mi:node

User Manual

Table of Contents

1) Introduction

1.1 Overview

The mi:node kit is a modular, safe and easy to use group of accessories that work along the BBC micro:bit to help introduce children to the Internet of Things (IoT) – a term created to describe the growing network of devices that are able to connect to the Internet, collect data and exchange information.

With this kit there is no need for soldering, just connect and use. The construction of a working circuit can be easily completed in less than one minute.

1.2 Features

- Includes sensors for environmental and physical monitoring, as well as user interface modules (such as a switch) to allow a number of exciting projects including wearable applications and smart home devices.
- Rich education guide with lesson documentation and many project stories.
- Expandable
- Reusable

1.3 Kit Contents

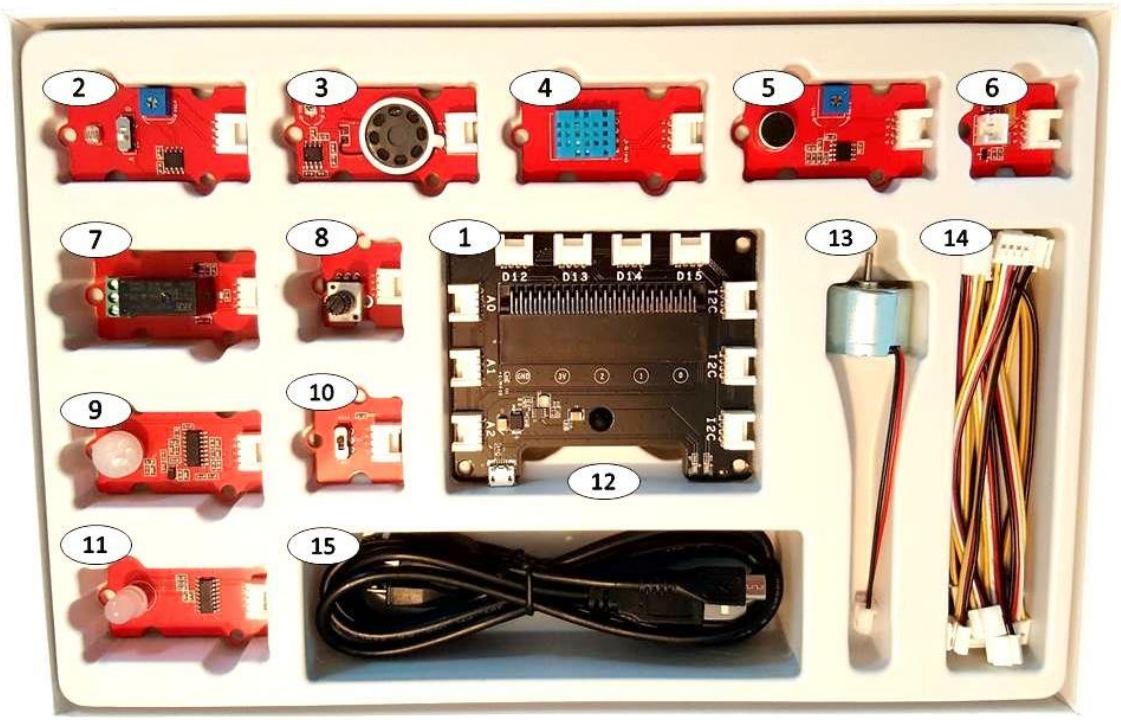


Image 1.3 – mi:node Kit Contents

Category	#	Module	Qty	Connector Type	Description
Connect Board	1	Connect Board	1	NA	A bridge between the micro:bit and the mi:node sensor modules
Sensor Modules (10)	2	Light Sensor	1	Analog Input	Detects the intensity of light in an environment
	3	Speaker	1	Analog Output/ PWM	Voice output amplifier
	4	Temperature and Humidity Sensor	1	Analog Input	Senses temperature and humidity within the environment
	5	Sound Sensor	1	Analog Input	Detects the sound strength within an environment
	6	Mini Fan	1	Analog Output/ PWM	A connector board for the DC Motor and Orbit Fan
	7	Relay	1	Digital Output	A digital switch used to control high-voltage electrical devices, up to a maximum of 250V
	8	Rotary Angle	1	Analog Input	A switch with a 0 - 300 degree dial used to control voltage output from 0V to maximum
	9	PIR Motion Sensor	1	Digital Input	Senses motion; usually human movement within range
	10	Switch	1	Digital Input	Used to switch voltage ON/OFF
	11	RGB LED	1	Digital Output	A colorful Light Emitting Diode. The color and brightness can be programmed.
Accessories	12	Orbit Fan	1	Analog Input	Small handheld fan (Located under the Connect Board in the box)
	13	DC Motor	1	Analog Output/ PWM	Direct Current (DC) motor used to run the Orbit Fan using the Mini Fan connector board. Can be used to run other small devices as well.
Cables	14	E-Brick Connector Cable	8	N/A	Double ended cables of two different lengths used to connect sensor modules to the Connect Board. 2 x 20cm cable 6 x 10cm cable
	15	Micro-USB to USB Cable	2	N/A	Two cables, One for power input to the Connect Board. One for micro:bit program upload.

Table 1.3 – Kit Contents

* Note: BBC mi*cro:bit not include; sold separately.*

2) Getting Started

2.1 The Connector Board

The kit is comprised of a Connector Board and several sensor modules. The Connect Board is a bridge between the BBC micro:bit and the mi:node sensor modules. It converts the BBC micro:bit edge connector into several e-brick connectors. The sensor modules can then easily be attached to it using the provided e-brick cables.

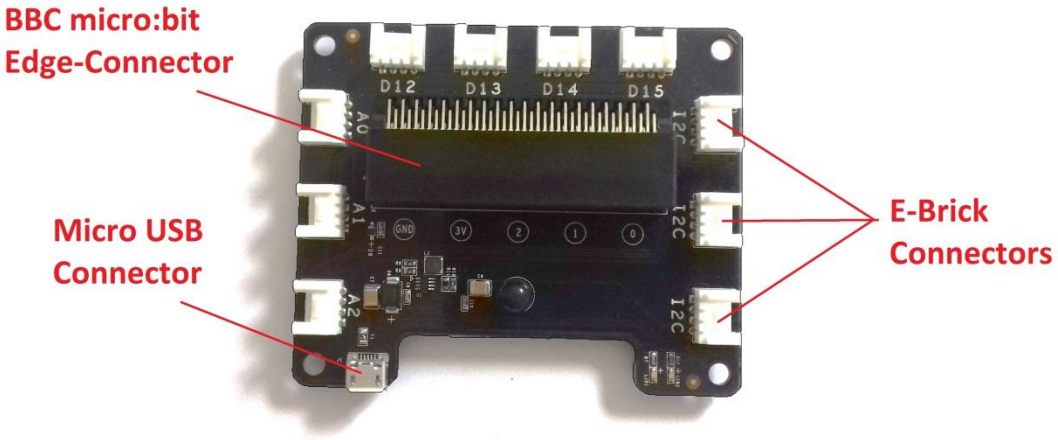


Image 2.1 – mi:node Connect Board

2.2 E-Brick Connectors

E-Brick Connectors are compatible with Grove –a standardized connector for prototyping systems created by the company called Seeed. This connector enables the mi:node to be a plug-and-play type product. The E-Brick Connectors each have 4 pins which connect directly to the E-Brick Connector Cables.

