can make any ideas comes out of your mind.

2.1 Feature

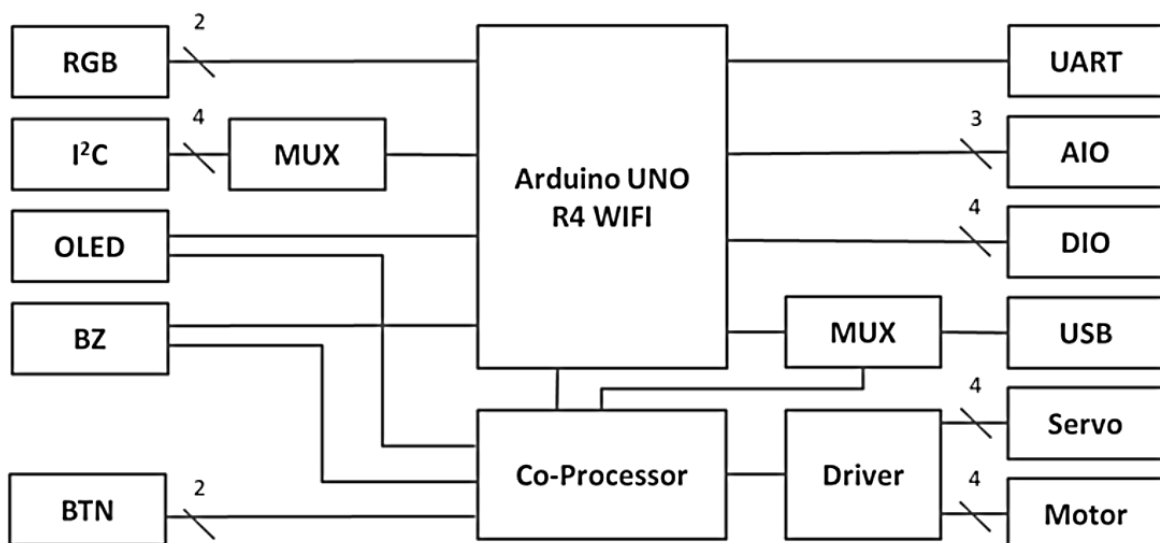
- Support 4 channel RC Servo control.
- Support 4 channel DC Motor with encoder.
- Support 4 channel I2C Interface.
- Support 8 channel GPIO.
- Arduino UNO R4 WiFi built-in.
- OLED, Buttons, RGB LED, Buzzer built-in.
- Co-processor for motor control and IMU.

2.2 Application

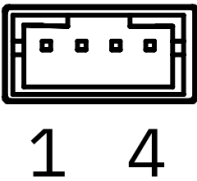
- Autonomous/TelOp Robotics
- IoT Projects Gateway
- Automatic Device



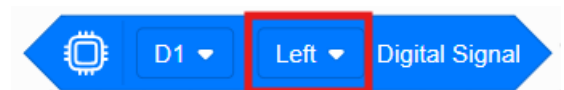
2.3 Block Diagram



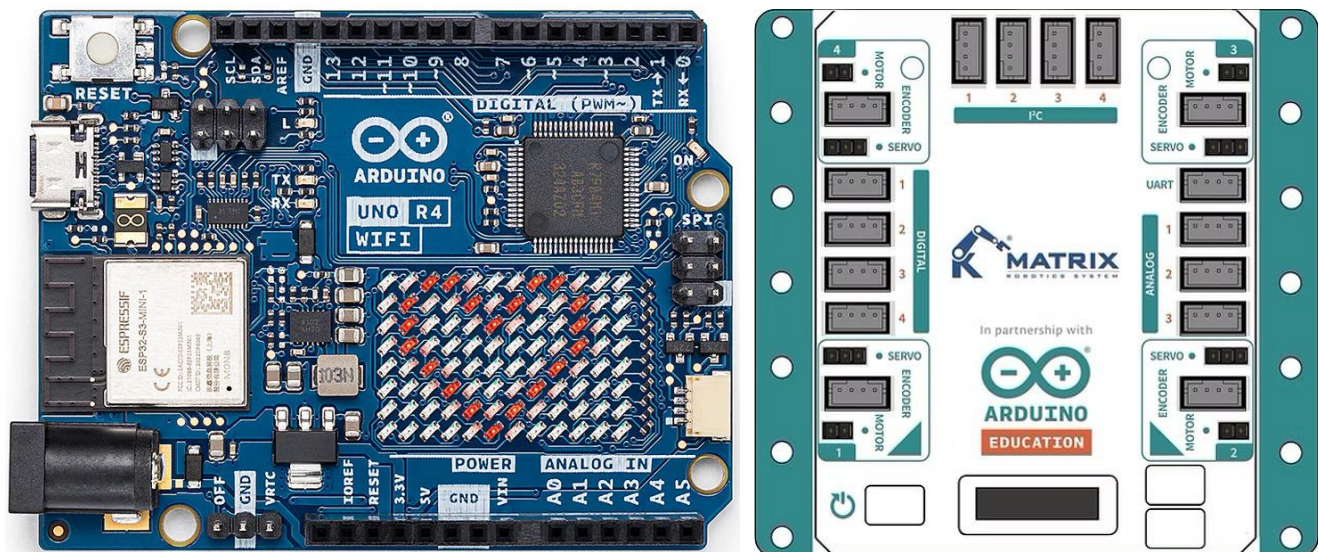
2.4 Matrix Mini R4 Pinout

	Pinout-I2C			
	NO.	Name	I/O	Description
	1	SDA	I/O	Serial data line.
	2	SCL	I	Serial clock line.
	3	VCC	O	Supply voltage.
	4	GND	-	Supply ground.
	Pinout-UART			
	NO.	Name	I/O	Description
	1	TX	O	Serial transmit line.
	2	RX	I	Serial receive line.
	3	VCC	O	Supply voltage.
	4	GND	-	Supply ground.
	Pinout-Analog In			
	NO.	Name	I/O	Description
	1	AINA	I	Analog input A.
	2	AINB	I	Analog input B.
	3	A5V	O	Supply voltage.
	4	GND	-	Supply ground.
	Pinout-Digital I/O			
	NO.	Name	I/O	Description
	1	DIOA	I/O	GPIO A.
	2	DIOB	I/O	GPIO B.
	3	VCC	O	Supply voltage.
	4	GND	-	Supply ground.
	Pinout-Encoder			
	NO.	Name	I/O	Description
	1	CHA	I	CH input A.
	2	CHB	I	CH input B.
	3	M5V	O	Supply voltage.
	4	GND	-	Supply ground.
	Pinout-Servo Out			
	NO.	Name	I/O	Description
	1	GND	-	Supply ground.
	2	5V	O	Supply voltage.
	Pinout-Motor Out			
	NO.	Name	I/O	Description
	1	M-	O	H-bridge out M-.
	2	M+	O	H-bridge out M+.

Note: If you are controlling Analog or Digital pins directly with MATRIXBlock, pin number 1 considered as “Left” pin and pin number 2 considered as “Right” pin.



2.5 Arduino Pin Mapping

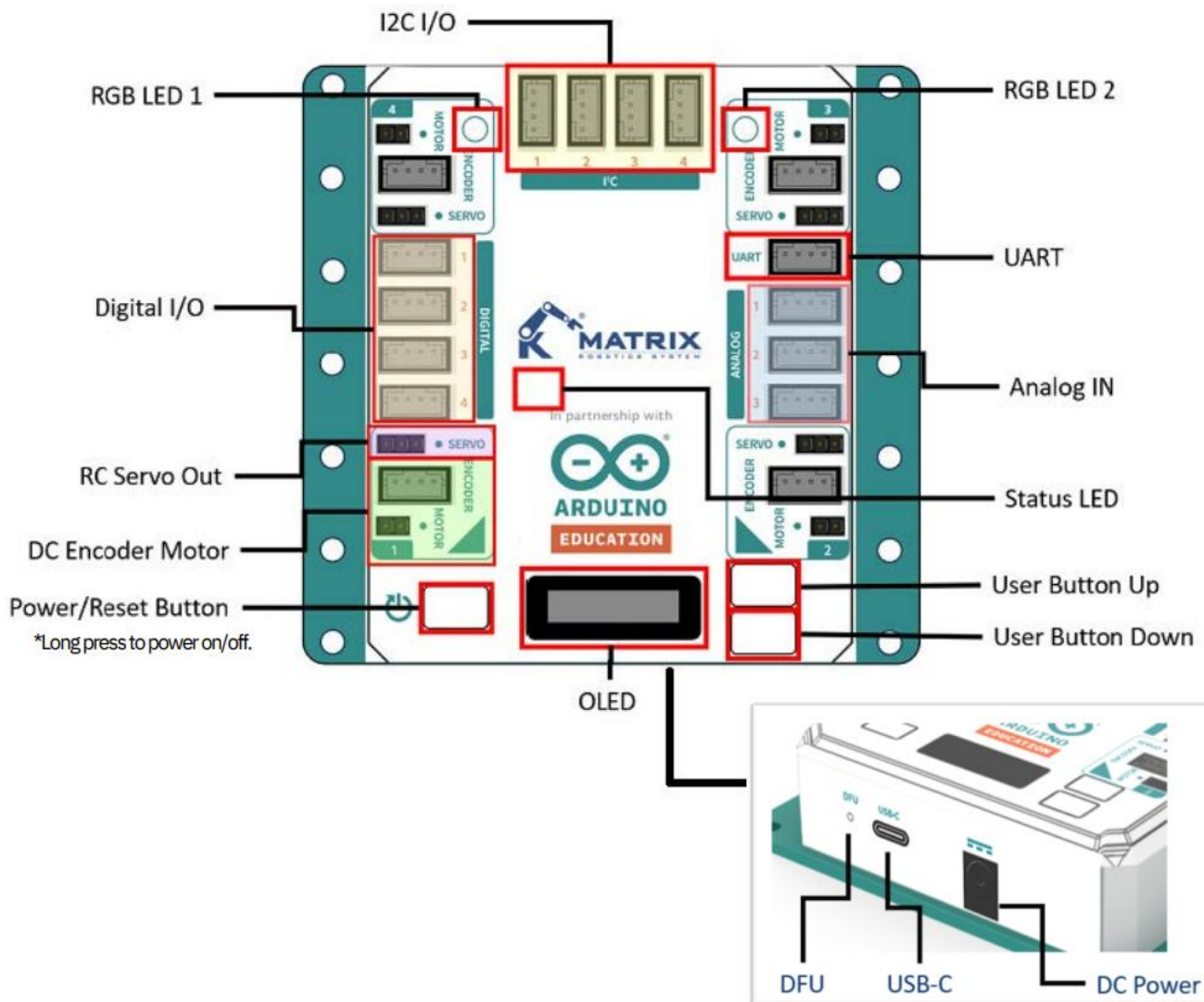


Matrix Mini R4		Arduino Uno R4 WiFi	Peripheral
D1	D1A	3	-
	D1B	2	-
D2	D2A	5	-
	D2B	4	-
D3	D3A	12	-
	D3B	11	-
D4	D4A	13	-
	D4B	10	-
A1	A1A	A1	-
	A1B	A0	-
A2	A2A	A3	-
	A2B	A2	-
A3	A3A	A4	-
	A3B	A5	-
UART	TX	1	-
	RX	0	-
I2C	SDA	-	PCA9548-SDA(0-3)
	SCL	-	PCA9548-SCL(0-3)
Looks	Buzzer	6	
	RGB LED	7	
RC		-	Co-Processor
DC		-	Co-Processor
BTN		-	Co-Processor

2.6 Electrical Characteristics

Parameter	Min	Typ	Max	Units
Input Voltage	6	-	24	V
I/O Voltage	-0.3	5	6.5	V
Digital I/O Pin Current	-	-	8	mA
Analog In Pin Current	-	-	8	mA
RC Servo Output Voltage	-	5	-	V
DC Motor Output Voltage	-	5	-	V
RC Servo Output Current	-	-	1	A
DC Motor Output Current	-	1.5	2	A
UART Buad	300	9600	115200	Bit/s
I2C operating speed	100	-	400	KHz
I2C Low-Level Input Voltage	-0.5V	-	0.33*VCC	-
I2C High-Level Input Voltage	0.7*VCC	-	VCC	-
LED R Wavelength	620	-	625	Nm
LED G Wavelength	522	-	525	Nm
LED B Wavelength	465	-	467	Nm
Operating Temperature	-40	25	85	°C

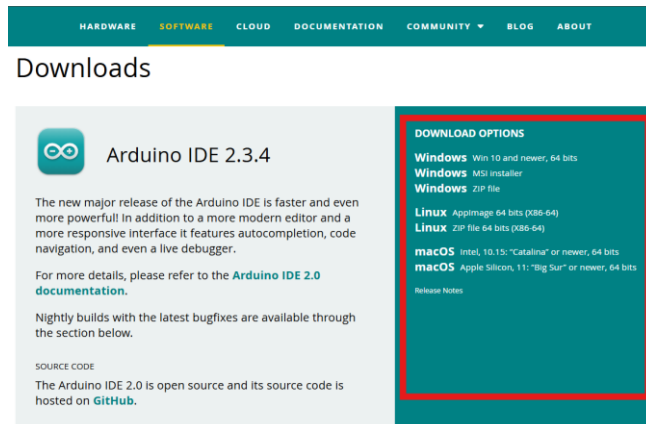
2.7 Hardware guide



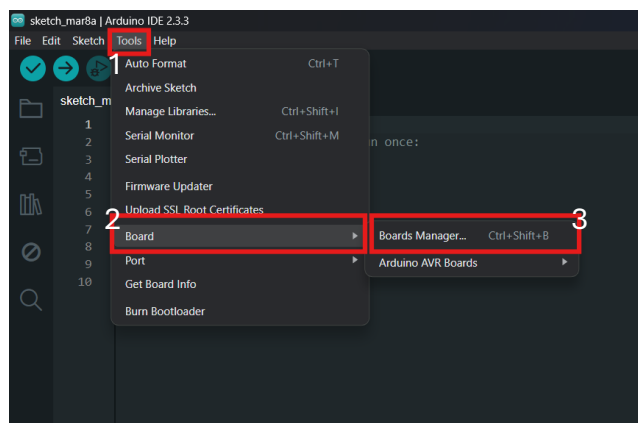
2.8 Software API

2.8.1 Arduino IDE

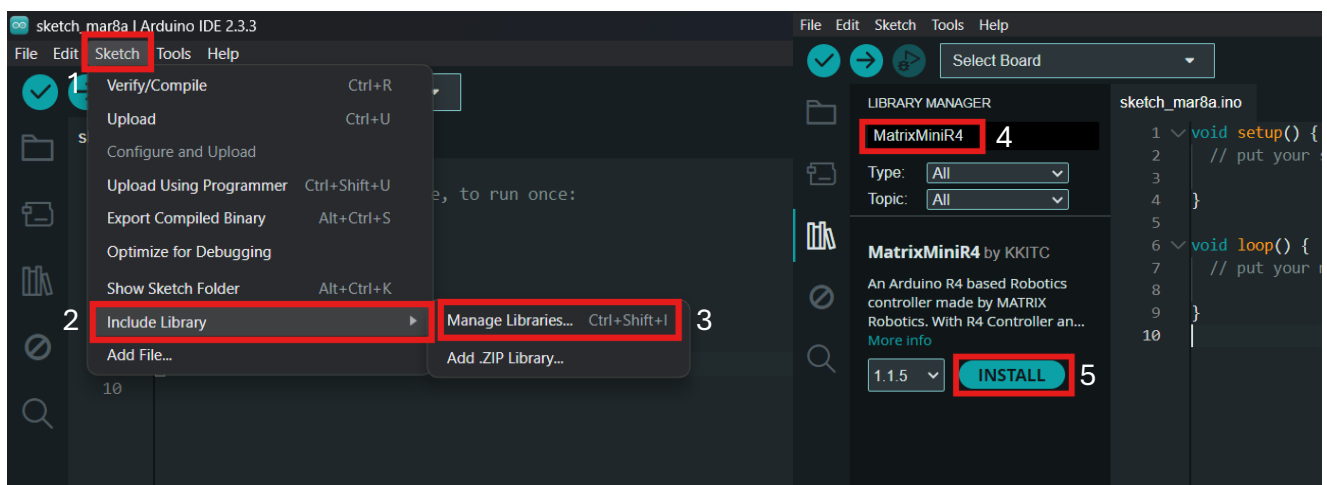
1. Download and install Arduino IDE (At least v2.0) from www.arduino.cc/en/software



2. Open Arduino IDE
3. Open the Boards Manager from the Tools -> Board menu and select "Arduino Uno R4 WiFi"

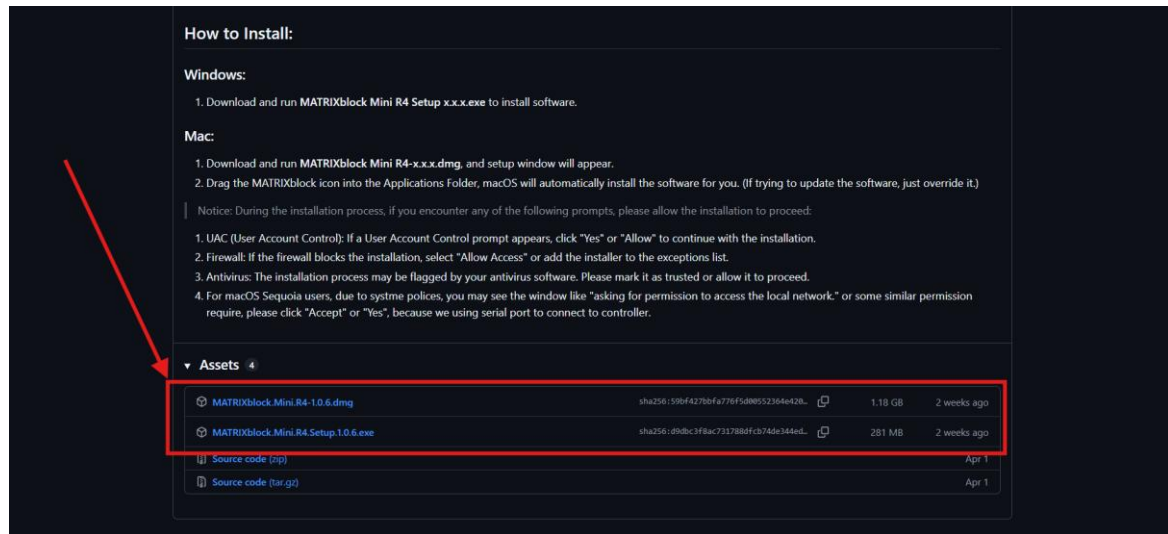


4. Open the Library Manager from the Sketch-> Include Library -> Manage Libraries and search "MatrixMiniR4"



2.8.2 MATRIXBlock

1. Download and install MATRIXBlock from [MATRIX GitHub Repository](#).



2.8.3 Firmware Update

1. While Power On/Reset the Mini R4 Controller, you can see the firmware version on OLED Screen. If firmware is not latest, you might need to upgrade your device to have best experience of Mini R4.



