

## Number Pad – TCA9534A – Trêo™ Module

### Module Features

- TCA9534A GPIO Expander
- RoHS Compliant
- Software Library
- NightShade Trêo™ Compatible
- Breakout Headers

### Features

- 16 Key Input Pad
- I2C Interface
- Interrupt Generation

### Applications

- User Input
- Security Locks
- Process Control

### Trêo™ Compatibility

#### Electrical

<b>Communication</b>	I2C
<b>Max Current, 3.3V</b>	4mA
<b>Max Current, 5V</b>	0mA

#### Mechanical

- 35mm x 35mm Outline
- 30mm x 30mm Hole Pattern
- M2.5 Mounting Holes



### Description

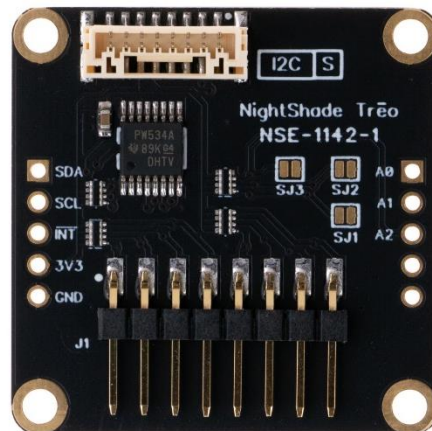
The Number Pad Trêo™ Module features Texas Instruments TCA9534A GPIO expander to interface with a membrane number pad. The module provides 16 input keys and it can generate a HW interrupt on a keypress. This module is a part of the NightShade Treo system, patent pending.

### Table of Contents

1	Summary .....	2
2	What is Trêo™? .....	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 Number Pad module is initialized with the `begin()` method and then it can be monitored for a keypress by polling the `buttonPressed()` method or by monitoring the hardware interrupt pin, which is asserted on any keypress. When a key is being pressed, it can be read with the `read()` method which returns the ASCII value of the key being pressed. The `connectorFlipped()` method can be used to reverse the wiring of the keypad connector so that it may be attached in either orientation.



## 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.

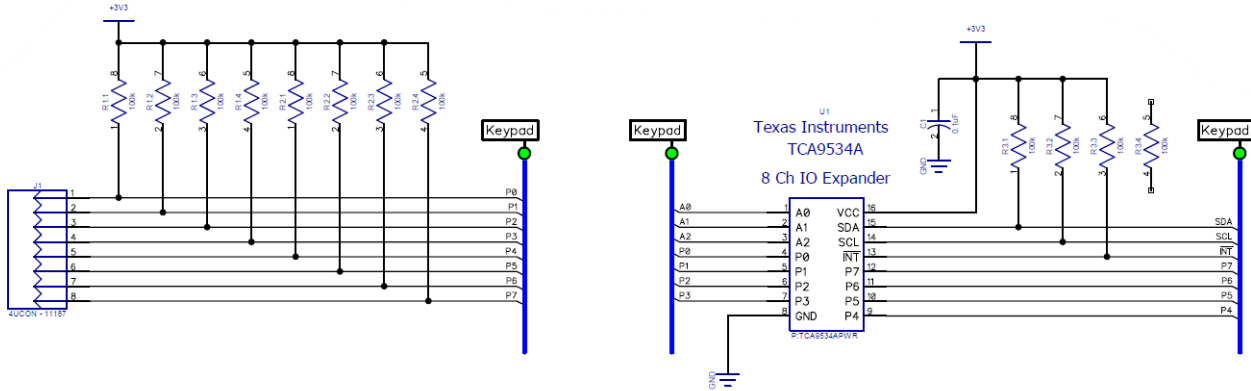
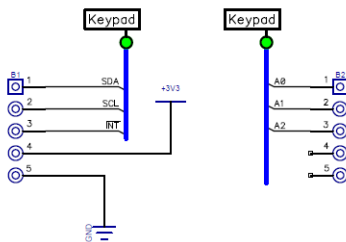
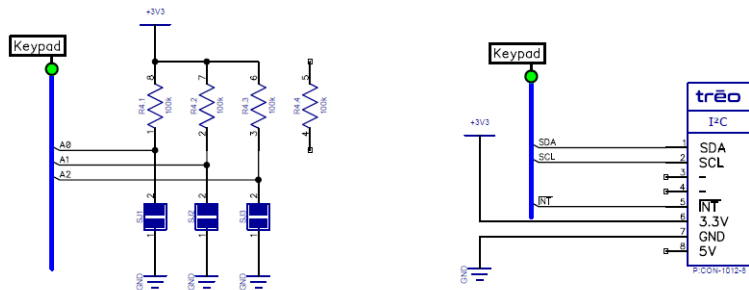
Learn more about Trēo [here](#).