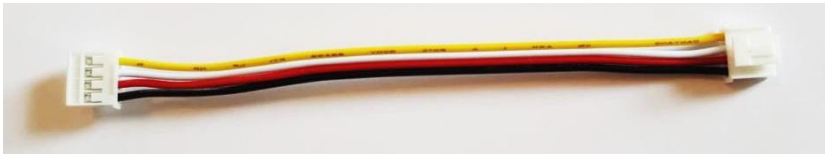


Image 2.2 – E-Brick Connector Cable

E-Brick Connector Pin-Out:

Pin No. (Cable Color)	Pin Name	Description
1 (Yellow)	Signal #1	This is a communication pin. It allows you to connect to the analog input, digital input/output, and the I2C functionality of the micro:bit so that you can read from and control your mi:node sensor modules.
2 (White)	Signal #2	This pin is a duplicate of Pin No. 1 described above. Note: Usually only one communication pin is required for most sensors.
3 (Red)	Vcc	Power connection.
4 (Black)	Gnd	Ground connection.

Table 2.2 – E-Brick Connector Pin-Out

2.3 Inputs

There are 3 different data transfer types between the E-Brick Connectors on the mi:node Connector Board.

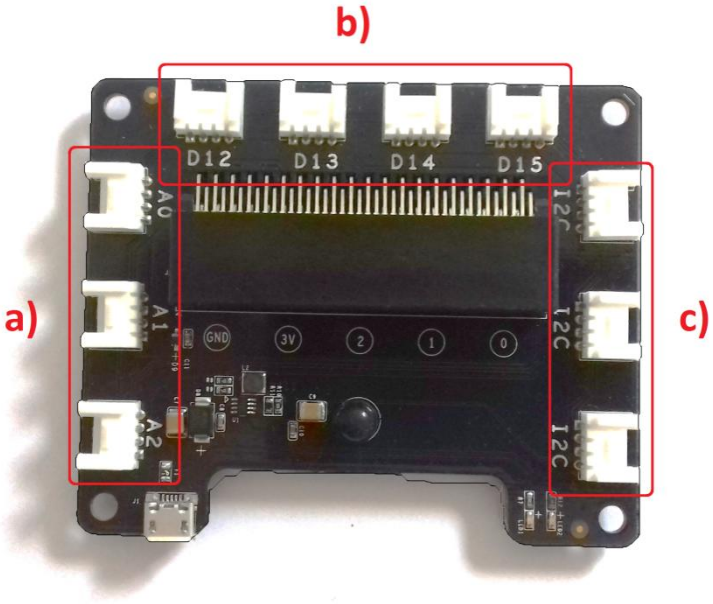


Image 2.3 – mi:node Connector Board 2

a) Analog In/PWM (Pulse Width Modulation)

Pulse Width Modulation (PWM) is a technique used to encode a message into a pulsing signal. Its main use is to allow the control of the power supplied to electrical devices, such as motors.

Pin Name	Description
A0	Connects to a micro:bit pin with analog input or PWM function
A1	Connects to a micro:bit pin with analog input or PWM function
A2	Connects to a micro:bit pin with analog input or PWM function

b) Digital IO

Pin Name	Description
D12	Connects to a micro:bit pin with digital IO (input or output) function
D13	Connects to a micro:bit pin with digital IO (input or output) function
D14	Connects to a micro:bit pin with digital IO (input or output) function
D15	Connects to a micro:bit pin with digital IO (input or output) function

c) I2C (Inter-Integrated Circuit)

This is a form of electronic communication standard that requires two signals, a data signal and a clock signal. It allows information to be sent in packages, on what is called a bus, to and from the BBC micro:bit and any attached sensor modules.
It also allows you to communicate to several sensor modules using the same bus. This is

done by identifying each sensor module with a different address.

Pin Name	Description
I2C SCL	I2C clock signal. Connect to micro:bit pin19
I2C SDA	I2C data signal. Connect to micro:bit pin20

2.4 Pin Mapping

Pin mapping is done to ensure the outputs from the mi:node E-Connector pins are correctly connected (or 'mapped')to the corresponding pins of the BBC micro:bit edge-connector. Below is a table illustrating the mi:node Pin Mapping.

Connector Type	Connector Name	micro:bit Pin Name
Analog Input /Digital IO	A0	pin0, pin1
	A1	pin1, pin2
	A2	pin2, pin3
Digital IO	D12	pin12, pin13
	D13	pin13, pin14
	D14	pin14, pin15
	D15	pin15, pin16
I2C	I2C	pin19, pin20
	I2C	pin19, pin20
	I2C	pin19, pin20

Table 2.4 – Pin Mapping

Typically using names that start with the letter 'A' (like A0,A1, A2) denote ANALOG inputs, but these connections can sometimes also be used as digital inputs or outputs.

Names starting with the letter 'D' can only be used to denote DIGITAL inputs and outputs. This is similar to the I2C in that it can only be used to denote I2C connections.

3) Usage

3.1 Programming Using mi:node

There are five different code editors to choose from on the BBC micro:bit official website.

- Microsoft PXT
- Code Kingdoms JavaScript
- Microsoft Block Editor
- Microsoft Touch Develop
- python

In this document we will focus on using the **Microsoft PXT** editor.

3.2 Working With mi:node Libraries

The mi:node has pre-created libraries that you can utilize for free. Working with these libraries saves the time of having to study the technical details of each module before beginning to program.

For example, the RGB LED module works using I2C communication. By utilizing the already-created mi:node library, you just need to call the RGB LED library function to control color and brightness. You don’t have to worry about the details of how the I2C communication does this.

With the pre-created libraries you also don't need to think about which micro:bit pin a mi:node sensor module connects to. You only need to know the name of the connector that your sensor module and E-Brick Cable is connected to.

Steps to follow:

- Go to https://makecode.microbit.org
- Add the mi:node library to your code

a) Click “More…” and select “Add Package…” from the drop down menu.