2. Find a paperclip, SIM card pin, small pen or a similar small pin tool.
3. Disconnect the Device from both the battery and USB power.
4. Connect the USB cable to the computer first.

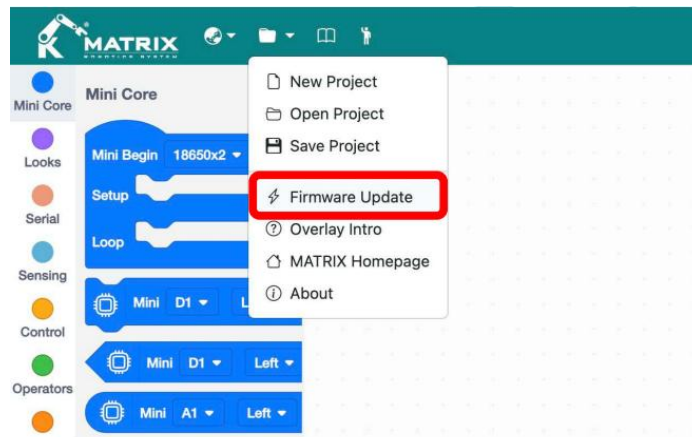
5. Insert the paperclip into the DFU mode pinhole on the Device. (Usually located on the side of the device, nearby USB)



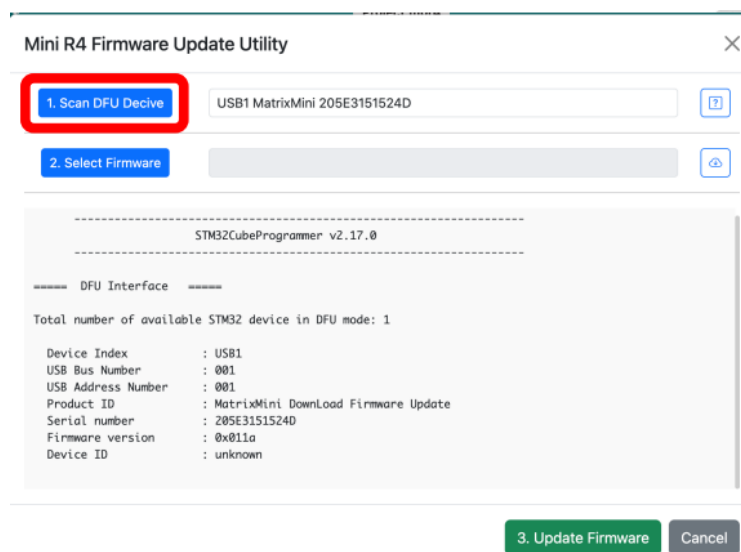
6. While holding the paperclip in the DFU pinhole, plug the other end of the USB cable into the Device.
7. If the blue LED near the pinhole starts flashing, the device has successfully entered DFU mode.



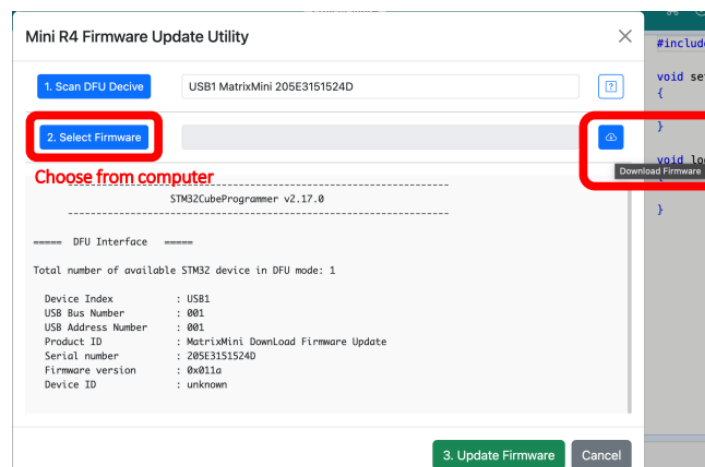
8. Choose the Firmware Update in the MATRIXblock.



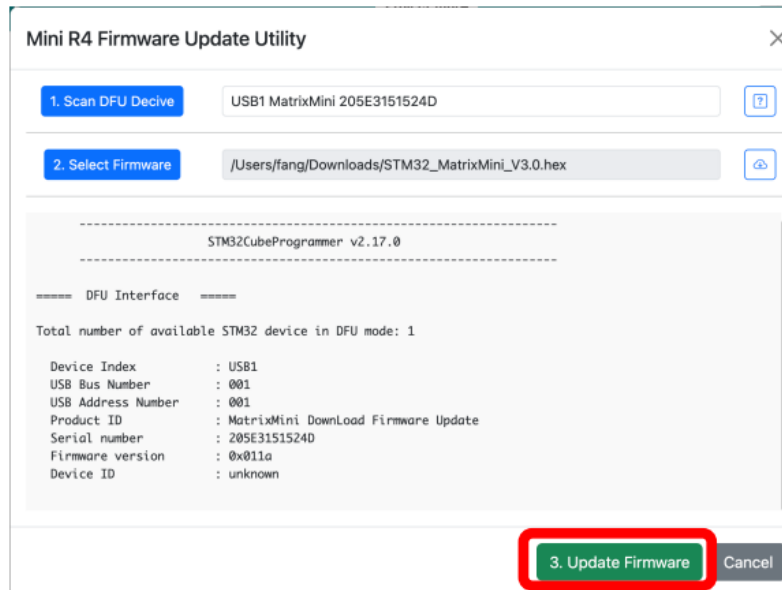
9. Click 1. Scan DFU Device to confirm the device is recognized.



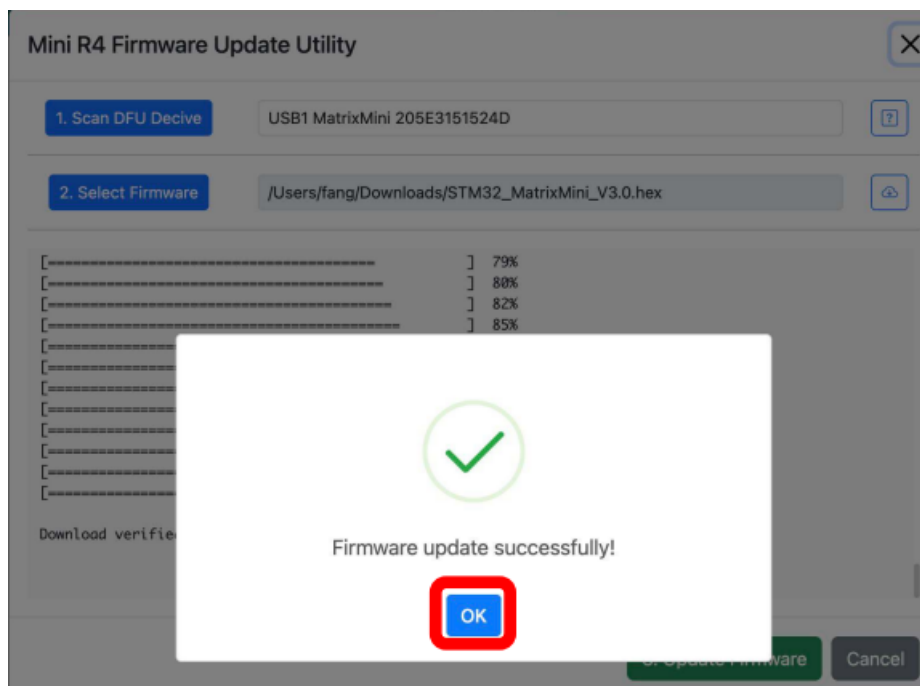
10. Click 2. Select Firmware to select the firmware on your computer or go online to download the latest version.



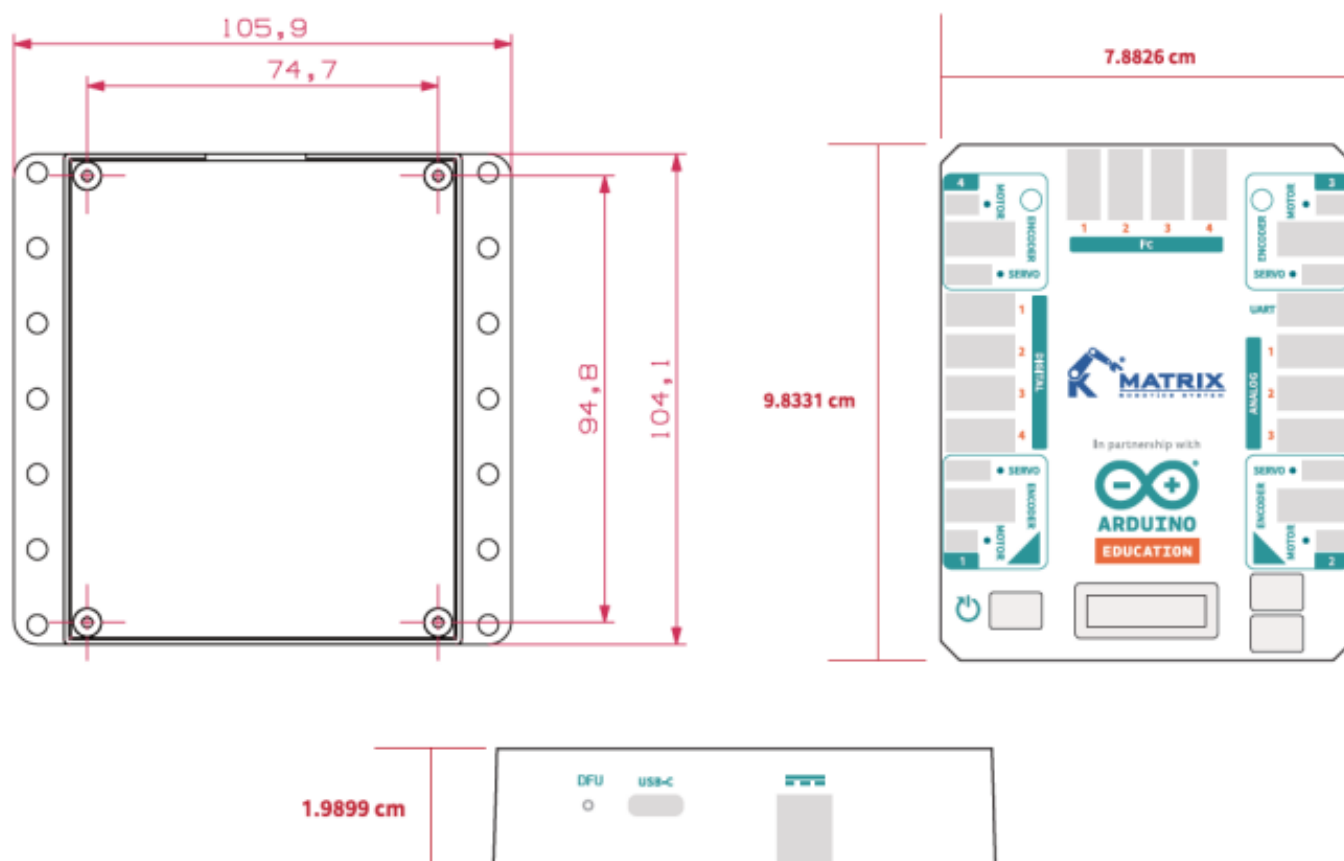
11. After selecting the firmware, click on 3. Update Firmware.



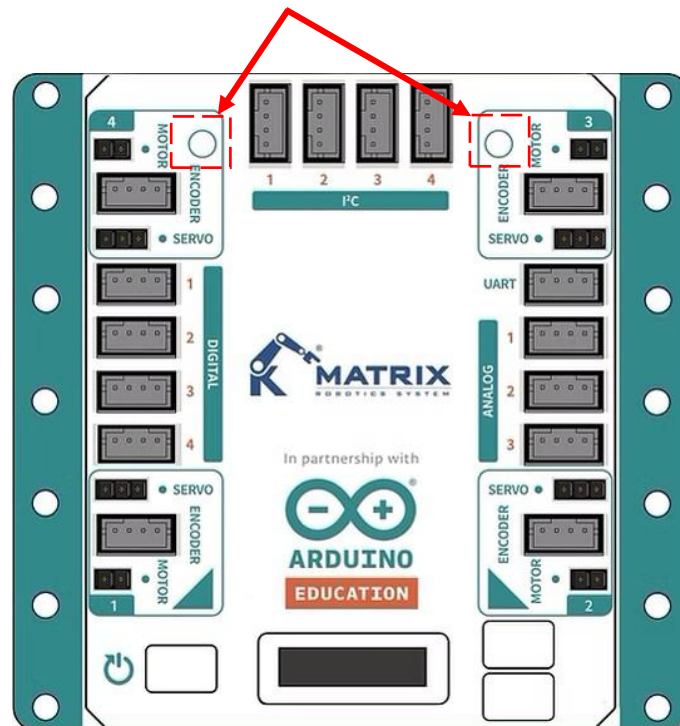
12. After the Firmware update successfully! Dialog comes, Disconnect and reconnect the Mini R4 Controller.



2.9 Dimensions



3. RGBLED



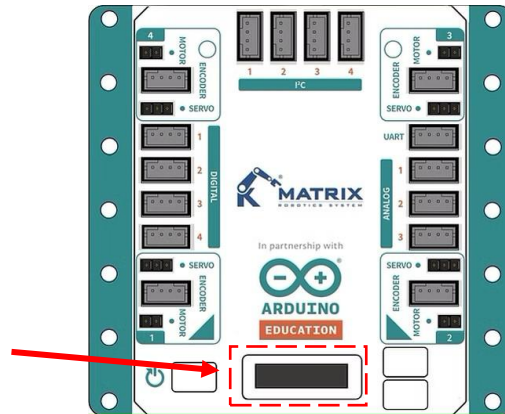
3.1 Sample Code

```

1 #include "MatrixMiniR4.h"
2
3 void setup()
4 {
5     MiniR4.begin();
6     MiniR4.PWR.setBattCell(2);
7     Serial.begin(9600);
8     MiniR4.LED.setColor(1, 255, 0, 0);
9 }
10
11 void loop()
12 {
13
14 }
    
```

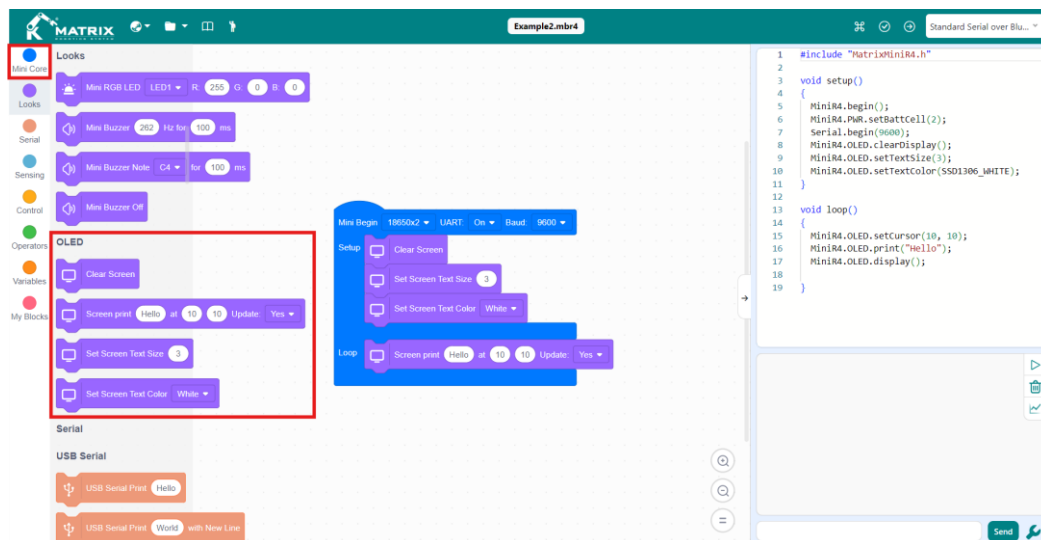
Result: The left LED on the MATRIX Mini R4 goes red.

4. OLED



Note: The OLED display is 128 pixels wide and 32 pixels tall. The two numbers in MatrixBlock simply represent the X (horizontal) and Y (vertical) coordinates on the screen.

4.1 Sample Code

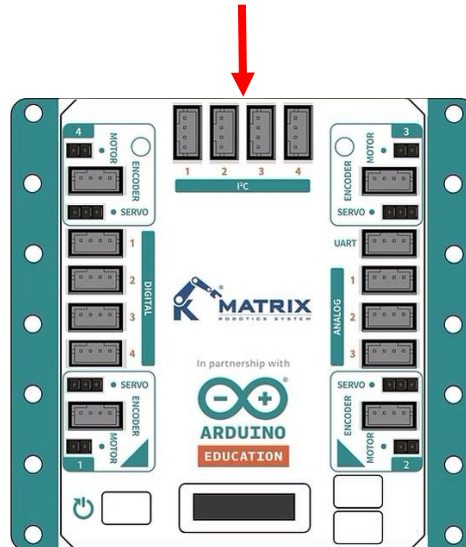


Result: The OLED on MATRIX Mini R4 shows Hello.

