

GPIO Expander – TCA9534A – Trēo™ Module

Module Features

- Texas instruments TCA9534A
- RoHS Compliant
- Software Library
- NightShade Trēo™ Compatible
- Breakout Headers
- Patent Pending



TCA9534A Features

(from Texas instruments)

- 5-V Tolerant I/O Ports
- Three Hardware Address Pins Allow up to Eight Devices on the I2C bus
- Pin-Change HW Interrupt

Applications

- Parallel Communication
- Interface Panels
- Industrial Automation

Trēo™ Compatibility

Electrical

Communication	I2C
Max Current, 3.3V	6mA
Max Current, 5V	0mA

Mechanical

- 45mm x 25mm Outline
- 40mm x 20mm Hole Pattern
- M2.5 Mounting Holes

Description

The TCA9534A Trēo™ Module is a GPIO Expander module that features Texas instruments' TCA9534A GPIO Expander. It provides eight additional I/O pins which can be used for many applications including parallel output, buttons, switches, or indicators. This module is a part of the NightShade Treo system, patent pending.

Table of Contents

1	Summary	2
2	What is Trēo™?	2
3	Electrical Characteristics	2
4	Electrical Schematic	3
5	Mechanical Outline	4
6	Example Arduino Program	5
7	Library Overview (C++ & Python)	6

1 Summary

The TCA9534A GPIO expander module provides 8 GPIO pins which can be used in countless ways. Each pin can be configured and read individually or the whole port can be accessed at one time by using either the *Pin* or *Port* commands. A hardware interrupt is generated on any rising or falling transition on a pin that is set to input mode. The interrupt is cleared when the port is read or when the pin is returned to the original state.

2 What is Trēo™?

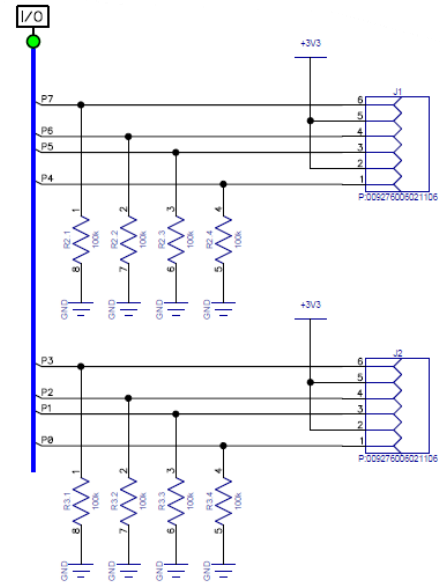
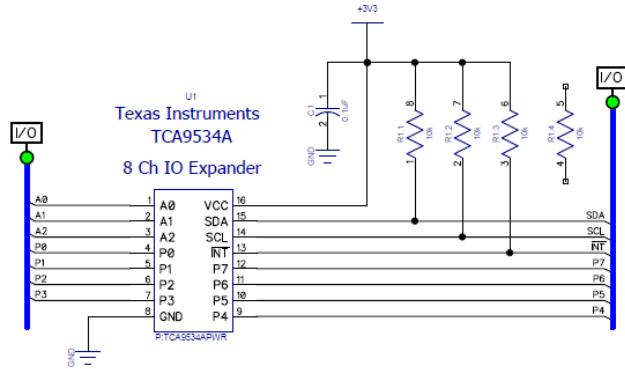
NightShade Trēo is a system of electronic modules that have standardized mechanical, electrical, and software interfaces. It provides you with a way to quickly develop electronic systems around microprocessor development boards. The grid attachment system, common connector/cabling, and extensive cross-platform software library allow you more time to focus on your application. Trēo is supported with detailed documentation and CAD models for each device.

Learn more about Trēo [here](#).