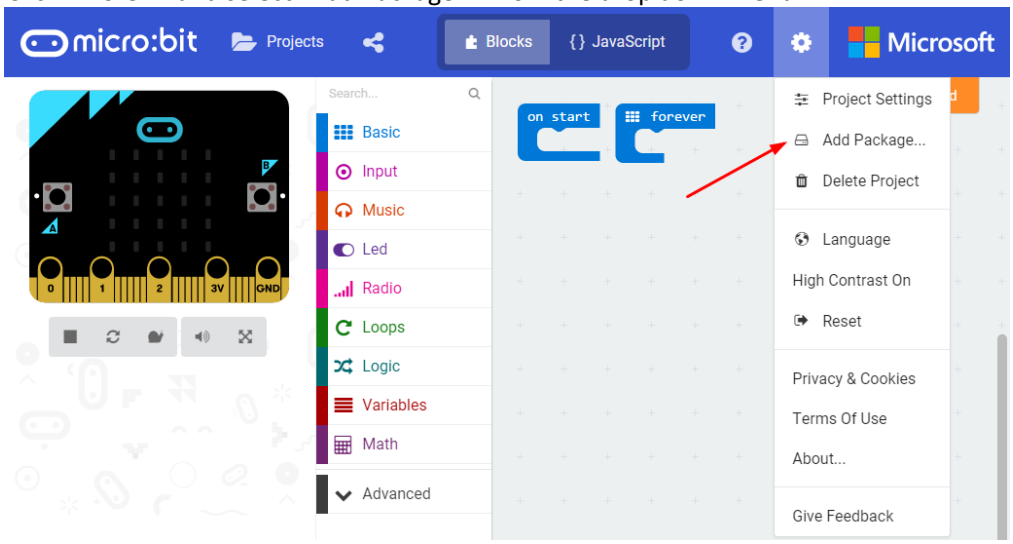


Image 3.2a – Click “Add Package…”

b) Type “minode” or “node”, then select the search button.

You can also enter the following project URL:
https://github.com/minodekit/pxt-minode before selecting the search button.

Both options will pull up the mi:node library package.

Select the “minode” library to add it to your project.

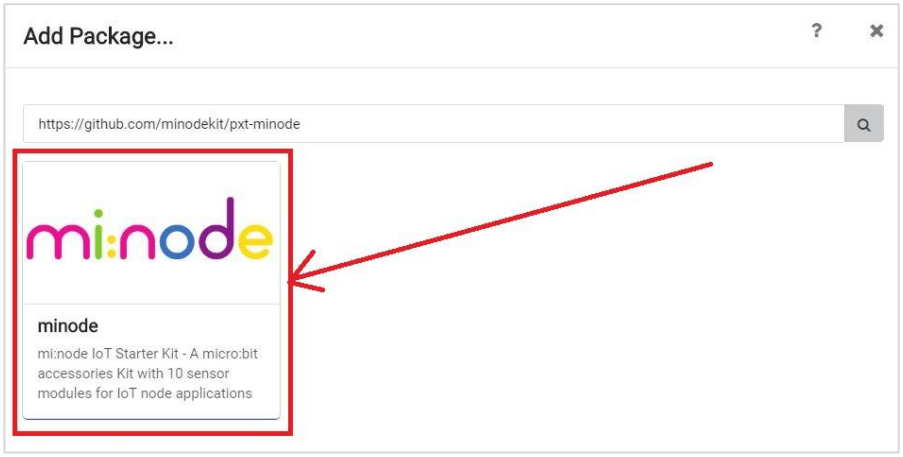


Image 3.2b – Search the package and add it

c) After selecting the mi:node library, wait a moment and you will see it being added to your project. This may take a few seconds.

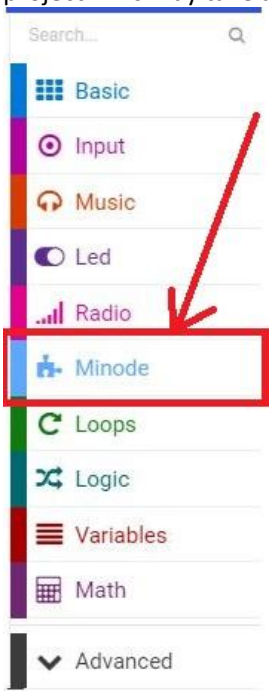


Image 3.2c–mi:node packed added

- Refer to the example code or API reference and drag the library blocks you want to use into your project space.

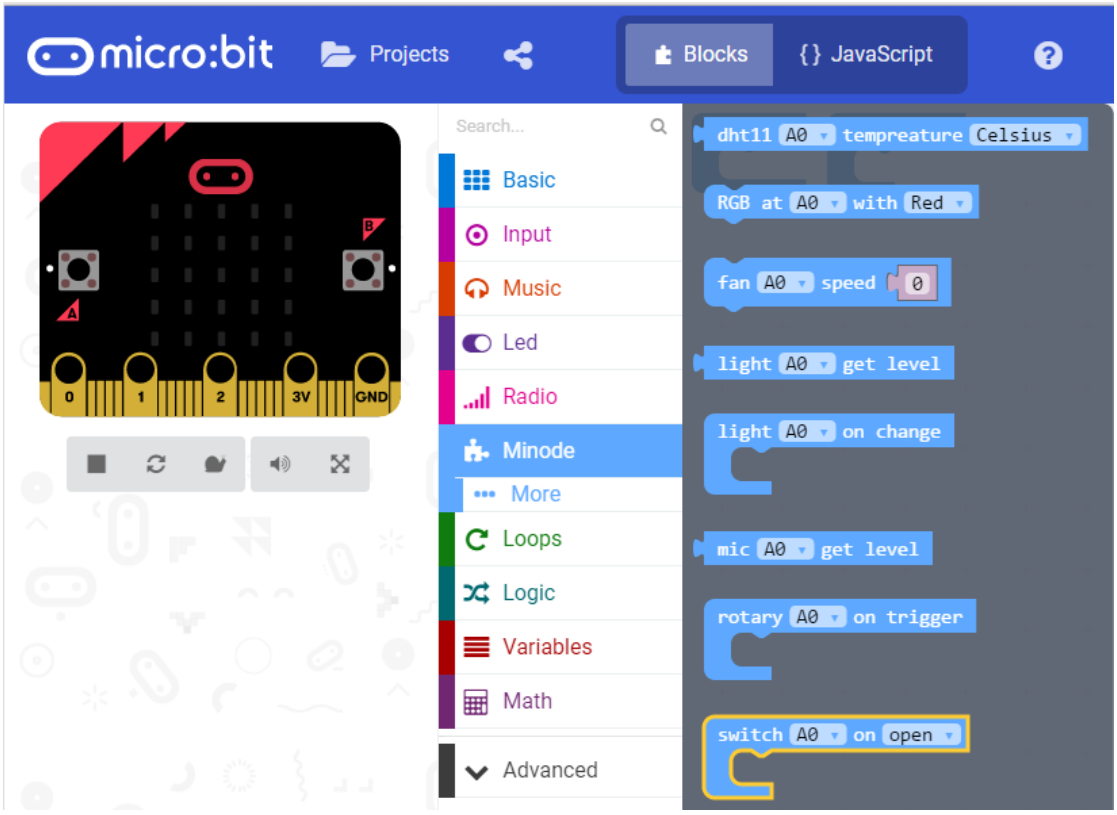
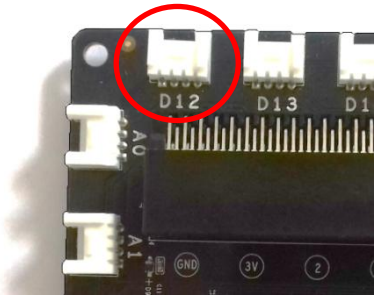


Image 3.2c– mi node package

3.3 Working Without mi:node Libraries

You can also use the micro:bit pin library to control the sensor modules directly. To do so, the micro:bit pin ID is required. This ID can be found using the E-brick connector ID and the lookup table shown in *Table 2.4 – Pin Mapping* above.