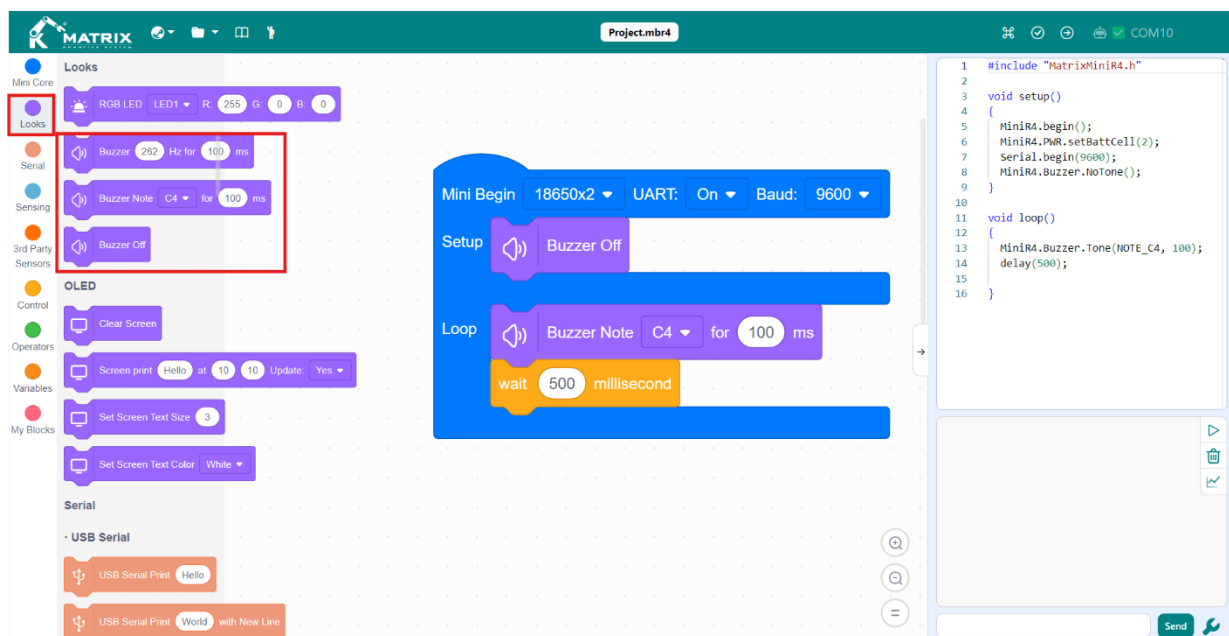
4.2 OLED Blocks Instruction

BLOCK	Instruction
	Clear what is currently displayed on the screen.
	Set the size of the text to be displayed, the smallest is 1, usually placed in the setup. (Default is 1)
	Setting the text colour, usually placed in the setup. (Default is White)
	Print the text to the OLED, you can decide update screen or not for printing multiple content without “flashing”. The screen pixel is 128*32.

5. Buzzer



5.1 Sample Code

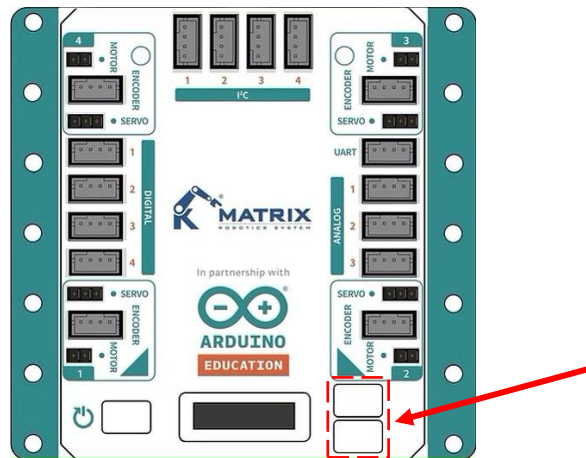


Result: The Buzzer sound goes on.

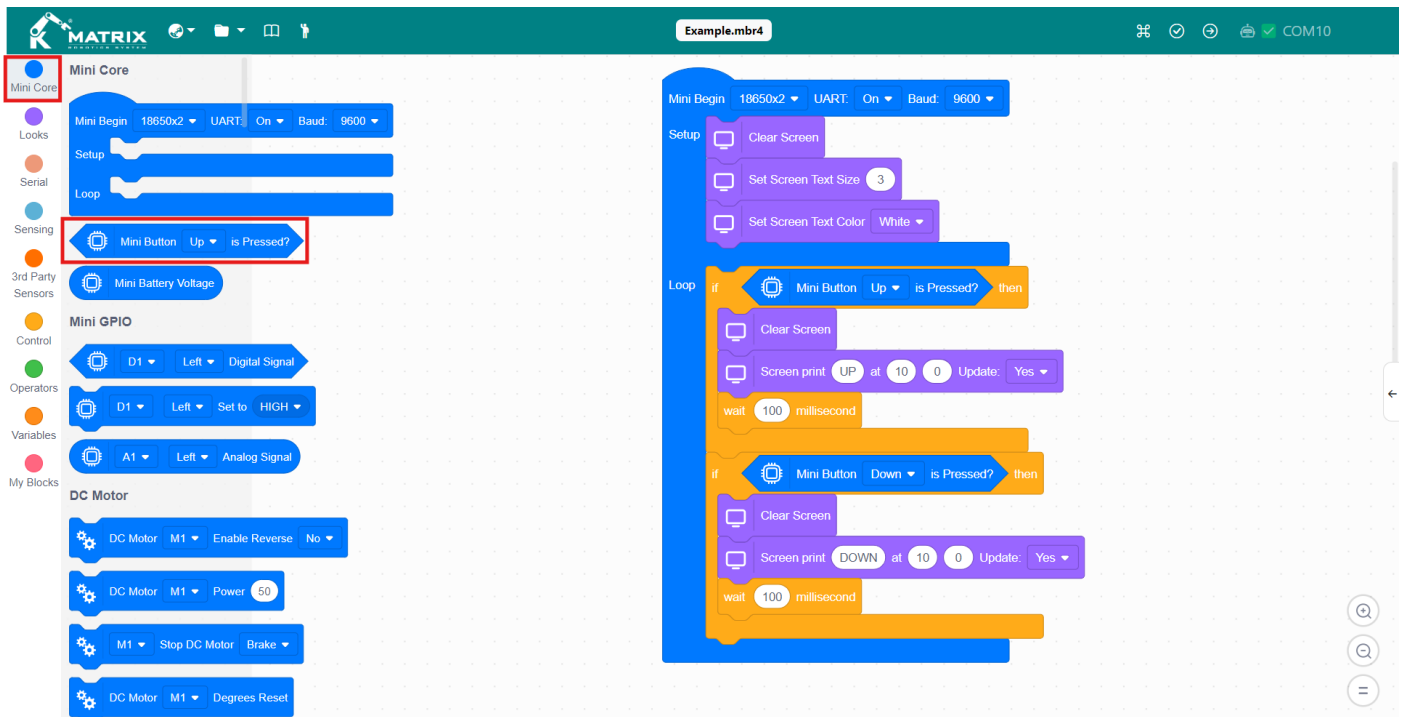
5.2 Buzzer Blocks Instruction

BLOCK	Instruction
	Setting the frequency and playback time of Buzzer, the process will continue in parallel with the program below.
	Setting the chords and playback time of Buzzer, the process will continue in parallel with the program below.
	Stop Playing Sound

6. User Buttons

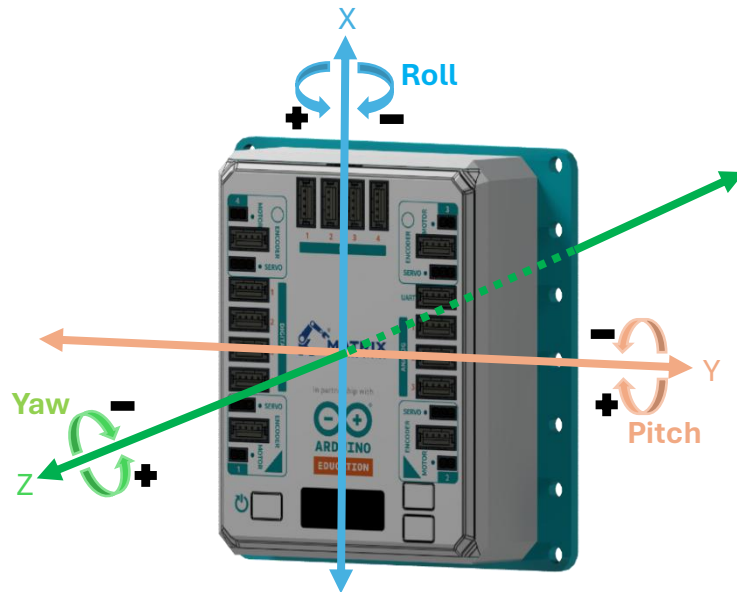


6.1 Sample Code



Result: The OLED will show “UP” and “DOWN” on Button press.

7. IMU



Note: The MATRIX Mini R4 features a built-in IMU (Inertial Measurement Unit) that tracks both movement and orientation in space. It measures acceleration along the X, Y, and Z axes, as well as angular velocity, how quickly the device is rotating, and calculates the angle of rotation around each axis.

Using the acceleration data, you can understand how the device is moving along each axis. The gyroscope measures rotation speed in degrees per second, and by combining this data, the IMU provides Euler angles to describe the device's orientation, how it is tilted or turned.

These angles are:

- **Roll:** Rotation around the X-axis
- **Pitch:** Rotation around the Y-axis
- **Yaw:** Rotation around the Z-axis

Below is an example of rotation around the Z-axis, which causes a change in the yaw value.



7.1 Sample Code

```

1 #include "MatrixMiniR4.h"
2
3 void setup()
4 {
5   MiniR4.begin();
6   MiniR4.PWR.setBattCell(2);
7   Serial.begin(9600);
8   MiniR4.OLED.clearDisplay();
9   MiniR4.OLED.setTextSize(1);
10  MiniR4.OLED.setTextColor(SSD1306_WHITE);
11  MiniR4.Motion.resetIMUValues();
12 }
13
14 void loop()
15 {
16   MiniR4.OLED.setCursor(10, 0);
17   MiniR4.OLED.print(MiniR4.Motion.getAcce
18   MiniR4.OLED.display();
19   MiniR4.OLED.setCursor(10, 10);
20   MiniR4.OLED.print(MiniR4.Motion.getgyro
21   MiniR4.OLED.display();
22   MiniR4.OLED.setCursor(10, 20);
23   MiniR4.OLED.print(MiniR4.Motion.getEule
  
```

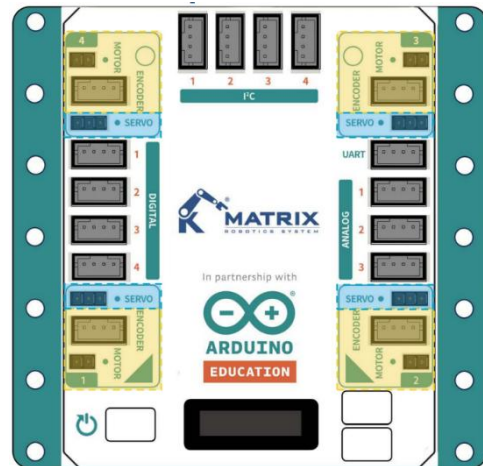
Result: The OLED will display the acceleration on the X-axis, the gyroscope data on the X-axis, and the roll value from the Euler angles.

Note: Whenever the robot powers on or resets, it automatically runs a quick IMU calibration. This only takes about 1.5 seconds, but it's important to keep the robot completely still during that time. Even a small shake or vibration can throw off the calibration and affect how accurately the IMU works.

7.2 Specifications

Parameter	Min	Typ	Max	Units
Supply voltage (VCC)	3	3.3	5	V
Sample rate	-	100	-	Hz
Acceleration measurement range	-4	-	4	g
Angular rate measurement range	-2000	-	2000	dps

8. DC Motors



Note: The dot (•) on the motor port indicates the correct side for connecting the ground (GND) wire, which is usually black or sometimes brown in certain motors or servos. Additionally, DC Motors require an external battery power source to function properly.

8.1 Sample Code

Hardware Connection: Connect the DC Motor to the M1 port of the MATRIX Mini R4.

Sample Code:

```

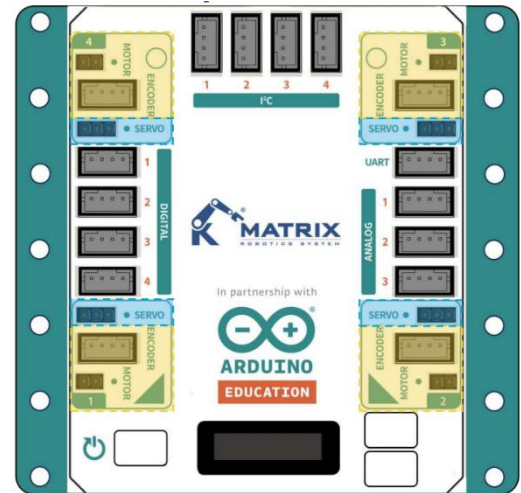
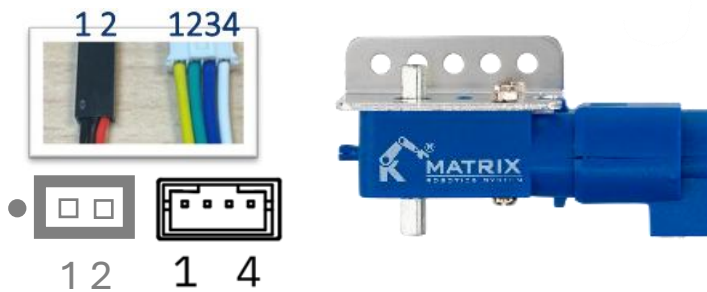
1 #include "MatrixMiniR4.h"
2
3 float DHT11_D1_temp;
4 int DHT11_D1_hum;
5
6 void setup()
7 {
8   MiniR4.begin();
9   MiniR4.PWR.setBattCell(2);
10  Serial.begin(9600);
11  MiniR4.M1.setReverse(false);
12  for(int i_0 = 0; i_0 < 4; i_0++)
13  {
14    MiniR4.M1.setPower(20);
15    delay(2000);
16    MiniR4.M1.setPower(80);
17    delay(1000);
18  }
19  MiniR4.M1.setBrake(true);
20 }
21
22 void loop()
23 {
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

Hardware Connection: Connect the DC Motor to the M1 port of the MATRIX Mini R4.

Result: the motor connected to M1 starts to Move with 50% of speed. After 2 seconds the motor will be move with 80% of speed for 1 second. After 4 times, the motor will be stopped.

9. DC Motors with Encoder



Note: The dot (•) on the motor port indicates the correct side for connecting the ground (GND) wire, which is usually black or sometimes brown in certain motors or servos. Additionally, DC Motors with Encoders require an external battery power source to function properly.

9.1 Sample Code

Hardware Connection: Connect the DC Motor with encoder to the M1 port of the MATRIX Mini R4.

