Table of Contents

Auxiliary Control Components

Momentary Pushbutton Module Datasheet 2

Momentary Pushbutton Module Datasheet

Overview

The Momentary Pushbutton Module, CE-MD-014-0001, extends MachineMotion 2's functionality with two momentary pushbuttons. This plug-and-play module only requires a single connection to the MachineMotion 2 controller. Compatible modules, such as the Power Switch (CE-MD-005-0000) & additional pushbutton modules can also be daisy chained to each other, making it possible to connect up to eight modules per MachineMotion 2 controller.

For details on Vention's Latching Pushbutton Module, refer to the 'Documentation for Previous Product Versions' section at the bottom of this page.

Features

- Includes two momentary pushbuttons
- Connects (daisy chain) with compatible modules
- Configurable address

Technical Specifications

General Specifications

Part Number	CE-MD-014-0001
Certifications	• EN 61000-6-2 (EMC Directive) • EN 55011:2016 (EMC Directive) • EN 63000:2016 (RoHS Directive)
Weight	0.45 kg
Dimensions	46 x 88 x 133.0 mm
Material	• Bottom enclosure: ABS • Top enclosure: Aluminum
Operating Temperature	0 to 40°C
Included in the Box	• 1x Momentary Pushbutton Module (CE-MD-014-1001) • 1x Control Device Extension Cable, 5m (CE-CA-022-5000) • 1x Module Termination Jumper (CE-JP-001-0001) • 1x Mounting Bracket (CE-HW-005-1002) • 2x M8 Drop-in Spring Loaded T-Nut (HW-FN-002-0001) • 2x M8 x 18mm Screw (HW-FN-003-0018)

Momentary Pushbutton Module Physical Interface

Figure 1: Physical interface

Status LED Indicators

Name	LED Color	Indicated (when ON)
POWER	White	24 VDC supplied to module
COMM	Yellow and Blue	RS-485 communication functional
FUSE	Red	Module internal fuse tripped

Pushbuttons (black/white)

Pushbutton type	Momentary
Mechanical life (minimum)	250,000 operations

Connecting to a MachineMotion V2 Controller

Figure 2: Momentary Pushbutton module with MachineMotion V2

CTRL IN Male M12 connector pinout

Pin	Description
Pin 1	24 VDC (input)
Pin 2	Ground (input)
Pin 3	RS-485 A (input)
Pin 4	NRS-485 B (input)
Pin 5	Reserved
Pin 6	Reserved
Pin 7	N/A
Pin 8	Reserved

CTRL OUT Female M12 connector pinout

Pin	Description
Pin 1	24 VDC (output)
Pin 2	Ground (output)
Pin 3	RS-485 A (output)
Pin 4	NRS-485 B (output)
Pin 5	Reserved
Pin 6	Reserved
Pin 7	N/A
Pin 8	Reserved

MQTT Topics

Topic	Message	Type	Description
`devices/push-button-v2/+/available`	`true|false`	READ	true if the device is available/connected
`devices/push-button-v2/+/hw-revision`	`vX`	READ	Hardware Revision
`devices/push-button-v2/+/firmware`	`vX.X`	READ	firmware Revision
`devices/push-button-v2/+/digital-input/0`	`0/1`	READ	State of the black button on top
`devices/push-button-v2/+/digital-input/1`	`0/1`	READ	State of the white button on the bottom
`devices/push-button-v2/+/set-led`	`{"red":INT,"green":INT,"blue":INT}`	WRITE	Sets the LED colors. 0=OFF, 1=ON

MachineLogic Sequence Example

Sequence example to handle a button being pushed and to change the LED color of the device to Green:

Topic for Wait For **Event**:

- If Black Button: devices/push-button-v2/+/digital-input/0
- If White Button: devices/push-button-v2/+/digital-input/1

Message: 1

Topic for Output **Generate Event**:

- devices/push-button-v2/+/set-led

Message: {"red":0,"green":1,"blue":0}

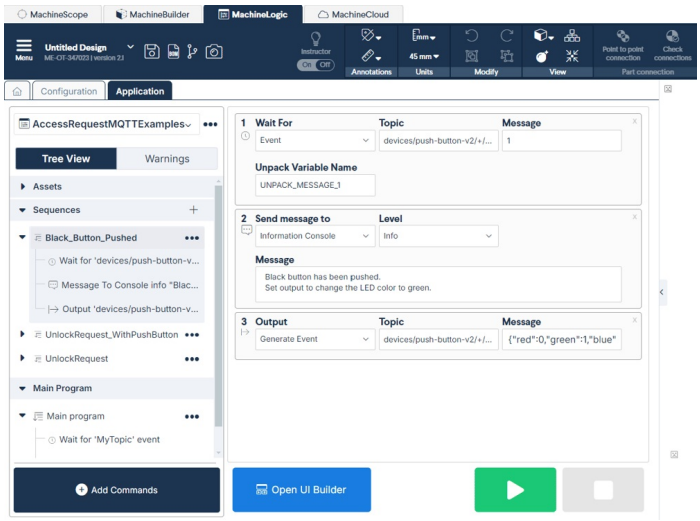

Figure 3: MachineLogic example to handle a button being pushed and to change the LED color of the device

How to replicate the LED color of the Access Request Module on the Momentary Pushbutton

If you are using the Momentary Pushbutton in conjunction with an Access Request Module (CE-SA-017-0001) you can use the example below to ensure that the LEDs have the same behavior. Follow these steps below:

1. Find your **Device ID **as defined in Table 1 below
2. Find your **Serial Number** on the back of the Access Request Module
3. Add your **Device ID** & **Serial Number** in the example below

Plaintext Copy

```
from machinelogic import Machine
import json
from time import sleep

access_request_serial_number = 1110003 # serial number can be found on device label
pushbutton_device_id = 1

m = Machine()
payload = "{}"
old_payload = None

def mirror_access_request_led(topic, message):
    global payload
    print("Access Request LED Change: ", message)
    message = json.loads(message.replace("'","\""))
    # incoming status will be: {"red": INT,"green":INT,"blue":INT,"blink": INT}
    red = message['red'] % 254
    green = message['green'] % 254
    blue = message['blue'] % 254
    payload = json.dumps({"red": red,"green":green,"blue": blue})

m.on_mqtt_event(
f'safety-module-hub/access-request/{access_request_serial_number}/led',
mirror_access_request_led
)

while True:
    if payload != old_payload:
        m.publish_mqtt_event(f'devices/push-button-v2/{pushbutton_device_id}/set-led', payload)
        old_payload = payload
    sleep(0.1)
```

Momentary Pushbutton module address configurations

Below are the valid address configurations that can be used for the Pushbutton module

Valid address configurations

Switches								Module Address
Device ID				Device Type				
1	2	3	4	5	6	7	8	
OFF	OFF	OFF	OFF	ON	OFF	OFF	OFF	Pushbutton Module 1
ON	OFF	OFF	OFF	ON	OFF	OFF	OFF	Pushbutton Module 2
OFF	ON	OFF	OFF	ON	OFF	OFF	OFF	Pushbutton Module 3
ON	ON	OFF	OFF	ON	OFF	OFF	OFF	Pushbutton Module 4
OFF	OFF	ON	OFF	ON	OFF	OFF	OFF	Pushbutton Module 5
ON	OFF	ON	OFF	ON	OFF	OFF	OFF	Pushbutton Module 6
OFF	ON	ON	OFF	ON	OFF	OFF	OFF	Pushbutton Module 7
ON	ON	ON	OFF	ON	OFF	OFF	OFF	Pushbutton Module 8

Table 1: Address Configuration

Documentation for Previous Product Versions

Latching Pushbutton Module

Latching Pushbutton Module Datasheet
Latching Pushbutton Module User Manual