For example, if you connect a sensor module to D12 on the Connector Board, from the table you can see that this connector corresponds to pin12 on the micro:bit.



Connector Type	Connector Name	micro:bit Pin Name
Digital IO	**D12**	**pin12**, pin13

3.4 Module Usage & API Reference

This section describes the main uses for each of the mi:node sensor modules and the API (Application Program Interface) references required to carry them out.

a) Light Sensor

The Light Sensor module can be used to detect the intensity of light in the surrounding environment.

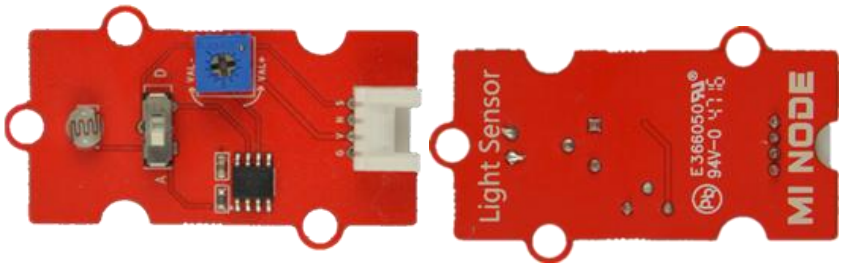


Image 3.4a – Light Sensor Module

This module requires an analog connection, therefore can only be plugged into A0, A1, or A3 on the Connector Board.

Module	Connect Type	Available Connectors
Light Sensor	Analog	A0, A1, A2

Electrical Characteristics

Parameter	Min.	Typical	Max.	Unit
Photoresistor (light intensity is 10lux)	5	-	10	kΩ
Threshold hysteresis ΔUth	-	VCC*0.09	-	V

Block API:

Obtaining the Light Level

The light sensor module divides brightness into five distinct levels, 1 – 5, with one being the brightest and five being the most dull.

Function

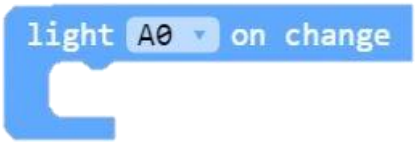
LightSensorGetLevel(connName: AnalogConnName): number;

Parameters

'ConnName' is the analog connector's name.

Light Sensing Event

This function is used to check the light level periodically and then execute the associated code block whenever the light level changes.



Function

onLightSensorEvent(connName: AnalogConnName, body: () => void): void;

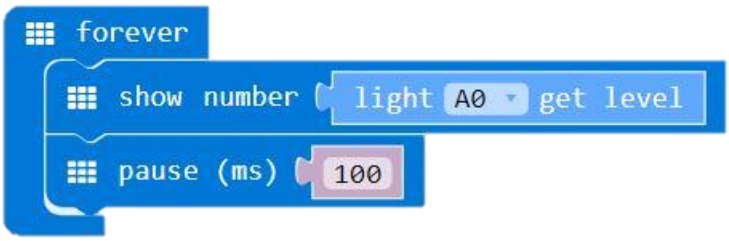
Parameters

'ConnName' is the analog connector's name.

EXAMPLE:

Display Light Level

This example shows you how to get the current light level and display it on the LED screen.



Light Level Change Notification

This example displays the word 'Change' on the LED screen whenever the light level changes.

b) Temperature and Humidity Sensor (DHT11)

This sensor module features a temperature & humidity sensor with a calibrated digital signal output. It can measure temperature and humidity in the surrounding environment.

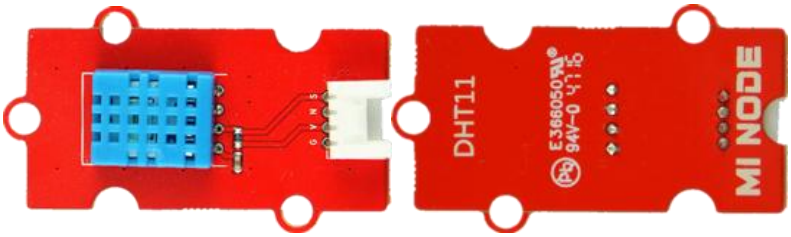


Image 3.4b – Temperature& Humitity Sensor Module

Module	Connect Type	Available Connectors
DHT11	Digital IO	D12, D13, D14, D15

Electrical Characteristics

Humidity

Parameter	Min.	Typical	Max.	Unit
Accuracy (25℃)	-	±4	-	%RH
Accuracy (0-50℃)	-	-	±5	%RH
Measurement range (25℃)	20	-	95	%RH
Response time: 1/e (63%) 25℃， 1m/s air	6	10	15	s

Temperature

Parameter	Min.	Typical	Max.	Unit
Accuracy	±1	-	±2	℃
Measurement range	0	-	50	℃
Response time /e (63%)	6		30	s

Block API

Measure Temperature

Obtain the current temperature. This value can be configured the format of Celsius or Fahrenheit.

Function

DHTGetTemperature(connName: ConnName, style: DHTTemStyle): number;

Parameters

'ConnName' is the connector's name.

'style' is the format of the temperature. You can choose Celsius or Fahrenheit.

Measure Humidity

Obtain the current humidity.

Function

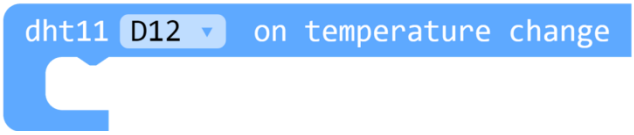
DHTGetHumidity(connName: ConnName): number;

Parameters

'ConnName' is the connector's name.

Temperature Change Event

Check the temperature periodically and execute the associated code block whenever the temperature changes. The smallest detectable unit of change is 1 degree Celsius.



Function

onDHTEvent(connName: ConnName, body: () => void): void;

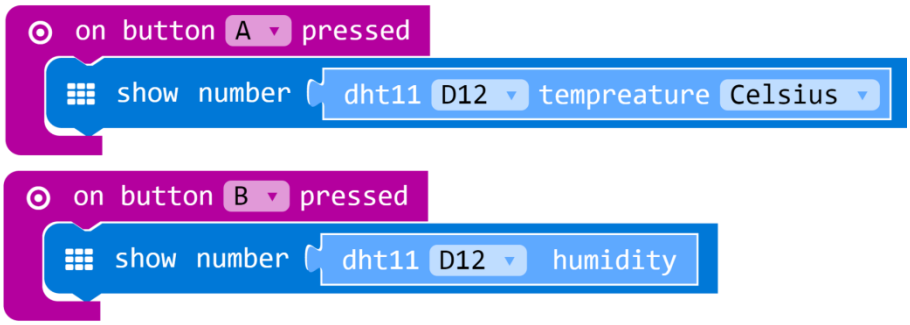
Parameters

'ConnName' is the analog connector's name.