```

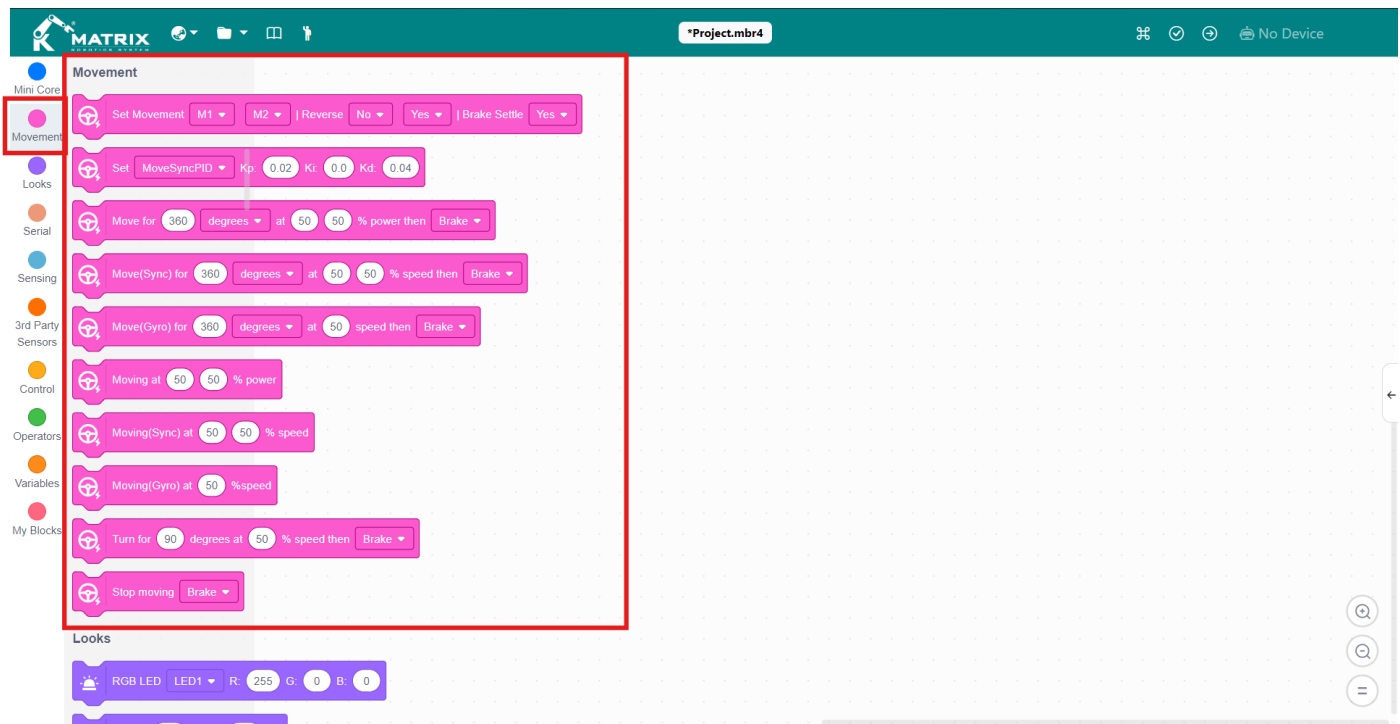
1 #include "MatrixMiniR4.h"
2
3 float DHT11_D1_temp;
4 int DHT11_D1_hum;
5
6 void setup()
7 {
8   MiniR4.begin();
9   MiniR4.PWR.setBattCell(2);
10  Serial.begin(9600);
11  MiniR4.M1.resetCounter();
12  MiniR4.OLED.clearDisplay();
13  MiniR4.OLED.setTextSize(3);
14  MiniR4.OLED.setTextColor(SSD1306_WHITE);
15  MiniR4.M1.setReverse(false);
16  for(int i_0 = 0; i_0 < 4; i_0++)
17  {
18    MiniR4.M1.setPower(20);
19    delay(2000);
20    MiniR4.M1.setPower(80);
21    delay(1000);
22  }
23  MiniR4.M1.setBrake(true);
24
25  // ... (rest of the code)

```

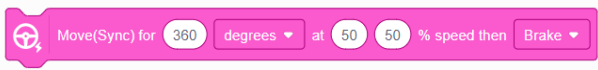
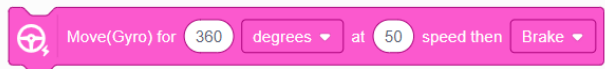
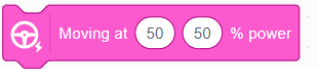
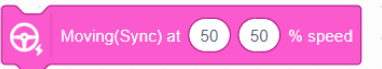
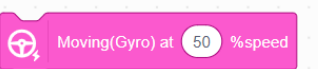
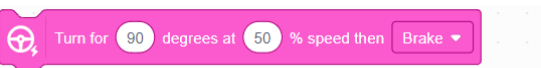
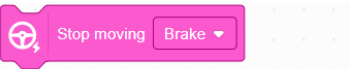
Result: the motor connected to M1 starts to Move with 50% of speed. After 2 seconds the motor will be move with 80% of speed for 1 second. After 4 times, the motor will be stopped and the degrees of rotation will be shown on the OLED.

9.2 Movement Blocks

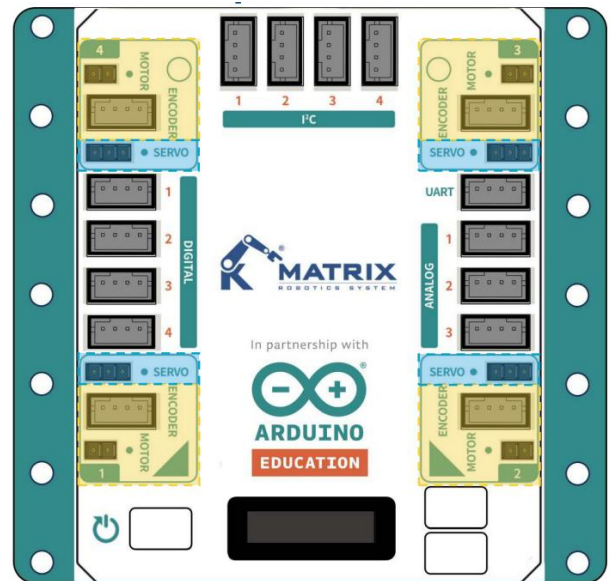
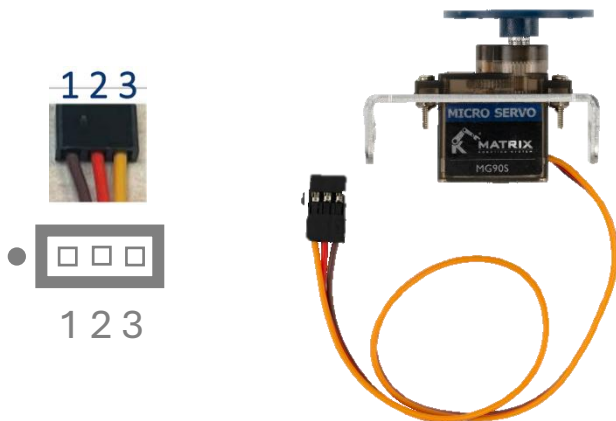
Note: The movement blocks are optimized for two-wheeled robots and work best with motors equipped with encoders, ensuring accurate and reliable motion control. However, if precise rotation in degrees is not required, you can also synchronize movement using a gyro sensor with non-encoder motors.



Block	Function
	This block configures the robot's drive motors by selecting which motors to use, setting their direction, and enabling or disabling brake settle. It ensures that both motors operate correctly and consistently during movement.
	This block defines the PID control values (Kp, Ki, Kd) used to keep both motors synced. These values help the robot correct any differences between the motors, resulting in smoother and more straight movement.
	This block commands the robot to rotate the motors for a specific number of degrees at a set power level and then apply a brake. It allows controlled movement over a fixed distance.

Block	Function
	This block works like “Move for Degrees” but uses PID control to synchronize both motors. It ensures they move at the same speed, allowing the robot to travel in a straight and accurate line before braking to stop.
	This block moves the robot for a specific number of degrees using gyro stabilization. It maintains a straight path by correcting deviations with the gyro sensor, then applies a brake when the movement is complete.
	This block drives the robot continuously at the specified power level for each motor. It does not stop automatically and runs until another command interrupts it.
	This block drives both motors at the same speed with synchronization, helping the robot move in a straight line without using sensors.
	This block drives the robot continuously at the specified speed while using the gyro sensor to keep the heading straight. It runs until another command stops it.
	This block rotates the robot by a set number of degrees at a chosen speed. After completing the turn, the robot can brake to stop.
	This block immediately stops the robot’s movement. Depending on the setting, it can either brake to stop quickly or coast to a smooth stop.

10. RC Servo



Note: The dot (•) on the motor port indicates the correct side for connecting the ground (GND) wire, which is usually black or sometimes brown in certain motors or servos. Additionally, RC Servos require an external battery power source to function properly.

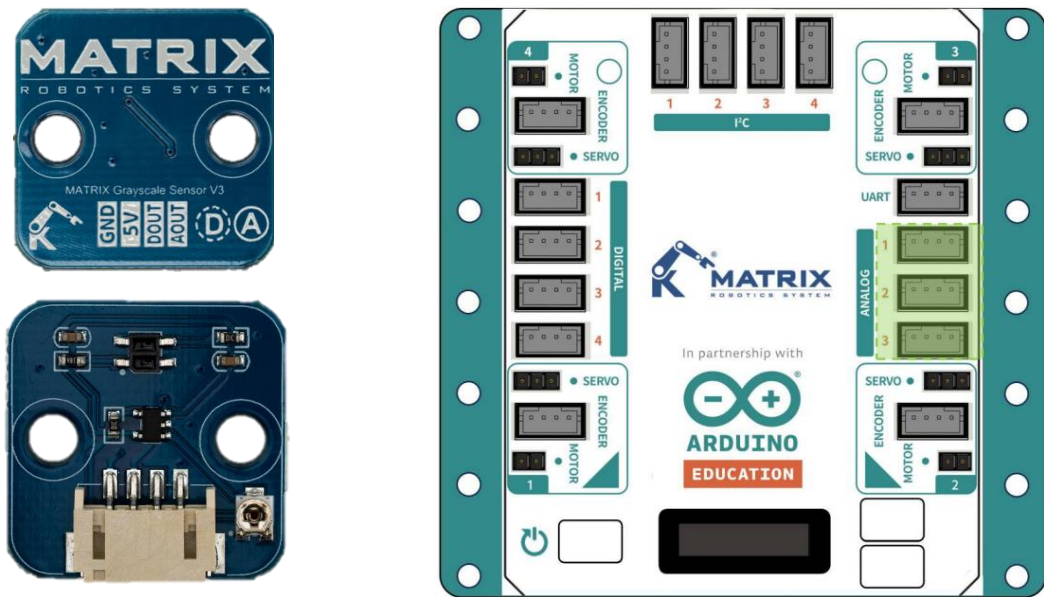
10.1 Sample Code

Hardware Connection: Connect the RC Servo to the M1 port of the MATRIX Mini R4.

The IDE interface shows the 'Mini Core' selected, and the 'RC Servo' block is highlighted in red. The block is configured with 'Servo Motor', 'RC1', 'Enable Reverse' set to 'On', and 'Set Position' set to 90. The 'Loop' block is also visible, with a 'wait 100 millisecond' block."/>

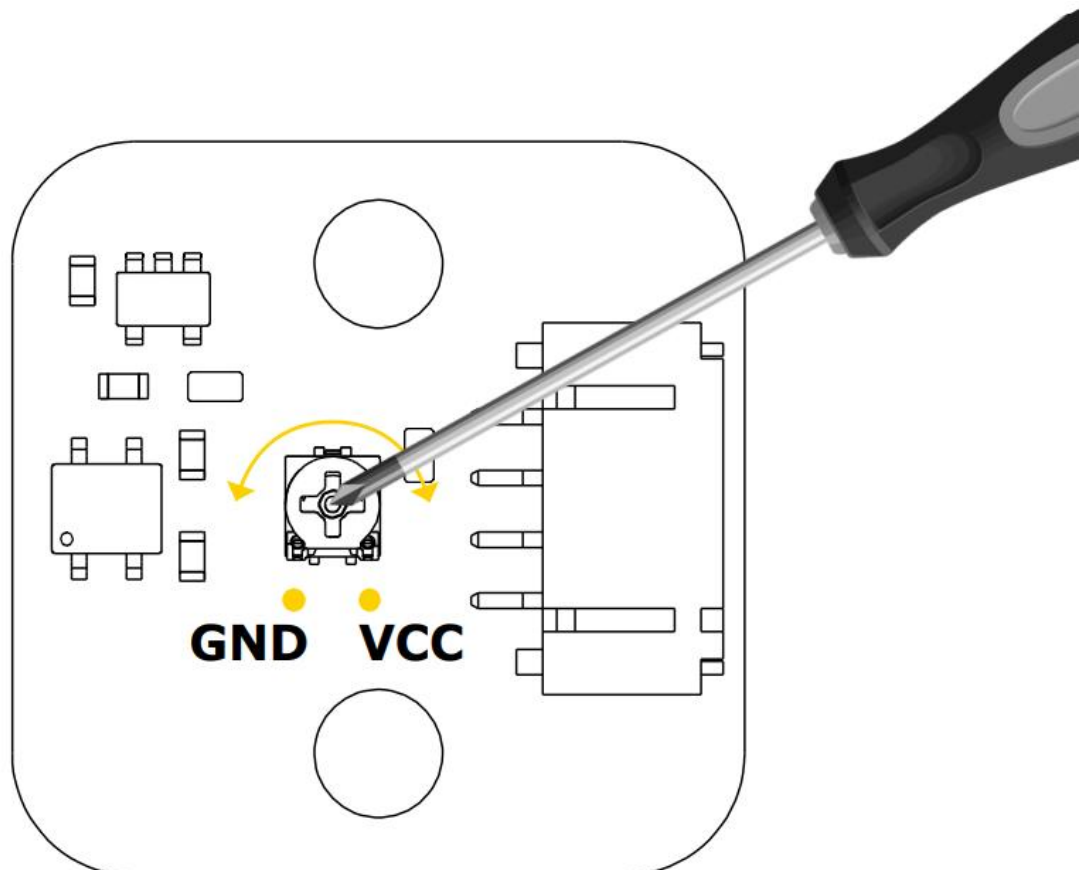
Result: the servo connected to M1 starts to Move and stops on 90°.

11. Gray Scale Sensor



11.1 Usage

AOUT is the IR sensor analog output, the voltage follows IR LED reflection intensity. DOUT only has 0 or 1 state, the changing spot defined by onboard screw. Turn the screw clockwise to bring the gate close to VCC, and turn it counterclockwise to close to GND.



11.2 Sample Code

Hardware Connection: Connect the Gray Scale sensor to the Analog1 port of the MATRIX Mini R4.

The screenshot shows the MBR4 IDE interface with a project named "Example.mbr4". The left sidebar displays various sensor blocks categorized under "Mini Built-in IMU", "Analog Sensors", and "Digital Sensors". The "Sensing" category is highlighted, and the "Grayscale Sensor" block is selected under "Analog Sensors". The main workspace shows a block-based code editor with the following logic:

- Mini Begin:** 18650x2, UART, On, Baud: 9600
- Setup:**
 - Clear Screen
 - Set Screen Text Size: 3
 - Set Screen Text Color: White
- Loop:**
 - Screen print: Grayscale Sensor, A1, Value, at 10, 0, Update: Yes
 - wait: 500 millisecond
 - Clear Screen

The right sidebar shows the corresponding C++ code:

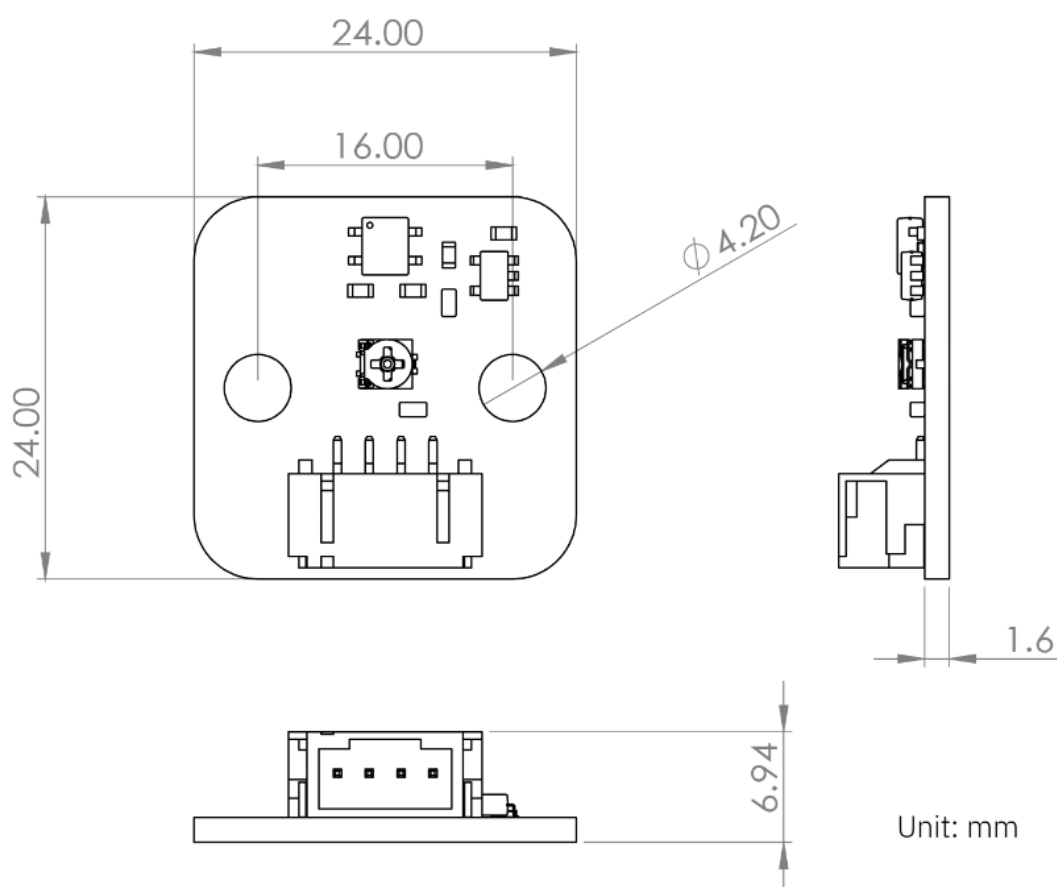
```

1 #include "MatrixMiniR4.h"
2
3 void setup()
4 {
5     MiniR4.begin();
6     MiniR4.PWR.setBattCell(2);
7     Serial.begin(9600);
8     MiniR4.OLED.clearDisplay();
9     MiniR4.OLED.setTextSize(3);
10    MiniR4.OLED.setTextColor(SSD1306_WHITE)
11 }
12
13 void loop()
14 {
15     MiniR4.OLED.setCursor(10, 0);
16     MiniR4.OLED.print(MiniR4.A1.getA1());
17     MiniR4.OLED.display();
18     delay(500);
19     MiniR4.OLED.clearDisplay();
20 }
21
  
```

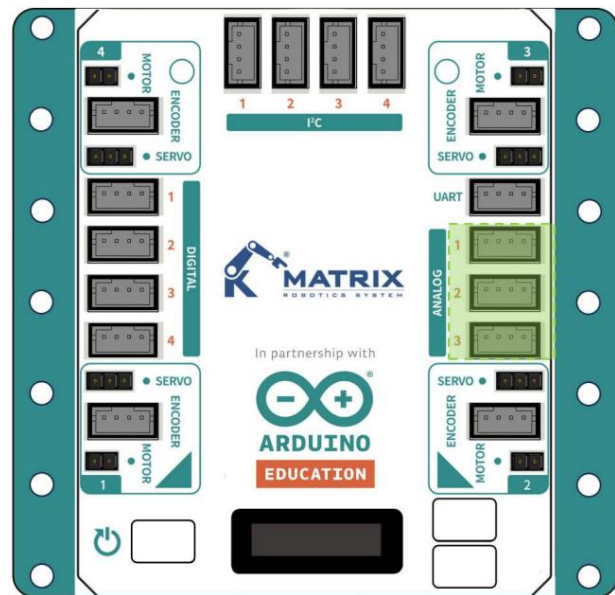
Result: The Gray Scale Sensor value will be shown on the OLED from 0 to 1023. The return value of black is larger and the return value of white is smaller.

11.2 Specification

Parameter	Min	Typ	Max	Units
Supply voltage (VCC)	3	3.3	5	V
Detection range	-	-	50	mm
IR wavelength	-	940	5	nm
Analog output impedance	-	330	-	K Ω
Digital output impedance	-	10	-	K Ω



12. Potentiometer



12.1 Sample Code

Hardware Connection: Connect the Potentiometer to the Analog1 port of the MATRIX Mini R4.

The screenshot shows the MATRIX IDE interface. On the left, the 'Sensing' block is selected in the 'Mini Built-in IMU' category, and the 'Potentiometer' block is selected in the 'Analog Sensors' category. The main workspace displays a block-based program with the following logic:

- Mini Begin:** 18650x2, UART, On, Baud: 9600
- Setup:**
 - Clear Screen
 - Set Screen Text Size: 3
 - Set Screen Text Color: White
- Loop:**
 - Screen print: Potentiometer, A1, Value, at 10, 0, Update: Yes
 - wait: 500 millisecond
 - Clear Screen

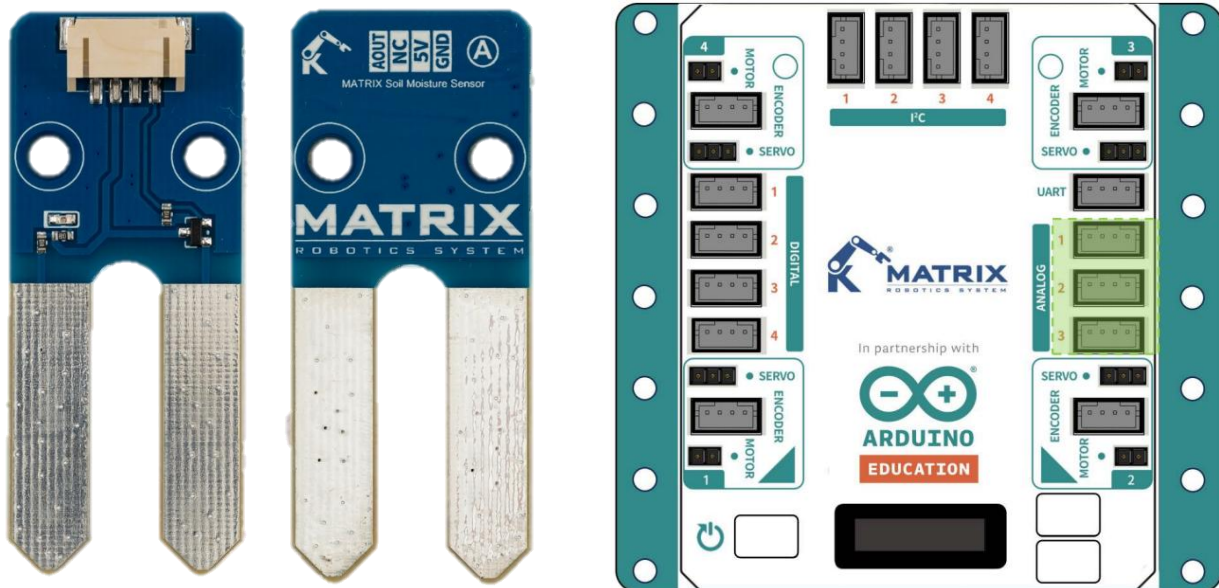
On the right, the corresponding C++ code is shown:

```

1 #include "MatrixMiniR4.h"
2
3 void setup()
4 {
5   MiniR4.begin();
6   MiniR4.PWR.setBattCell(2);
7   Serial.begin(9600);
8   MiniR4.OLED.clearDisplay();
9   MiniR4.OLED.setTextSize(3);
10  MiniR4.OLED.setTextColor(SSD1306_WHITE)
11 }
12
13 void loop()
14 {
15   MiniR4.OLED.setCursor(10, 0);
16   MiniR4.OLED.print(MiniR4.A1.getA1());
17   MiniR4.OLED.display();
18   delay(500);
19   MiniR4.OLED.clearDisplay();
20 }
21
  
```

Result: The Potentiometer value will be shown on the OLED from 0 to 1023.

13. Soil Moisture Sensor



13.1 Sample Code

Hardware Connection: Connect the Soil Moisture Sensor to the Analog1 port of the MATRIX Mini R4.

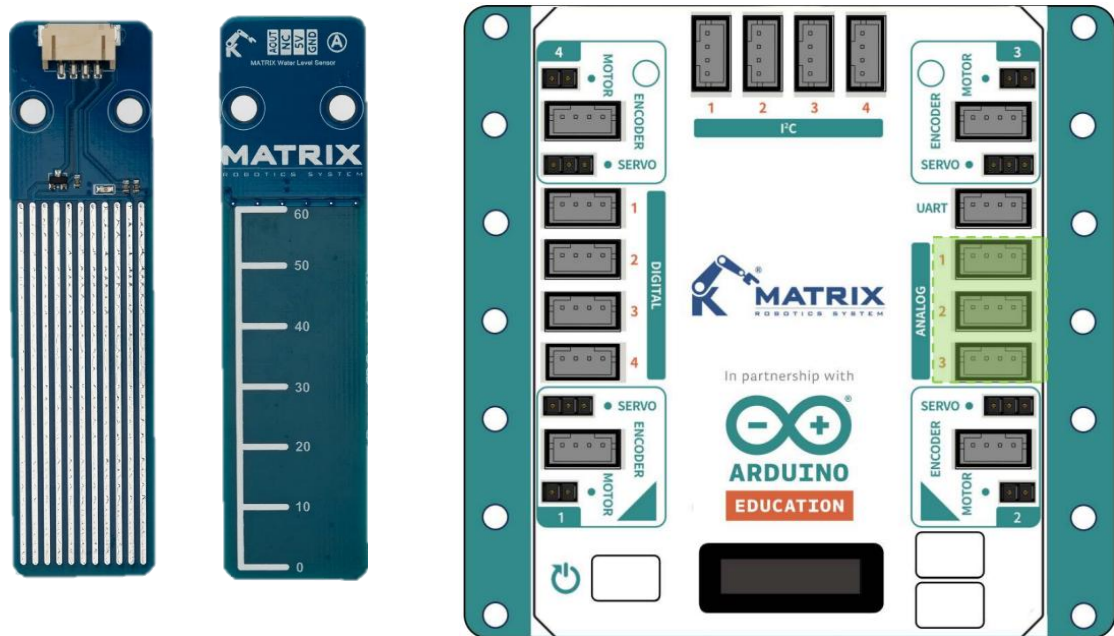
```

1 #include "MatrixMiniR4.h"
2
3 void setup()
4 {
5     MiniR4.begin();
6     MiniR4.PWR.setBattCell(2);
7     Serial.begin(9600);
8     MiniR4.OLED.clearDisplay();
9     MiniR4.OLED.setTextSize(3);
10    MiniR4.OLED.setTextColor(SSD1306_WHITE);
11 }
12
13 void loop()
14 {
15     MiniR4.OLED.setCursor(10, 0);
16     MiniR4.OLED.print(MiniR4.A1.getA1());
17     MiniR4.OLED.display();
18     delay(500);
19     MiniR4.OLED.clearDisplay();
20 }
21

```

Result: The Soil Moisture Sensor value will be shown on the OLED from 0 to 1023 (from low moisture to high moisture).

14. Water Level Sensor



Note: The sensor value is approximately zero when dry, between 300 and 450 when partially submerged, and above 550 when fully submerged.

14.1 Sample Code

Hardware Connection: Connect the Water Level Sensor to the Analog1 port of the MATRIX Mini R4.

```

1 #include "MatrixMiniR4.h"
2
3 void setup()
4 {
5     MiniR4.begin();
6     MiniR4.PWR.setBattCell(2);
7     Serial.begin(9600);
8     MiniR4.OLED.clearDisplay();
9     MiniR4.OLED.setTextSize(3);
10    MiniR4.OLED.setTextColor(SSD1306_WHITE);
11 }
12
13 void loop()
14 {
15     MiniR4.OLED.setCursor(10, 0);
16     MiniR4.OLED.print(MiniR4.A1.getAIL());
17     MiniR4.OLED.display();
18     delay(500);
19     MiniR4.OLED.clearDisplay();
20 }
21

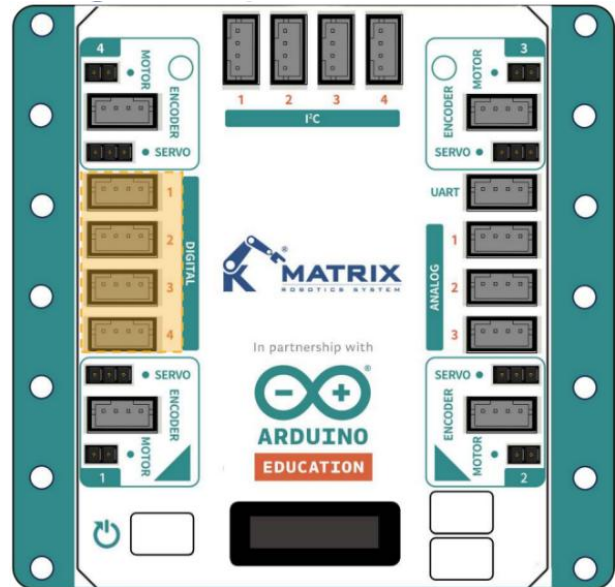
```

The screenshot shows the MATRIX IDE interface with the following components:

- Left Panel:** A sidebar with categories like Mini Core, Looks, Serial, Sensing (highlighted), 3rd Party Sensors, Control, Operators, Variables, and My Blocks. Under Sensing, the 'Water Level Sensor' block is highlighted.
- Main Area:** A block-based programming environment showing a 'Mini Begin' block, a 'Setup' loop with 'Clear Screen', 'Set Screen Text Size' (3), and 'Set Screen Text Color' (White). The main loop contains a 'Screen print' block connected to 'Water Level Sensor A1 Value', followed by a 'wait 500 millisecond' block and a 'Clear Screen' block.
- Right Panel:** A text editor showing the C++ code for the program, which includes the 'MatrixMiniR4.h' header and functions for setup and loop.

Result: The Water Level Sensor value will be shown on OLED from 0 to 1023.

15. Miniature Switch



Note: the OUT pin operates in an inverted manner.

15.1 Sample Code

Hardware Connection: Connect the Miniature Switch to the Digital1 port of the MATRIX Mini R4.

Code:

```

1 #include "MatrixMiniR4.h"
2
3 void setup()
4 {
5     MiniR4.begin();
6     MiniR4.PWR.setBattCell(2);
7     Serial.begin(9600);
8 }
9
10 void loop()
11 {
12     if(MiniR4.D1.get())
13     {
14         MiniR4.Buzzer.Tone(262, 500);
15         delay(100);
16     }
17     else
18     {
19         MiniR4.Buzzer.NoTone();
20     }
21 }
22

```

Block-based Code:

```

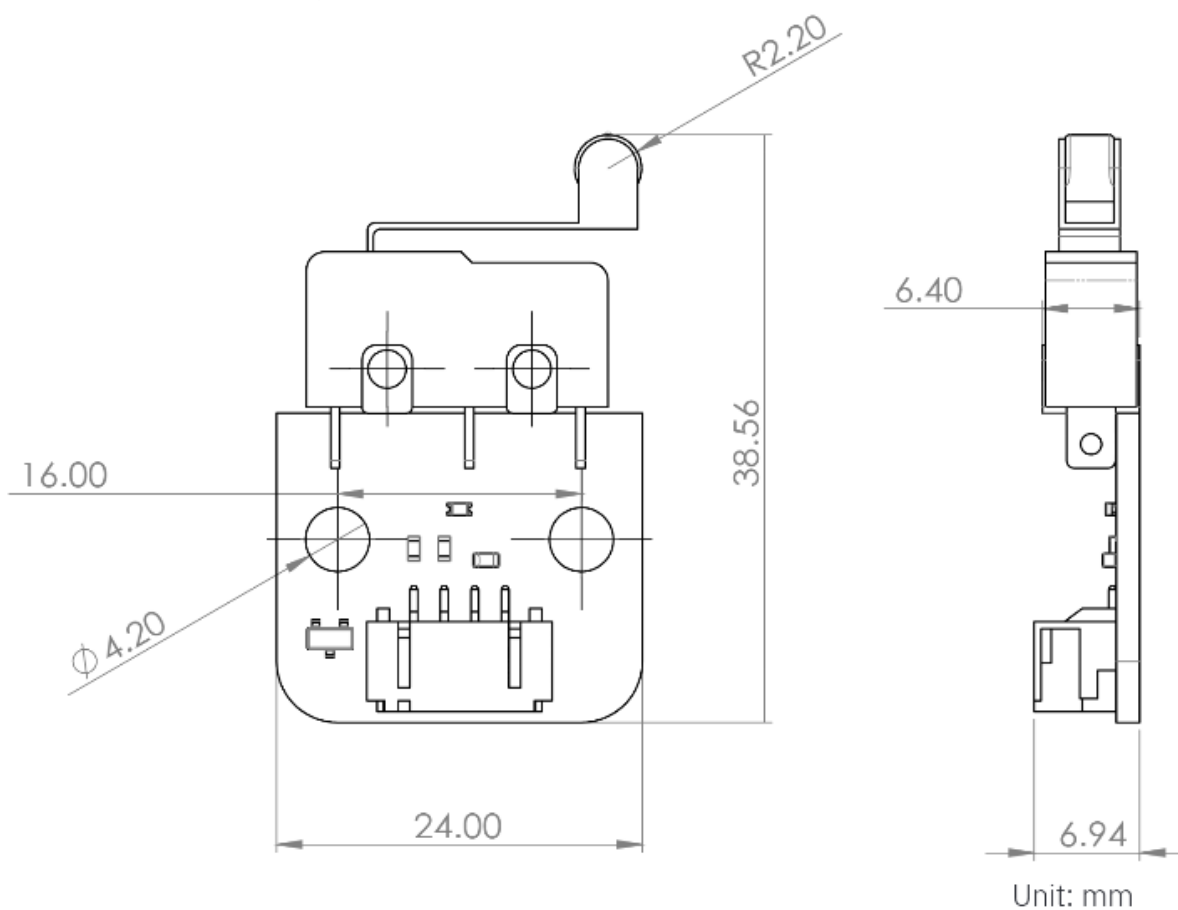
Mini Begin 18650x2 UART: On Baud: 9600
Setup
Loop
  if Miniature Switch D1 Pressed? then
    Buzzer 262 Hz for 500 ms
    wait 100 millisecond
  else
    Buzzer Off

```

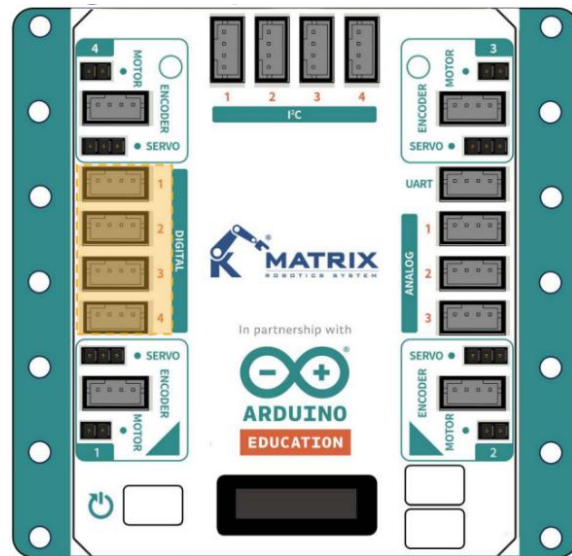
Result: The sound will be played on switch press and will be turned off on switch release.

15.2 Specification

Parameter	Min	Typ	Max	Units
Supply voltage (VCC)	3.3	-	5	V
Operating force	-	-	0.49	N
Switch bounce time	-	-	5	ms



16. PIR Sensor



Note: The sensor detects heat signature changes from a distance of up to 7 meters within a 100-degree field of view.

16.1 Sample Code

Hardware Connection: Connect the Miniature Switch to the Digital1 port of the MATRIX Mini R4.

The screenshot displays the Matrix Mini4 IDE interface. On the left, the 'Sensing' and 'Digital Sensors' categories are highlighted in the sidebar. The main workspace shows a Scratch-style block-based program. The 'Mini Begin' block is configured with '18650x2' for the battery, 'UART: On', and 'Baud: 9600'. The 'Setup' block is empty. The 'Loop' block contains an 'if' statement: 'if PIR Motion Sensor D1 Detected? then' followed by 'Buzzer 262 Hz for 500 ms' and 'wait 100 millisecond', and an 'else' block with 'Buzzer Off'. The right sidebar shows the corresponding C++ code.

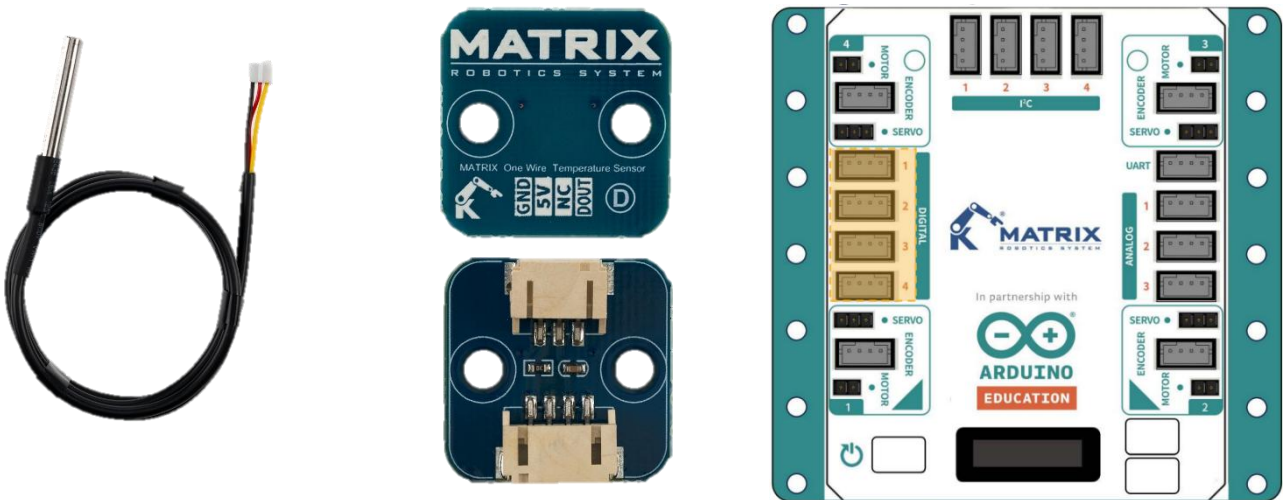
```

1 #include "MatrixMiniR4.h"
2
3 void setup()
4 {
5     MiniR4.begin();
6     MiniR4.PWR.setBattCell(2);
7     Serial.begin(9600);
8 }
9
10 void loop()
11 {
12     if(MiniR4.D1.getL())
13     {
14         MiniR4.Buzzer.Tone(262, 500);
15         delay(100);
16     }
17     else
18     {
19         MiniR4.Buzzer.NoTone();
20     }
21 }
22

```

Result: The sound will be played on motion and will be turned off after no motion.

17. Temperature Sensor (one wire)



Note: The DS18B20 one-wire sensor can measure temperatures from -55°C to $+125^{\circ}\text{C}$.

17.1 Sample Code

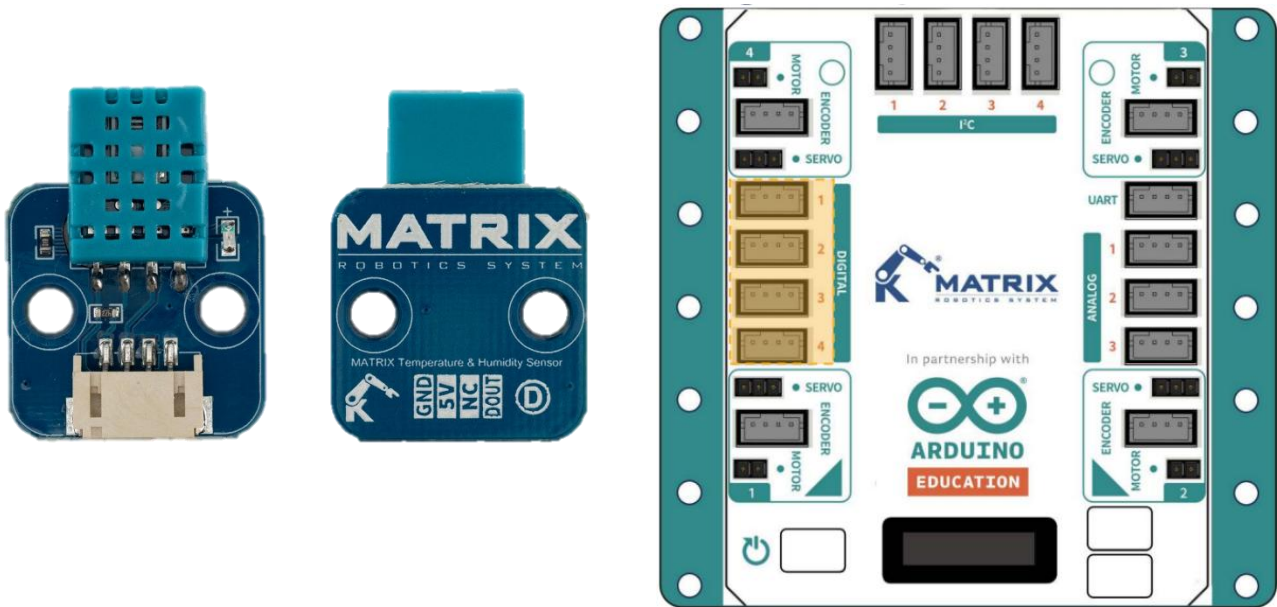
Hardware Connection: Connect the Temperature Sensor (one wire) to the Digital1 port of the MATRIX Mini R4.