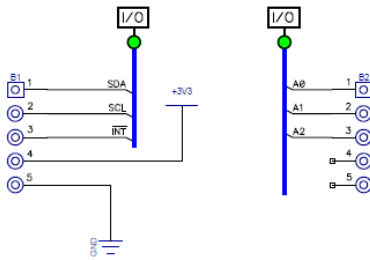
3 Electrical Characteristics

	Minimum	Nominal	Maximum
Voltages			
$V_{I/O}$ (SDA, SCL, INT)	-0.3V	-	3.6V
$V_{3.3V}$	3.1V	3.3V	3.5V
V_{5V}	4.8V	5.0V	5.2V
I/O Specifications			
V_{output} (P0 – P7)	0V	-	3.3V
V_{input} (P0 – P7)	-0.5V	-	5.5V
$I_{I/O \text{ pin, max}}$	-10mA (source)	-	25mA (sink, <65°C)
$I_{I/O \text{ total, max}}$	-80mA (source)	-	200mA (sink)
I2C Slave Address			
SJ1-SJ3 Open (Default)		0x3F	
Alt. Address (Soldered = 0)		B 0 1 1 1 [SJ3] [SJ2] [SJ1]	
Operating Temperature			
	-25°C	-	+85°C

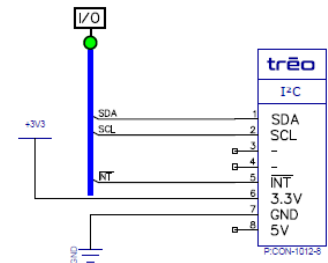
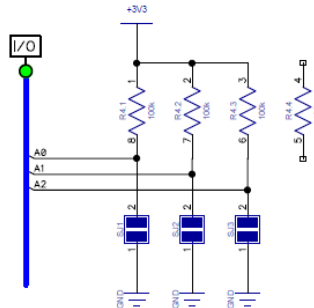
4 Electrical Schematic



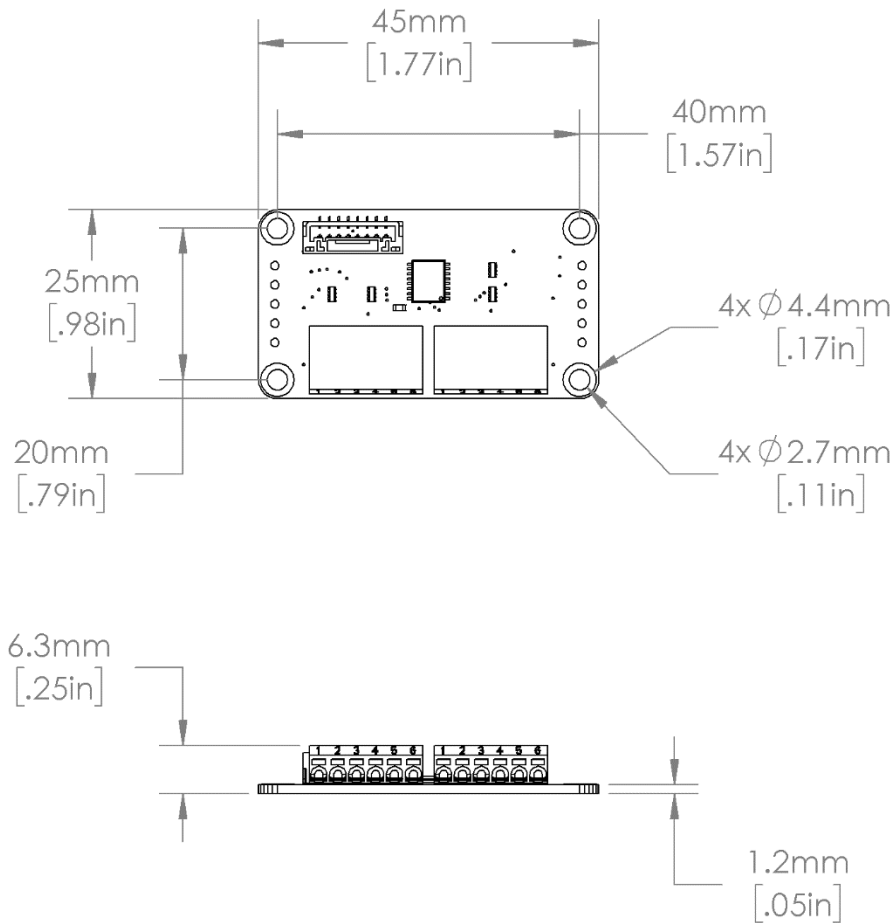
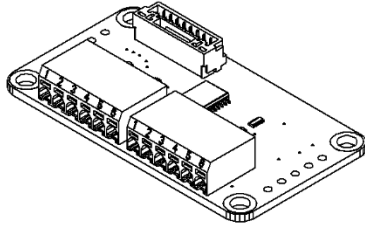
Breakout Headers