EXAMPLE:

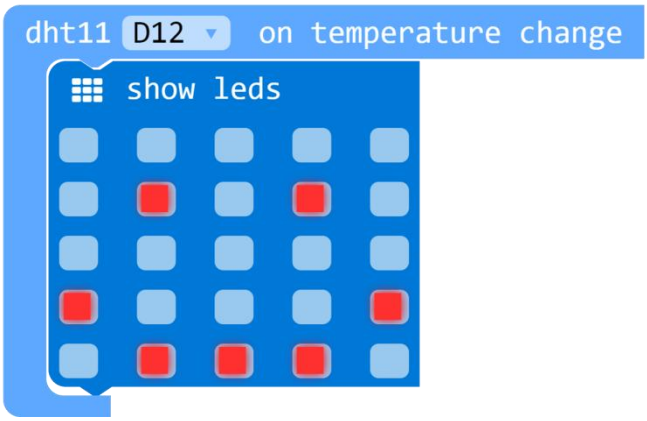
Pressing Buttons to Display Temperature and Humidity

This example shows you how to use Buttons A and B to obtain the temperature and humidity of the surrounding environment. When Button A is press the screen will show the current temperature. When Button B is pressed, it will show the humidity



Temperature Change Notification – Smiley Face

Display a smiley face on the LED screen when the temperature changes.



c) Sound Sensor

The Sound Sensor Module can be used to detect the sound strength of the surrounding environment.

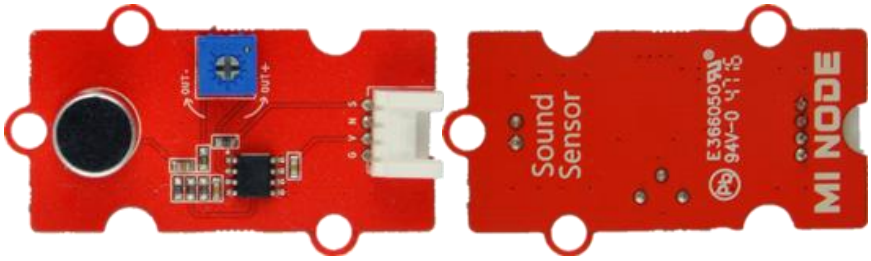


Image 3.4c – Sound Sensor Module

Module	Connect Type	Available Connectors
Sound Sensor	Analog	A0, A1, A2

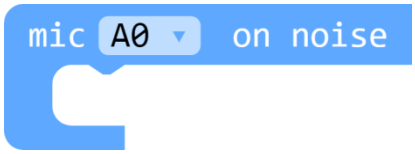
Electrical Characteristics

Parameter	Min.	Typical	Max.	Unit
Frequency range	100	-	10000	Hz
Sensitivity	-	-50	-	dB

Block API:

Sound Sensor Event

Check if there were any noise periodically and execute the associated code block whenever it detected a noise. It can be triggered by clapping your hand.



Function

onMICEvent(connName: AnalogConnName, body: () => void): void;

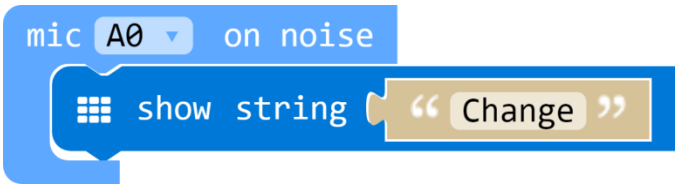
Parameters

'ConnName' is the analog connector's name.

EXAMPLE:

Noise detect Notification

When a noise is detected, the word 'Change' is displayed on the LED screen.



d) Rotary Module

The Rotary Module has a potentiometer that can produce analog output between 0 and Vcc (the supply voltage). This is done by turning the dial, adjusting the range from 0 – 300 degrees.

Image 3.4d – Rotary Module

Module	Connect Type	Available Connectors
Rotary Sensor	Analog	A0, A1, A2

Electrical Characteristics

Parameter	Min.	Typical	Max.	Unit
Resistance range	0	-	10	kΩ

Block API

Obtain the Rotary Position

Determine the current rotary position, the percentage of how much the rotary dial has been rotated from its zero position.

Function

RotaryGetPercentage(connName: AnalogConnName): number;

Parameters

'ConnName' is the analog connector's name.

Rotary Position Event

Check the angle of the rotary dial periodically and then execute the associated code block when a dial change occurs.

Function

onRotaryEvent(connName: AnalogConnName, body: () => void): void;

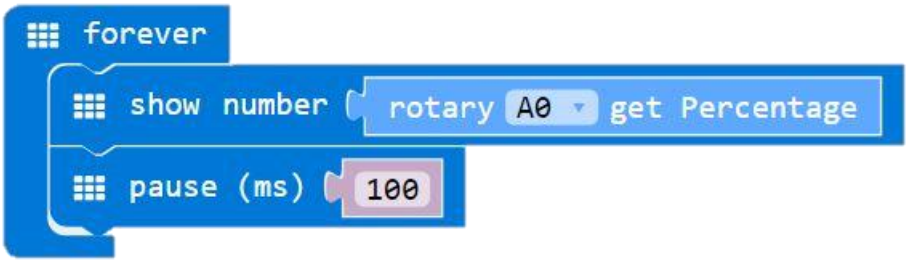
Parameters

'ConnName' is the analog connector's name. This module can only be plugged into analog connector A0,A1 and A2.

EXAMPLE:

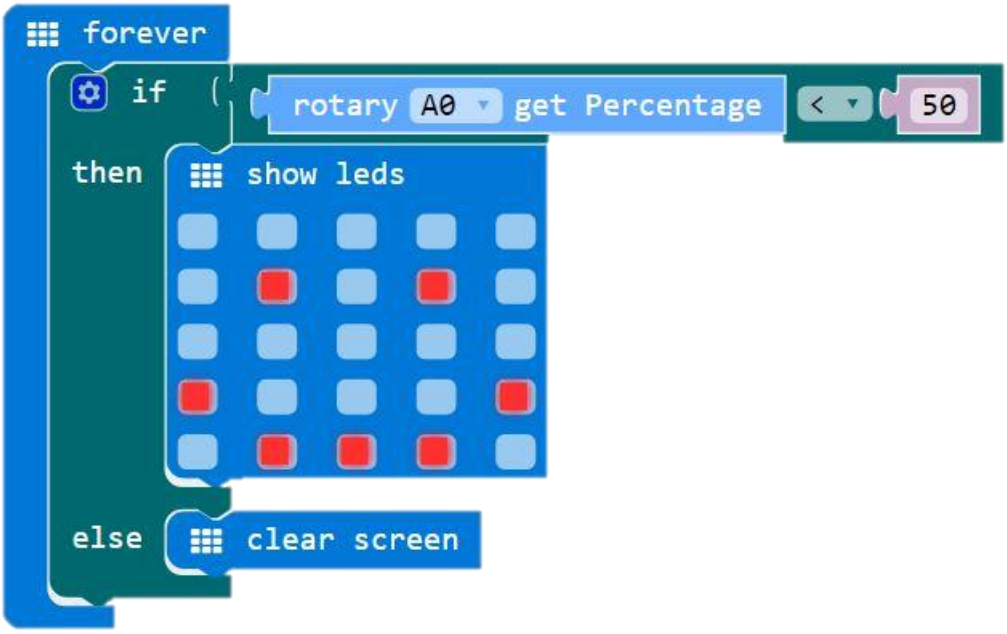
Display Rotary Angle Percentage

This example shows you how to get the current rotary angle percentage and display it on the LED screen.