```

1 #include "MatrixMiniR4.h"
2
3 void setup()
4 {
5     MiniR4.begin();
6     MiniR4.PWR.setBattCell(2);
7     Serial.begin(9600);
8     MiniR4.OLED.clearDisplay();
9     MiniR4.OLED.setTextSize(3);
10    MiniR4.OLED.setTextColor(SSD1306_WHITE);
11 }
12
13 void loop()
14 {
15     MiniR4.OLED.setCursor(10, 0);
16     MiniR4.OLED.print(MiniR4.D1.MXonewireDT.requestTemperature());
17     MiniR4.OLED.display();
18     delay(500);
19     MiniR4.OLED.clearDisplay();
20 }
21

```

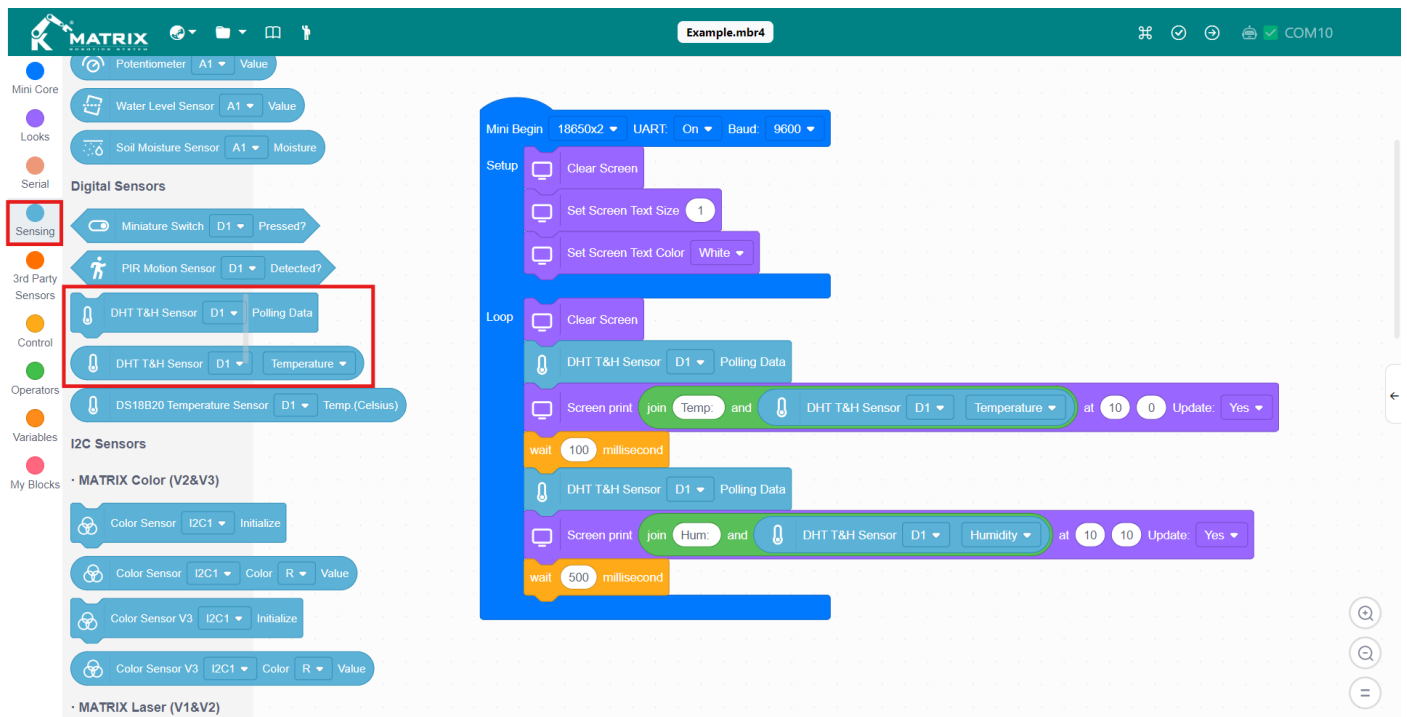
Result: The temperature will be shown on OLED.



Note: The DHT11 sensor has a temperature range of 0°C to 50°C and a humidity range of 20% to 90%.

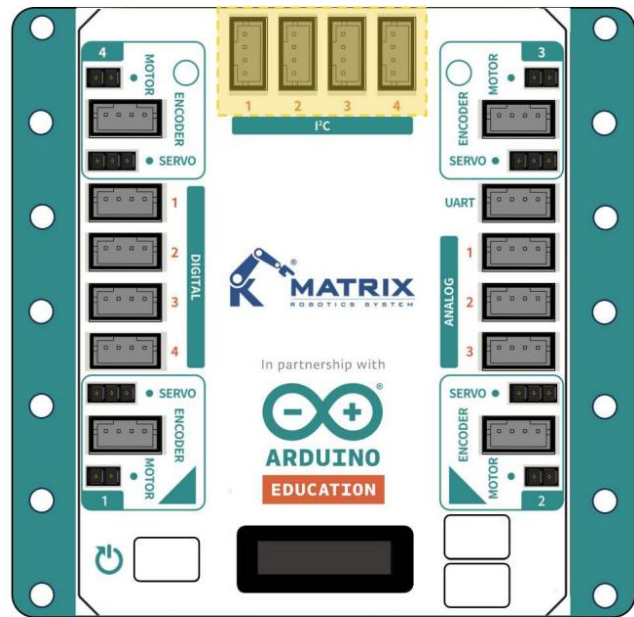
18.1 Sample Code

Hardware Connection: Connect Temperature and Humidity Sensor to the Digital1 port of the MATRIX Mini R4.



Result: The temperature and humidity will be shown on OLED under each other.

19. Laser Sensor



Note: The laser sensor that comes with the MATRIX Future Innovators Set is version 2 (V2). If you're using the older version (V1), just make sure to use the block that's made for the V1 laser sensor so everything works correctly. The V2 can measure up to 2 metres.

19.1 Sample Code

Hardware Connection: Connect Laser Sensor to the I2C1 port of the MATRIX Mini R4.

Block Palette:

- MATRIX Color (V2&V3)
 - Color Sensor I2C1 Initialize
 - Color Sensor I2C1 Color R Value
- Serial
- Sensing
 - MATRIX Laser (V1&V2)
 - Laser Sensor I2C1 Initialize
 - Laser Sensor I2C1 Distance (mm)
 - Laser Sensor V2 I2C1 Initialize
 - Laser Sensor V2 I2C1 Distance (mm)
- MATRIX Gesture
 - Gesture Sensor I2C1 Initialize
 - Gesture Sensor I2C1 Gesture Code
 - Gesture Sensor I2C1 is Gesture 0-None ?
- UART Sensors

Workspace:

- Setup:**
 - Clear Screen
 - Set Screen Text Size 3
 - Set Screen Text Color White
 - Laser Sensor V2 I2C1 Initialize
- Loop:**
 - Screen print Laser Sensor V2 I2C1 Distance (mm) at 10 0 Update: Yes
 - wait 500 millisecond
 - Clear Screen

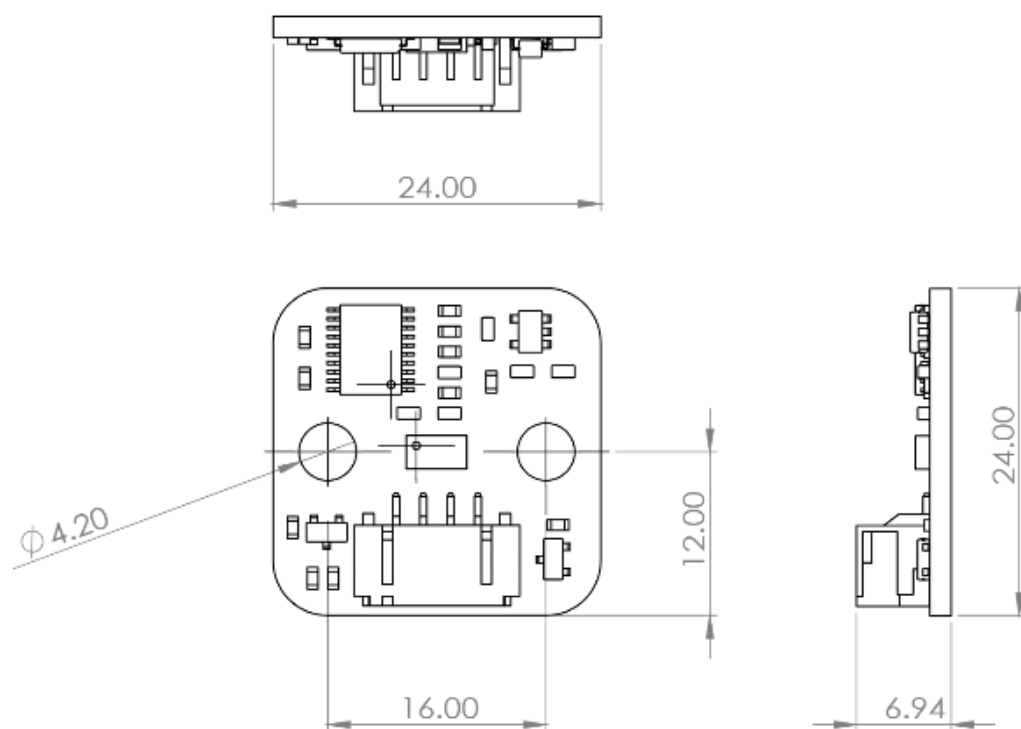
Code Editor:

```

1 #include "MatrixMiniR4.h"
2
3 void setup()
4 {
5     MiniR4.begin();
6     MiniR4.PWR.setBattCell(2);
7     Serial.begin(9600);
8     MiniR4.OLED.clearDisplay();
9     MiniR4.OLED.setTextSize(3);
10    MiniR4.OLED.setTextColor(SSD1306_WHITE);
11    MiniR4.I2C1.MXLaserV2.begin();
12 }
13
14 void loop()
15 {
16     MiniR4.OLED.setCursor(10, 0);
17     MiniR4.OLED.print(MiniR4.I2C1.MXLaserV2.getDistance());
18     MiniR4.OLED.display();
19     delay(500);
20     MiniR4.OLED.clearDisplay();
21 }
22
            