Address Configuration



5 Mechanical Outline



6 Example Arduino Program

```
/*
TCA9534_GPIOExpander - NightShade_Treo by NightShade Electronics

This sketch demonstrates the functionality of the
NightShade Trēo TCA9534 GPIO expander module.
(NSE-1147-1) It prints the digital value being input to
each GPIO channel to Serial at 115200 baudrate.

Created by Aaron D. Liebold
on February 15, 2021

Links:
NightShade Trēo System: https://nightshade.net/treo
Product Page: https://nightshade.net/product/treo-gpio-expander-tca9534a/

Distributed under the MIT license
Copyright (C) 2021 NightShade Electronics
https://opensource.org/licenses/MIT
*/

// Include NightShade Treo Library
#include <NightShade_Treo.h>

// Declare Objects
NightShade_Treo_TCA9534A sensor(1);

void setup() {
  sensor.begin();
  Serial.begin(115200);

  sensor.setPortMode(0xFF); // Set all pins to input
}
void loop() {
  // Read input state
  uint8_t input = sensor.readPortInput();

  // Print input with leading zeros
  for (int x = 7; x > 0; --x) {
    if ( !((1 << x) & input) ) {
      Serial.print('0');
    } else {
      break;
    }
  }
  Serial.println(input, BIN);

  delay(1000);
}
```

7 Library Overview (C++ & Python)

C++ Class

```
NightShade_Treo_<MODULE_NAME> <classObject>();
```

Python Module

```
<classObject> = NightShade_Treo.<MODULE_NAME>()
```

7.1 Constructors

NightShade_Treo_TCA9534A(int port, uint8_t slaveAddress, uint32_t clockSpeed)

Creates a TCA9534A object.

Arguments:

port	Integer of the I2C port used (e.g. 0 = "/dev/i2c_0")
slaveAddress	7-bit slave address
clockSpeed	Desired clock speed for the bus

Returns:

Nothing

NightShade_Treo_TCA9534A(int port)

Creates a TCA9534A object assuming the default slave address and clock speed.

Arguments:

port	Integer of the I2C port used. (e.g. 0 = "/dev/i2c_0")
------	---

Returns:

Nothing

7.2 Methods

begin()

Initializes the TCA9534A.

Arguments:

None

Returns:

Error	0 = Success
-------	-------------



setPinMode(int pin, int mode)

Sets the direction of a GPIO pin.

Arguments:

pin	GPIO Pin Number
mode	0: Output 1: Input

Returns:

Error	0 = Success
-------	-------------

setPinOutput(int pin, int output)

Sets the output state of a GPIO pin.

Arguments:

pin	GPIO Pin Number
mode	0: Low 1: High

Returns:

Error	0 = Success
-------	-------------

setPinPolarity(int pin, int polarityInverted)

Sets the input polarity of a GPIO pin.

Arguments:

pin	GPIO Pin Number
mode	0: Normal 1: Inverted

Returns:

Error	0 = Success
-------	-------------

readPinMode(int pin)

Reads the current mode of a pin.

Arguments:

pin	GPIO Pin Number
-----	-----------------

Returns:

pinMode	0: Output 1: Input
---------	-----------------------



setPortOutput(uint8_t portOutput)

Sets the output state of the I/O pins for the whole port. Setting a bit (1) makes the corresponding pin high while a cleared bit (0) is makes the pin low.

Arguments:

portOutput B7 – B0 set the output state of P7 – P0

Returns:

Error 0 = Success

setPortPolarity(uint8_t portPolarityInversion)

Sets the input polarity of the I/O pins for the whole port. Setting a bit (1) inverts the input value of the corresponding pin.

Arguments:

portOutput B7 – B0 set the input polarity of P7 – P0

Returns:

Error 0 = Success

readPortMode()

Reads the port direction setting register.

Arguments:

None

Returns:

portMode Register (uint8_t)

readPortOutput()

Reads the output state setting register.

Arguments:

None

Returns:

portOutput Register (uint8_t)

readPortInput()

Reads the port input state register.

Arguments:

None

Returns:

portInput Register (uint8_t)



readPortPolarity()

Reads the input polarity setting register.

Arguments:

None

Returns:

portPolarity Register (uint8_t)