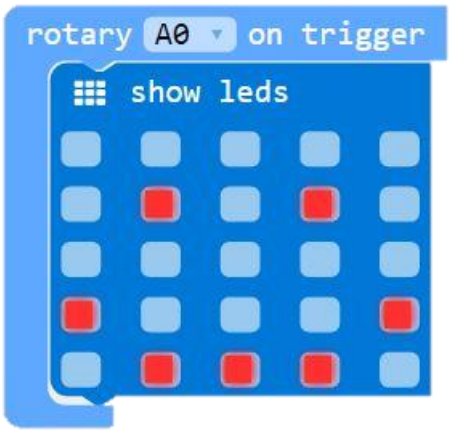
Controlling the LED Screen – Smiley Face

When the rotary percentage is smaller than 50%no LEDs will be lit on the screen. When the percentage is larger than 50% the LEDs will light up in the form of a smiley face.



Rotary Change Notification – Smiley Face

While rotating the rotary dial, the LED screen will display a smiley face. When the dial is stationary, no LEDs will be lit on screen.



e) Mini Fan Module

The mini fan module is designed to be used with the DC motor and Orbit Fan. The speed of the motor can be controlled according to different situations.

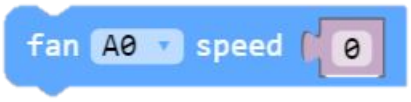
Image 3.4e – Mini Fan Module with DC Motor and Orbit Fan

Module	Connect Type	Available Connectors
Mini Fan Module	Analog	A0, A1, A2

Block API

Control the Motor Speed

To change the speed of the motor/fan, adjust the second parameter value to a number within the range of 0 to 100.



Function

FanControl_1(connName:AnalogConnName , speed:number): void

Parameters

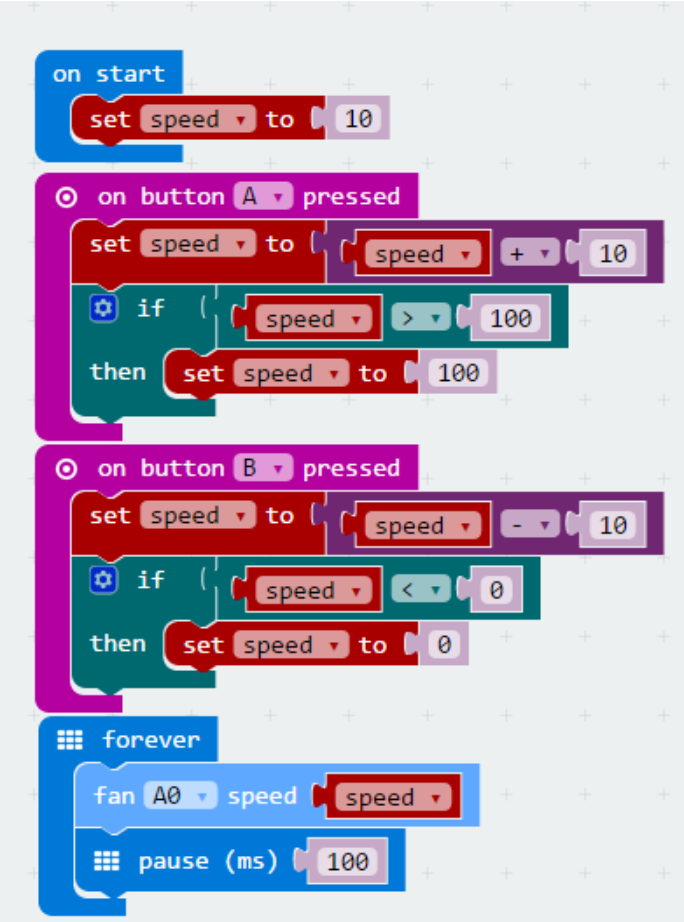
'ConnName' is the analog connector's name.

'Speed' is the velocity of the motor; where 0 is used to stop the motor and 100 is the fastest speed.

EXAMPLE:

Use the Button to Control Motor Speed

This example demonstrate show to use Button A and B to control the motor speed. Button A tells the motor to speed up and Button B tells the motor to slow down.



f) Speaker Module

The Speaker can be used to make a sound.

NOTE: As a default, the speaker is connected through pinP0 of the micro:bit, therefore it is important to connect the speaker module to Connector A0.

Image 3.4f – Speaker Module

Module	Connect Type	Available Connectors
Speaker Module	Analog	A0

EXAMPLE:

Play a Sound Through the Speaker

The speaker will cycle through three sounds/beats.

g) PIR Sensor Module

This PIR (passive infrared sensor) module detects movement, usually human, within its range. When the PIR detects motion the modules acts like a switch, the signal line will change from low to high and then stay high for 3 seconds before dropping back to low.

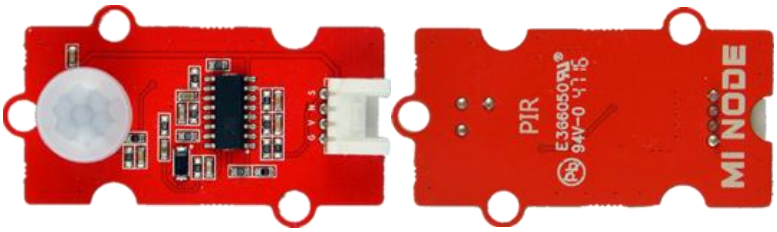


Image 3.4g – PIR Sensor Module

Module	Connect Type	Available Connectors
PIR Module	Digital IO	D12, D13, D14, D15

Electrical Characteristics

Parameter	Min.	Typical	Max.	Unit
Detection angle(solid angle)	-	-	110	°
Detection range	-	-	7	m
Delay time of high level	-	2.5	-	s

Block API

Check the PIR status

Check whether the PIR module has been triggered or not. When the PIR module is triggered the signal line will be detected as high.

Function

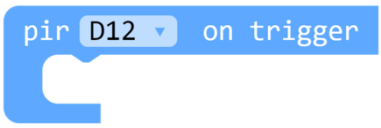
PIRIsTriggered(connName: ConnName): boolean;

Parameters

'ConnName' is the connector's name.

PIR Detection Event

This code block configures the selected specified pin for a digital input and then executes the associated code block whenever the PIR is triggered (movement is detected).