```

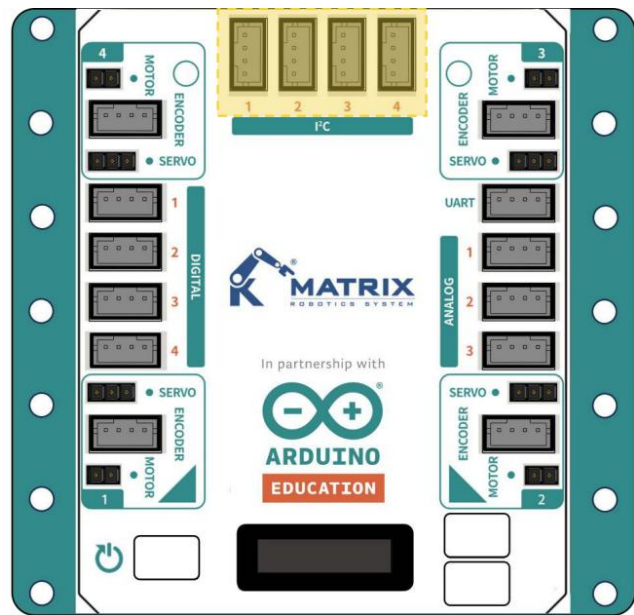
Result: The Laser Sensor value will be shown on the OLED from 50mm to 1200mm.

19.2 Specification



Unit: mm

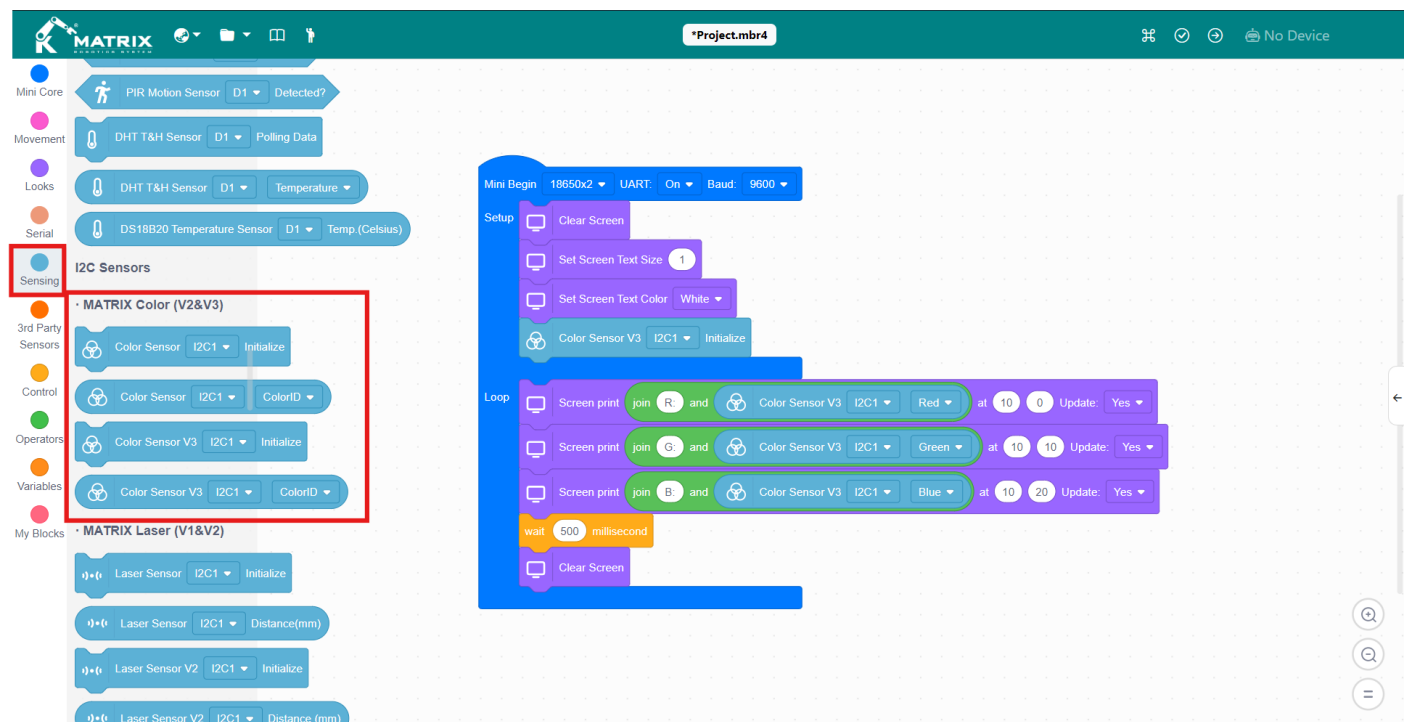
20. Color Sensor



Note: The color sensor included in the Matrix Future Innovators Set is **V3**. It supports both **RGBC** (Red, Green, Blue, Clear) and **HSV** (Hue, Saturation, Value) readings. Additionally, it features a **Color ID** mode, which returns a specific number corresponding to a detected color. In this system, ID 0 represents black, ID 1 violet, ID 3 blue, ID 6 green, ID 7 yellow, ID 9 red, ID 10 white, and ID -1 indicates no color.

20.1 Sample Code

Hardware Connection: Connect the Color Sensor to the I2C1 port of the MATRIX Mini R4.

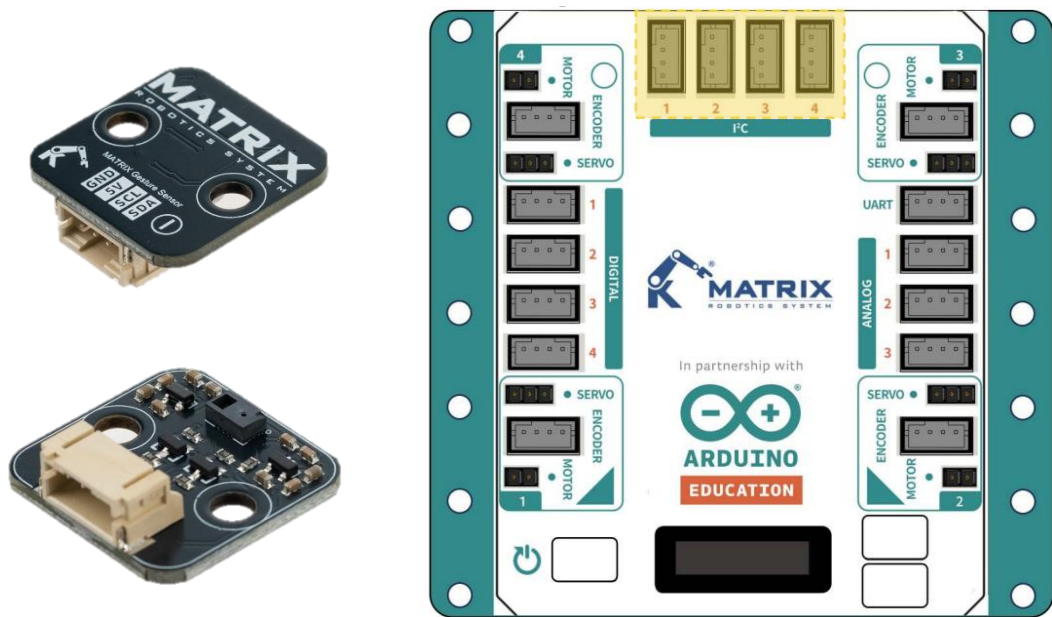


Result: The Color Sensor RGB value will be shown on the OLED under each other.

20.2 Specification

Parameter	Min	Typ	Max	Units
Supply voltage (VCC)	3	3.3	5	V
Red channel wavelength	-	465	-	Nm
Green channel wavelength	-	525	-	Nm
Blue channel wavelength	-	615	-	Nm
I2C operating speed	100	-	400	KHz
I2C low-level input voltage	-0.5V	-	0.33*VCC	-
I2C high-level input voltage	0.7*VCC	-	VCC	-

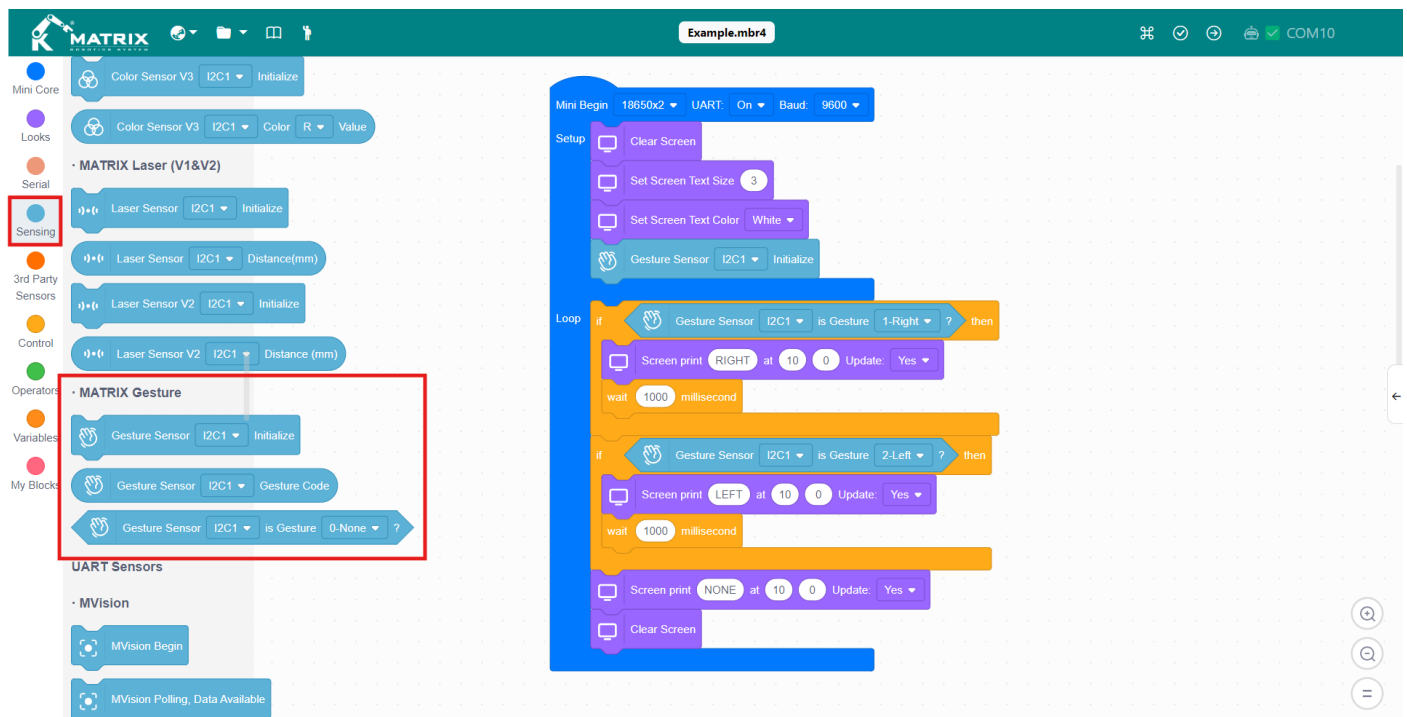
21. Gesture Sensor



Note: Gesture sensor values are assigned as follows: 0 for None, 1 for Right, 2 for Left, 3 for Up, 4 for Down, 5 for Forward, 6 for Backward, 7 for Clockwise, 8 for Anticlockwise, 9 for Wave. It can detect motion from 5 to 15 centimeters far from sensor.

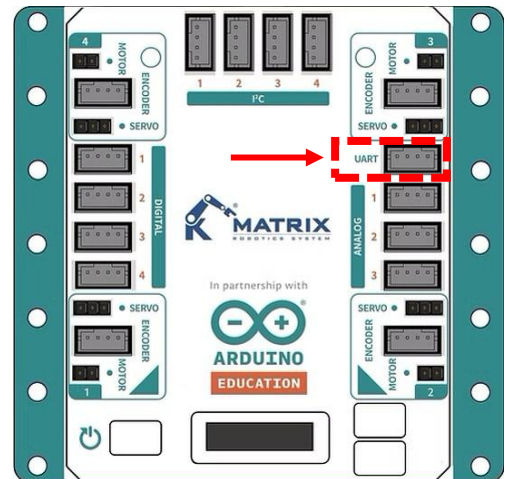
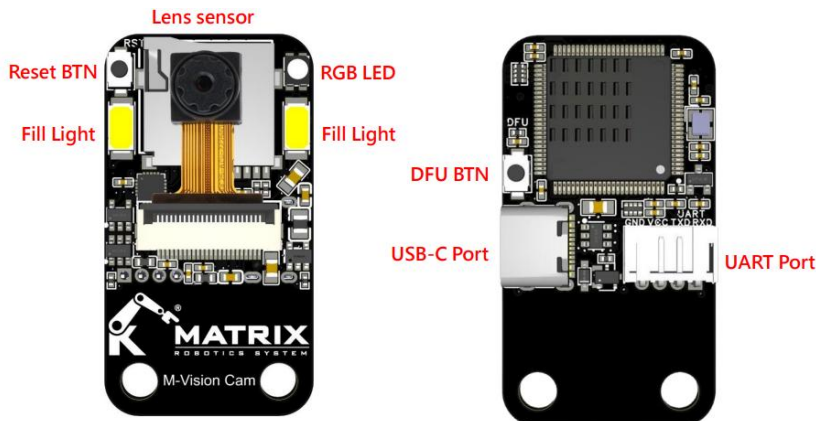
21.1 Sample Code

Hardware Connection: Connect the Gesture Sensor to the I2C1 port of the MATRIX Mini R4.



Result: The OLED will show RIGHT and LEFT on hand gestures.

22. M-Vision Camera



22.1 Overview

- AI camera based on Open MV solutions
- Color and Object detection
- STM32H7 Processor
- UART and Type- C port

22.2 Setup and Operating Environment for OpenMV IDE

1. Visit the official OpenMV website to download and install the version suitable for your operating system on <https://openmv.io/pages/download>.

OpenMV IDE v4.4.7

OpenMV IDE is the premier integrated development environment for use with your OpenMV Cam. It features a powerful text editor, debug terminal, and frame buffer viewer with a histogram display. OpenMV IDE makes it easy to program your OpenMV Cam.

[OpenMV IDE v4.4.7 - Release Notes](#)

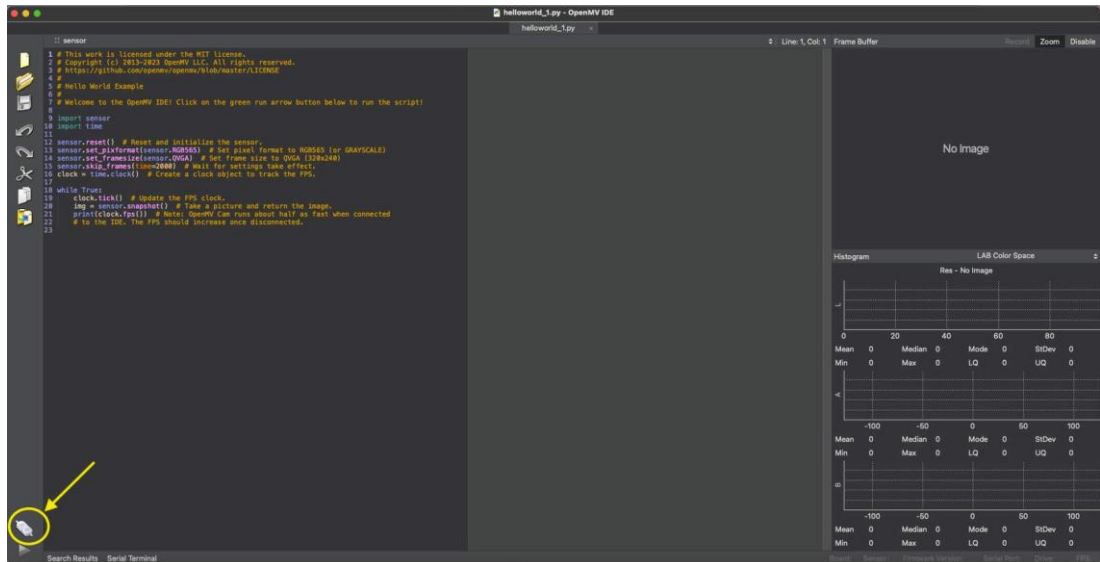
Download Now

Installer EXE For Windows 7, 8, 10, 11, or later	Installer DMG For macOS Monterey or later	Installer RUN For Ubuntu 20.04 LTS 64-bit or later
Installer ZIP For Windows 7, 8, 10, 11, or later	Installer TAR.GZ For Raspberry Pi OS 64-bit only	Installer TAR.GZ For Ubuntu 20.04 LTS 64-bit or later

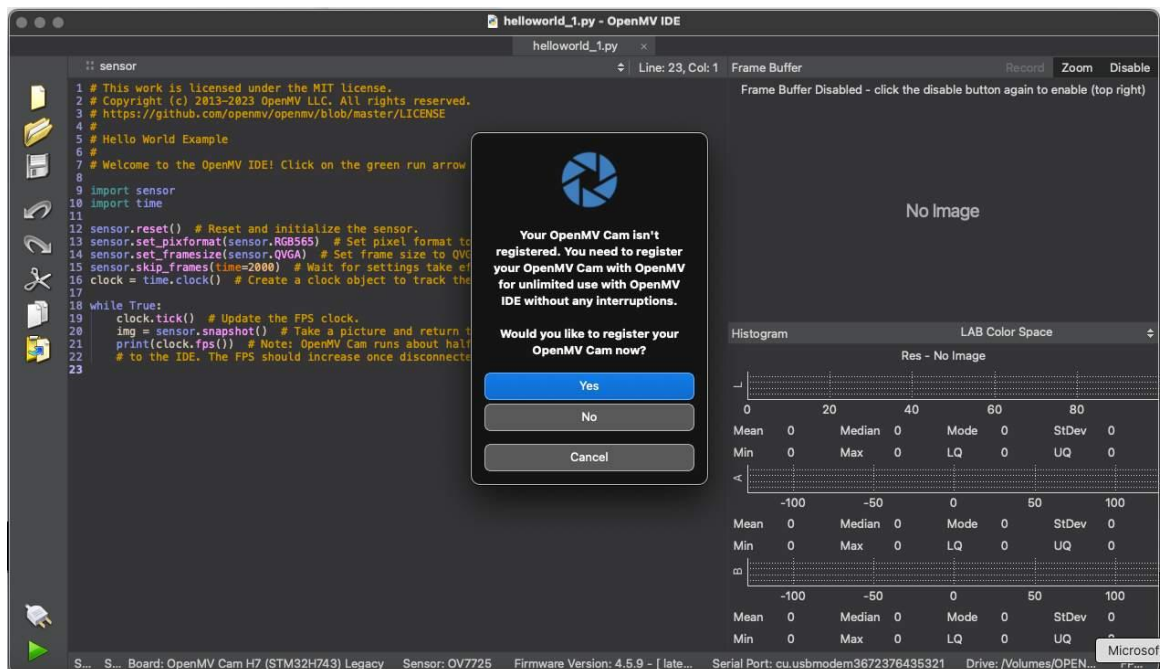
- After installation is complete, open the OpenMV IDE.
- Use the USB-A to USB-C cable included in the package to connect your computer to M-Vision Camera. The LED will light up green for a few seconds after the connection is established and then turn off.

Important Note: Must use USB-A to USB-C cable to connect M-Vision Camera to the Computer.

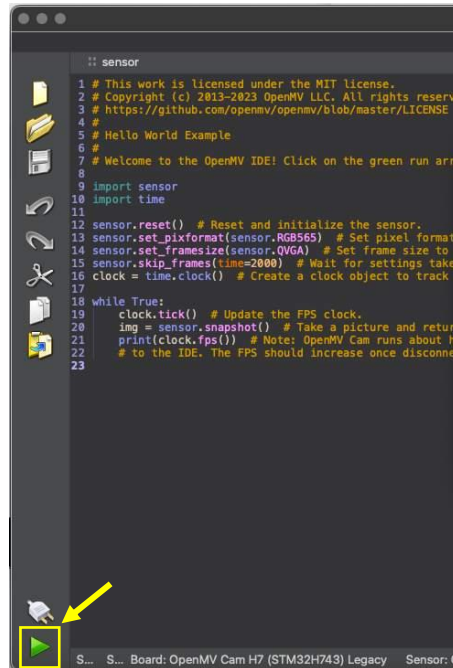
- Click the connect button in the lower left corner of the OpenMV screen.



- After successfully connecting M-Vision to the computer, a window will appear asking if you want to register this Camera three times in a row. Just select No or Cancel.

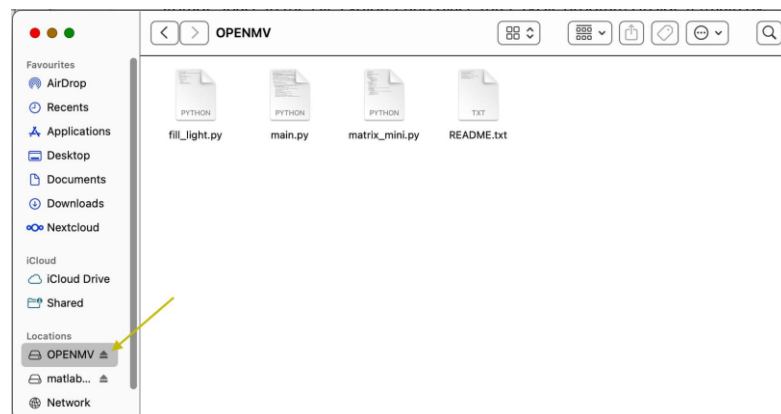


- When the OpenMV IDE launches, it automatically displays a sample code named "helloworld_1.py". If it doesn't appear, you can find it under the File > Examples menu. More examples will become accessible once the camera is connected as well.
- When the camera is connected, the Start (Run Script) button, located in the lower-left corner of the screen, will become active. By clicking it, you can run your first code and view the camera's output. Click the same button again to stop the code.

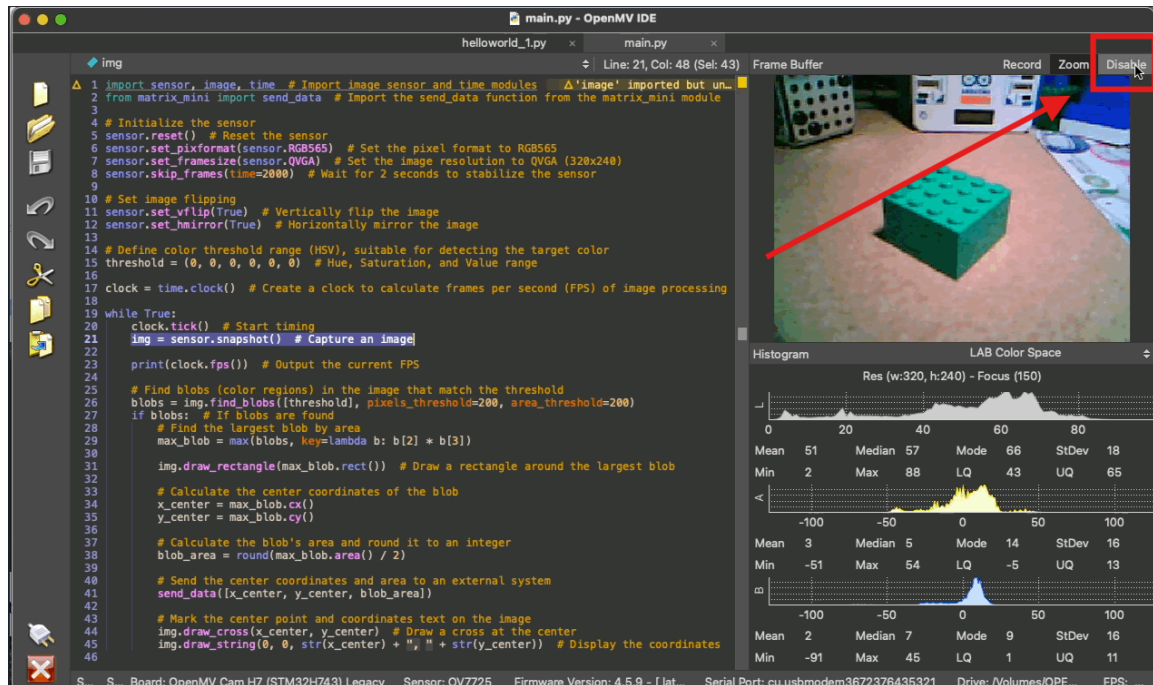


22.3 Getting Started with the Main Code

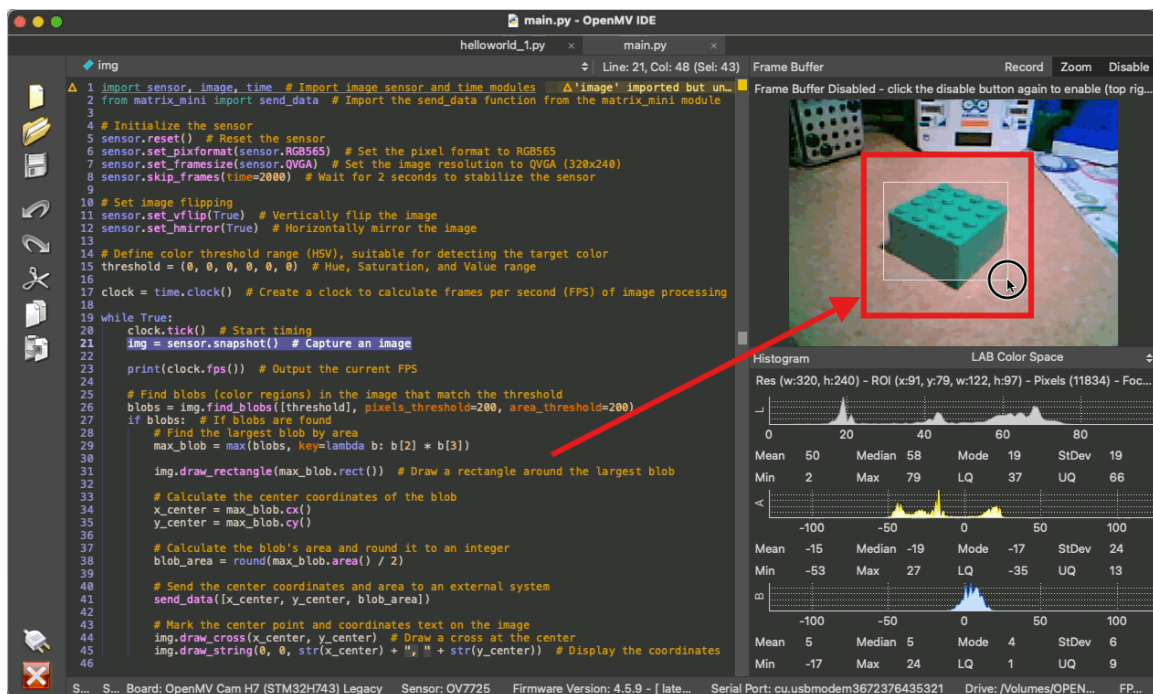
- Once M-Vision is successfully connected to your computer, its storage space will appear in File Explorer (or Finder on macOS under the Locations section in the left sidebar) with the name "OPENMV". Inside, you will find three basic programs: main.py, matrix_mini.py, and fill_light.py. Alternatively, you can download these files from [github.com/Matrix-Robotics/Products-documents/tree/main/Sensors/MVision_Resources/mPython%20Lib\(for%20MATRIX%20or%20Arduino\)](https://github.com/Matrix-Robotics/Products-documents/tree/main/Sensors/MVision_Resources/mPython%20Lib(for%20MATRIX%20or%20Arduino))



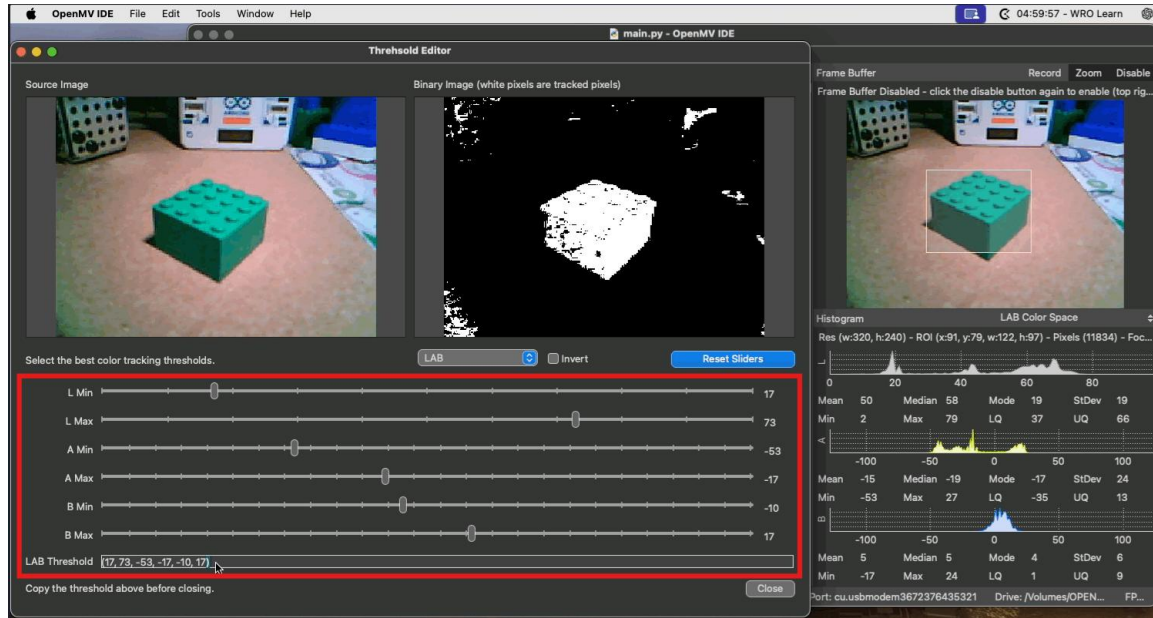
2. Navigate to Files > Open Files in the OpenMV IDE.
3. Select the OPENMV storage space (as explained in the first step) and choose main.py to open it.
4. Run the main.py code.
5. Point the M-Vision Camera at the target and click Disable (located at the top-right corner of the Frame Buffer).



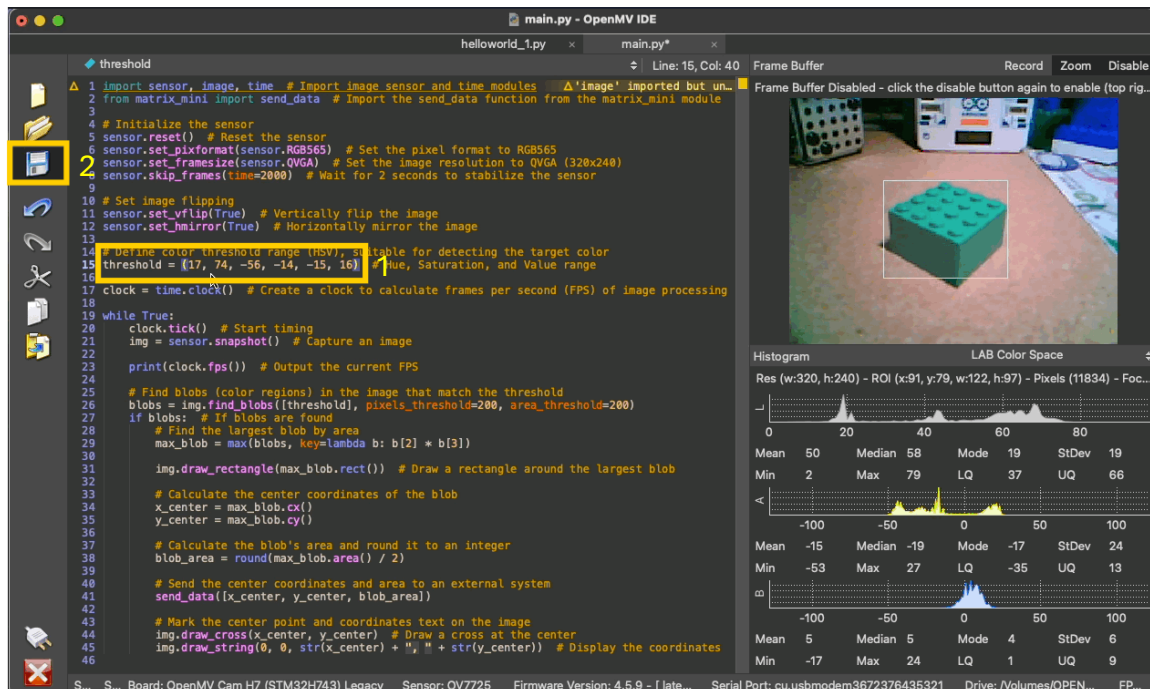
6. Use the mouse to select the target you want the camera to detect.



- Open the Threshold Editor by navigating to Tools > Machine Vision > Threshold Editor > Frame Buffer.
- Adjust the LAB threshold to the desired Min and Max values, using the histogram below the Frame Buffer in OpenMV IDE as a reference for the L, A, and B values. Aim to make the desired object appear mostly white while the rest of the area appears black. Once satisfied with the adjustments, copy the threshold values and close the Threshold Editor.

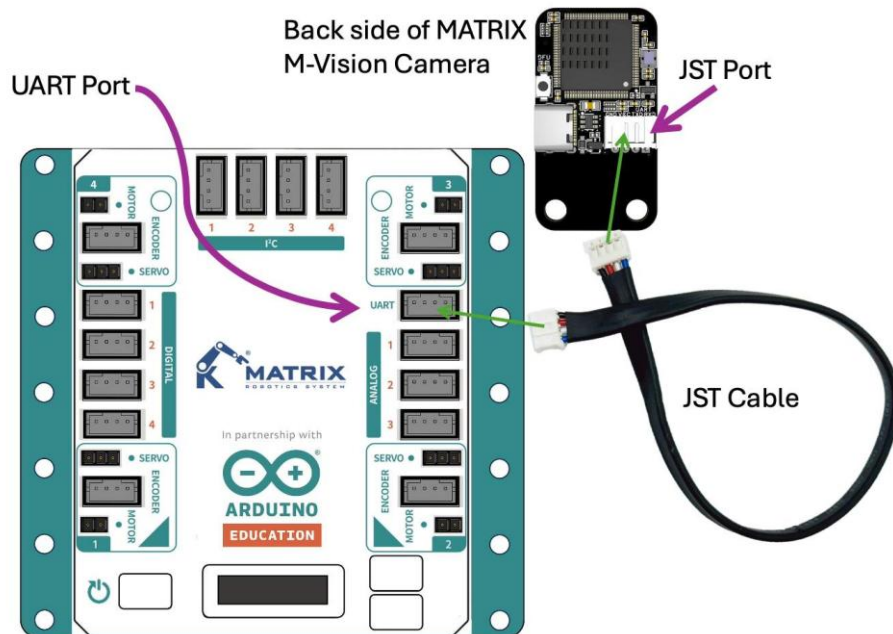


- After copying the LAB threshold values and closing the Threshold Editor window, replace the threshold values in your main.py code. Make sure to save the changes. When the Red LED light turns off, the data is saved, and the Camera is ready to use.

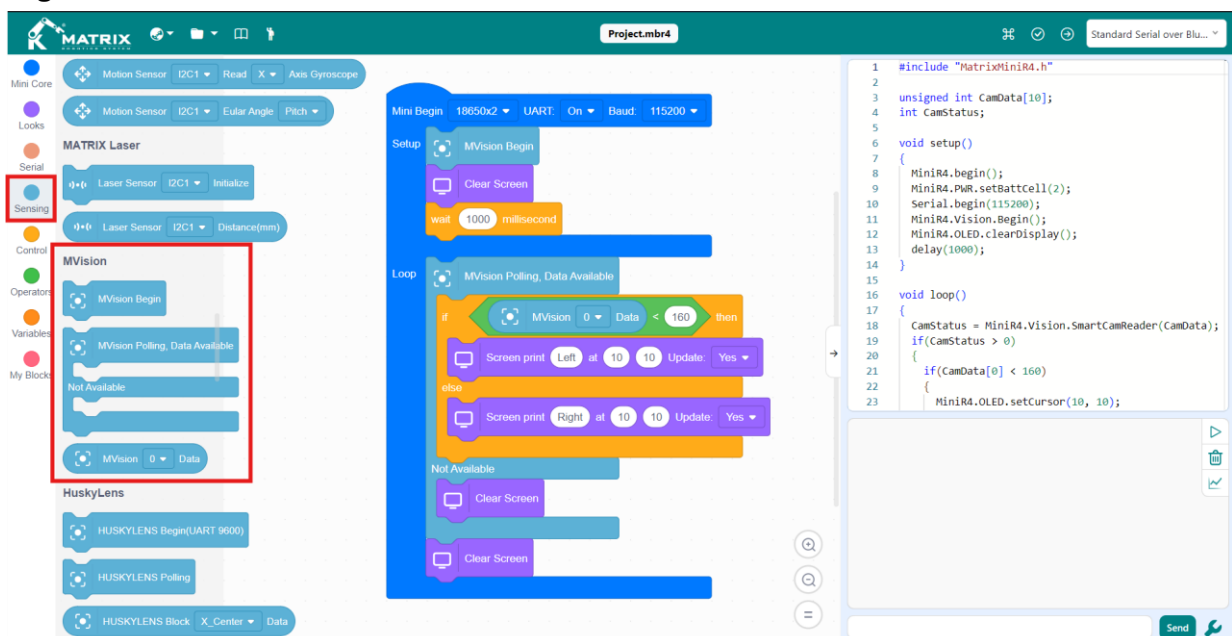


22.4 Connecting and Programming the M-Vision Camera with MATRIXblock

1. Use the USB-C cable to connect the MATRIX Mini R4 microcontroller to the computer.
2. Use JST Cable to connect M-Vision Camera to the UART port of MATRIX Mini R4.



3. Open MATRIXBlock on your computer.
4. Write the following code in MATRIXBlock to detect an object's position and display "Left" or "Right" on the OLED screen:



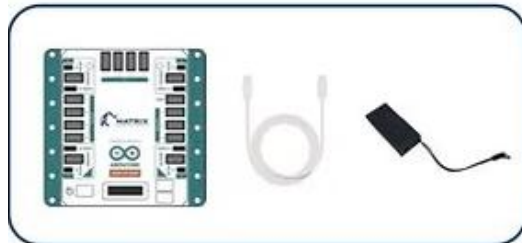
5. **Select the correct device** in MATRIXBlock.
6. Click on the **"Verify and Upload"** button (top right of the screen) to upload the code and test the results

22.5 Specifications

Parameter	Specification
Processor	STM32H7
Clock Speed	480 MHz
Flash Memory	2 MB (User available: ≥ 100 KB)
SRAM	1 MB
Storage Expandability	Supports up to 32 GB SD card
Max Supported Resolution	640×480 and below
Operating Voltage	5V
Operating Current	200mA (without fill-in light) / 350mA (with fill-in light)
Interfaces	USB Type-C, PH2.0-4P (UART communication)
Mounting Holes	M4 × 2

23. Packing List

23.1 Electrical Components



MATRIX Mini R4 Controller



TT Encoder Motor with metal gear box & Brackets(1 Pair)



TT Motor with metal gear box & Brackets(1 Pair)



2x Micro Servo with metal gear box



PIR Motion Sensor



Water Level Sensor



2x Color Sensor V3



Soil Moisture Sensor



Temperature & Humidity Sensor



Grayscale Sensor V3



2x Laser Sensor V2



2x Miniature Switch V3



Potentiometer Sensor



Gesture Sensor



One Wire Temperature Sensor



10x JST Cable 200mm



10x JST Cable 400mm

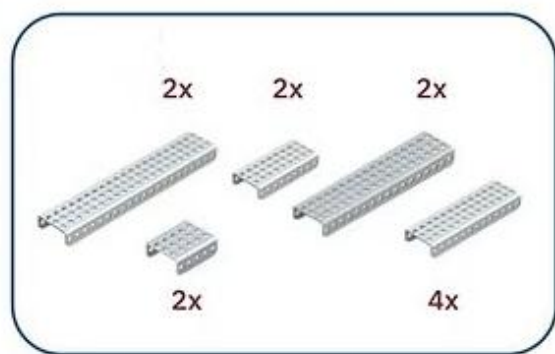
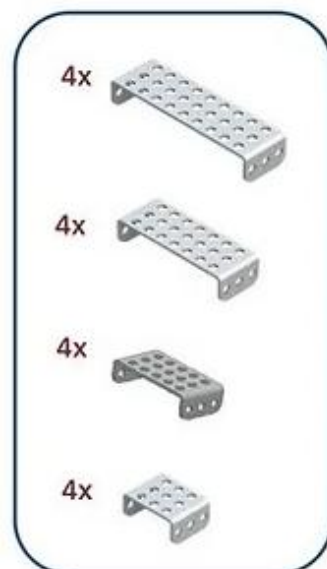
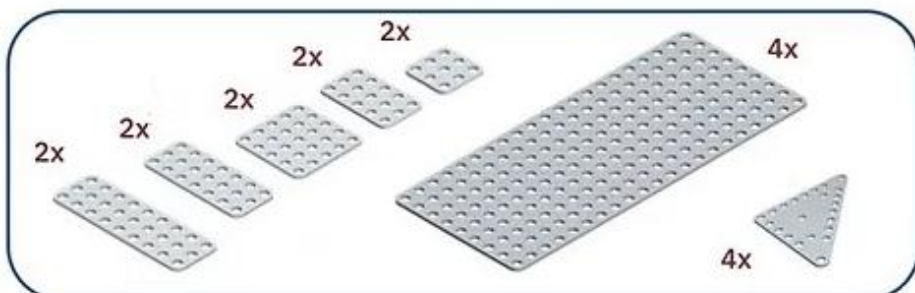
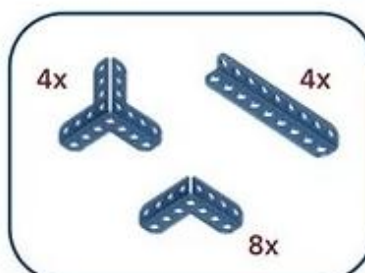
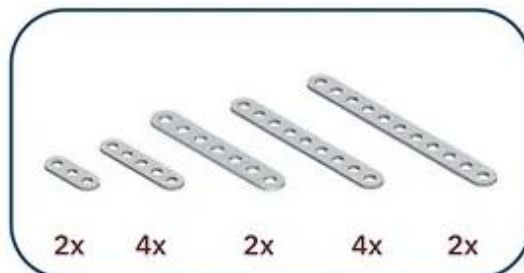
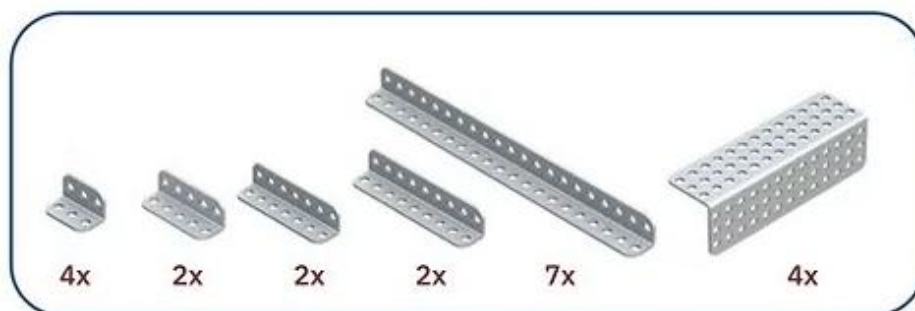


10x JST Cable 600mm



5x JST to Grove Cable

23.2 Construction components





24. Additional Details and References



github.com/Matrix-Robotics



www.arduino.cc