Function

onPIREvent(connName: ConnName, body: () => void): void;

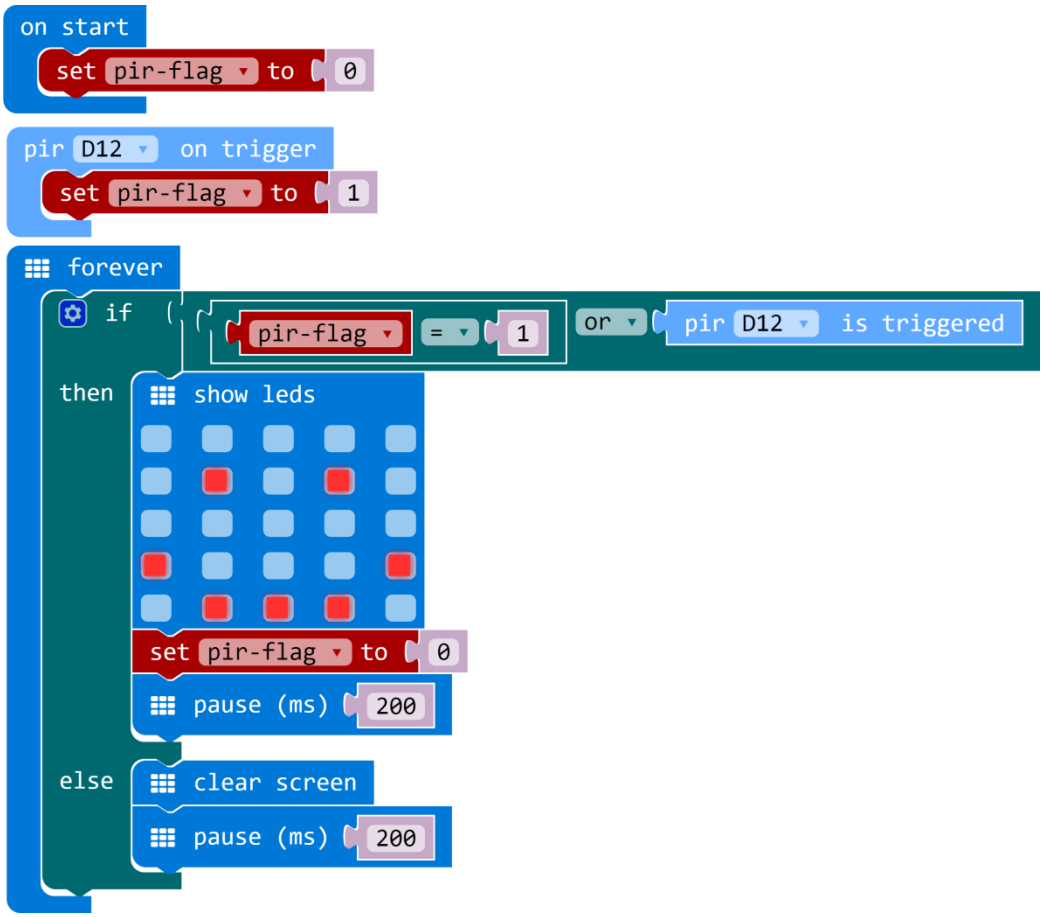
Parameters

'ConnName' is the connector's name.

EXAMPLE:

Detecting Movement – Smiley Face

When the PIR detects a moving object, the LED screen will show a smiley face. When there is no movement detected within the PIR modules range no LEDs will be lit.


on start
set pir-flag to 0
pir D12 on trigger
set pir-flag to 1
forever
if
pir-flag = 1
or
pir D12 is triggered
then
show leds
set pir-flag to 0
pause (ms) 200
else
clear screen
pause (ms) 200

h) RGB LED Module

This module includes an LED that can display a number of different colors. The color and brightness can be programmed and are controlled by the greyscale value of red, green and blue.

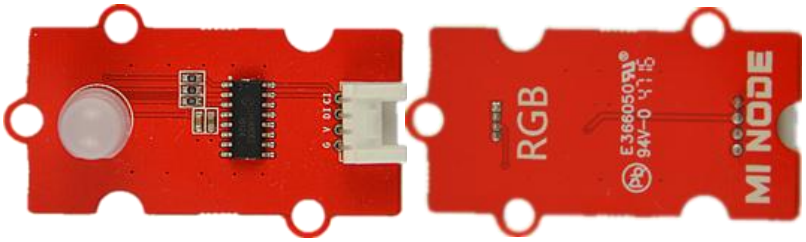


Image 3.4h – RGB LED Module

Module	Connect Type	Available Connectors
RGB LED	Digital IO	D12, D13, D14, D15

Block API

Choose a Pre-Coded Color

The color the RGB LED displays can be selected.

Function

RGBChooseColor(connName: ConnName, color: MiNodeColor): void;

Parameters

'ConnName' is the analog connector's name.

'Color' is your choice of one color from the following pre-coded mi:node colors: Red, Green, Blue, Yellow, Pink, Cyan or White.

Select Any Color

To change the color of the RGB, adjust the three greyscale values.

Function

RGBSetColor(connName: ConnName, red: number, green: number, blue: number): void;

Parameters

‘ConnName’ is the analog connector's name.

‘red’ is the greyscale value of red, ranging from 0 to 255.

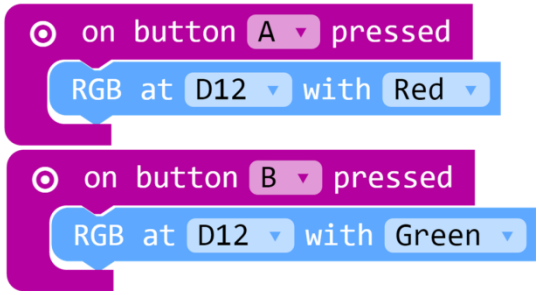
‘green’ is the greyscale value of green, ranging from 0 to 255.

‘blue’ is the greyscale value of blue, ranging from 0 to 255.

EXAMPLE:

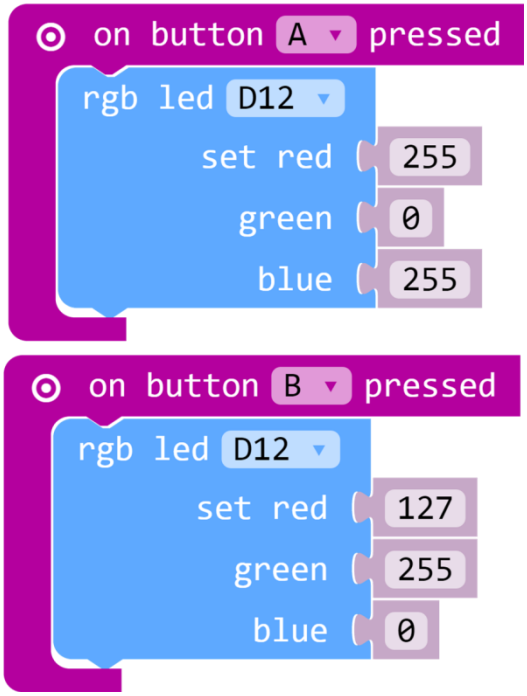
Use Button A and B to Display Pre-Coded Colors

When you press button A the RGB LED will light up red. Pressing Button B will cause the RGB LED to turn green.



Use Button A and B to Display Any Color

This example shows you how to set a specific color for Button A and B.



i) Switch Module

The switch module can used to switch between two options (ie. on/off).

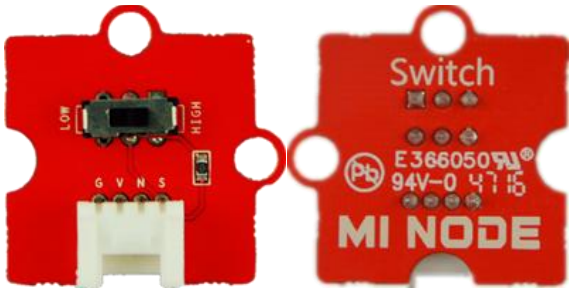


Image 3.4i – Switch Module

Module	Connect Type	Available Connectors
Switch Module	Digital IO	D12, D13, D14, D15

Block API

Check the switch's status

Check whether the switch is open or closed.

Function

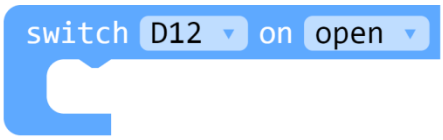
switchIsOpened(connName: ConnName): boolean;

Parameters

'ConnName' is the connector's name.

Switch Event

This code block configures the specified pin for a digital input and then executes the associated code block whenever the switch is either opened or closed, as selected.



Function

onSwitchEvent(connName: ConnName, event: SwitchEvent, body: () => void): void;

Parameters

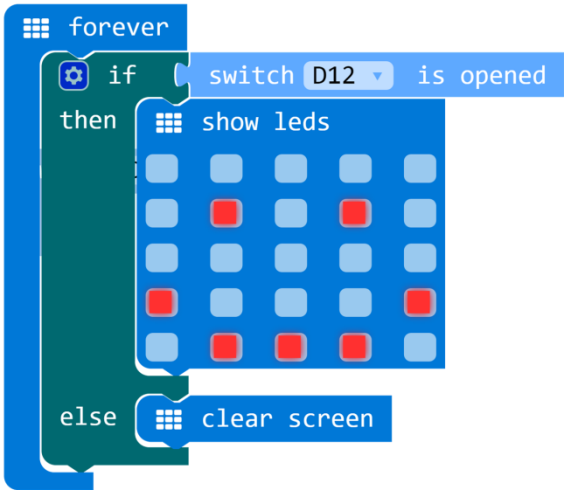
'ConnName' is the analog connector's name.

'SwitchEvent' represent the status of switch; either open or closed.

EXAMPLE:

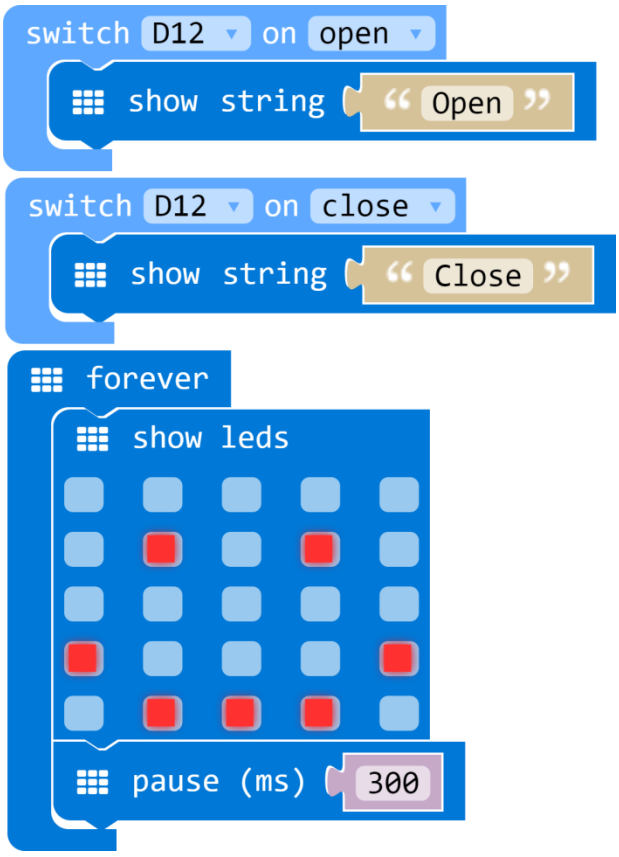
Switch Control – Smiley Face

If the switch is opened, the LED screen will show a smile face. If the switch if closed, the LED screen will not be illuminated.



Display Switch Status

In this example we use the switch event to display the status of the switch, whether it is open or closed. When the switch is open the led screen will display the word 'Open'. When the switch is closed it will display the word 'Close'.



j) Relay Module

The relay is an electrically operated switch. It is a digital switch that can be used to control high-voltage electrical devices, such as some home appliances. (Up to a maximum of 250V).

***IMPORTANT:** It can be dangerous to attach the relay module to an AC(110V/220V) device. Our purpose for this module is to illustrate how home appliances can be controlled. It is NOT necessary to connect a real appliance. A 'clicking' sound can be heard when the relay switches ON/OFF.*

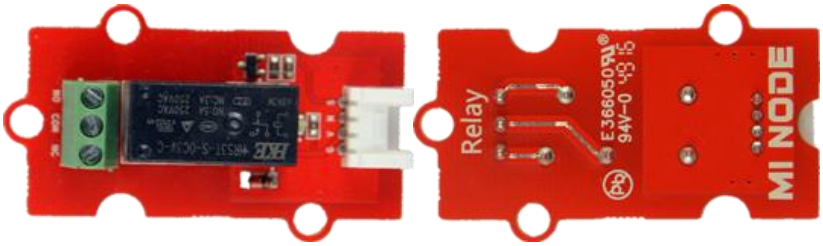


Image 3.4j – Relay Module

Module	Connect Type	Available Connectors
Relay Module	Digital IO	D12, D13, D14, D15

Electrical Characteristics

Parameter	Description
Contact Rating	NO:5A250VAC/28VDC NC:3A 250VAC/28VDC
Operate Voltage ≤ V dc	2.25
Release Voltage ≥ V dc	0.3

Block API

Set Relay Status

Use this code block to set the relay module's status to open or close.

Function

RelayControl(connName:ConnName , status:FanStatus): void

Parameters

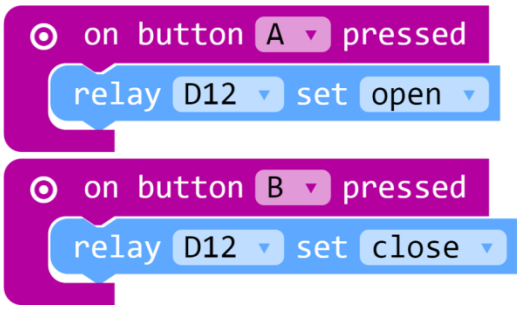
'ConnName' is the connector's name.

'status' is the status of the relay open or close.

EXAMPLE:

Control the Relay using Button A and B

In this example pressing Button A will set the relay to the open position. Pressing Button B will set the relay to the closed position.



4) Appendix

4.1 Microsoft MakeCode

- Home: https://makecode.microbit.org
- Reference: https://makecode.microbit.org/reference
- MakeCode (PXT) Documentation: https://makecode.com/docs

4.2 Support

- Website: http://www.embest-tech.com/prod_view.aspx?TypeId=83&Id=383&Fid=t3:83:3
- Github Repository: http://github.com/minodekit