## 3 Electrical Characteristics

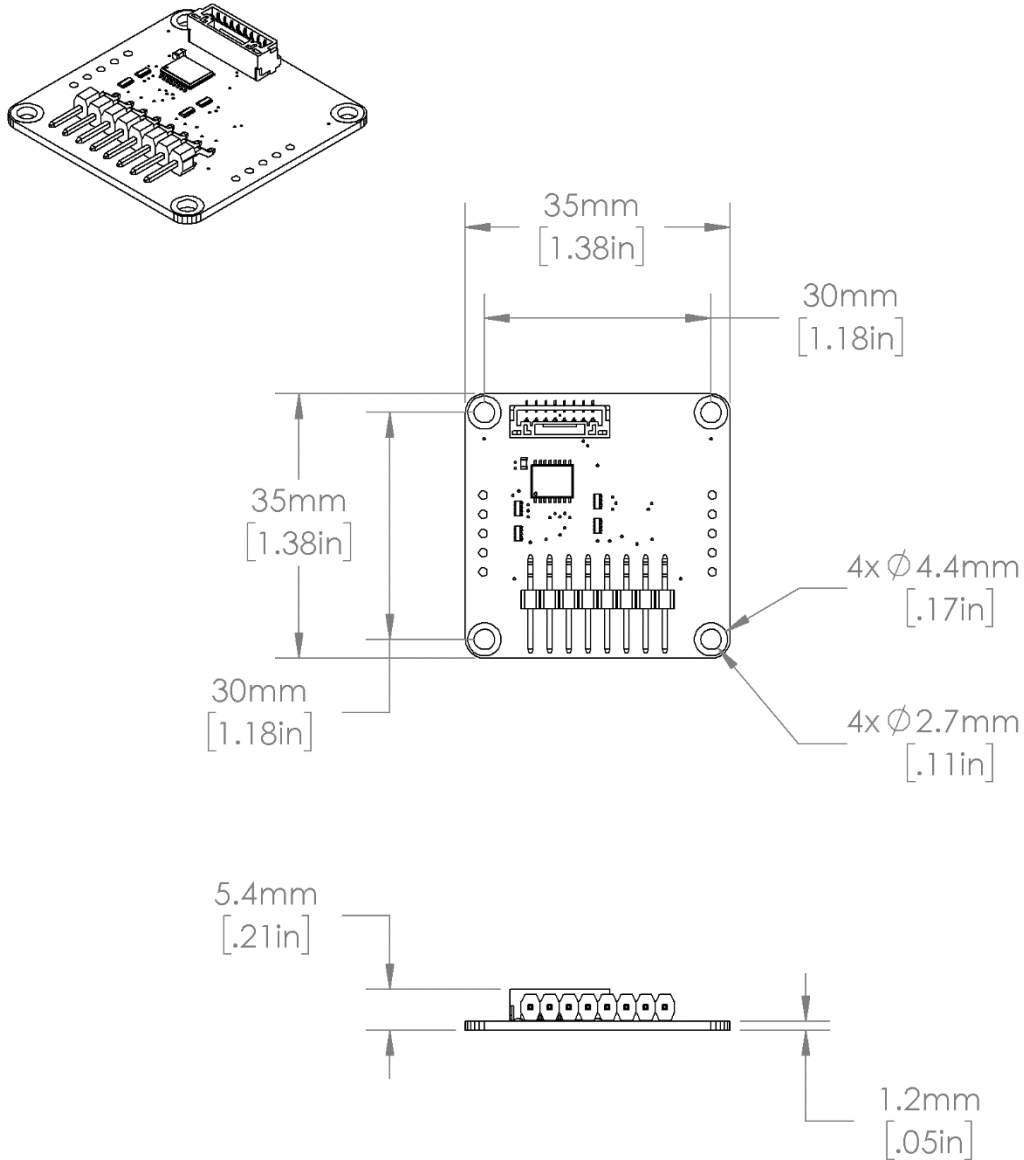
	Minimum	Nominal	Maximum
<b>Voltages</b>			
$V_{i/o}$ (SDA, SCL, INT)	-0.3V	-	3.6V
$V_{3.3V}$	3.1V	3.3V	3.5V
<b>I2C Slave Address</b>			
SJ1-SJ3 Open (Default)		0x3F	

Alt. Address (Soldered = 0)		B 0 1 1 1 [S <sub>J</sub> 3] [S <sub>J</sub> 2] [S <sub>J</sub> 1]	
Operating Temperature	-25°C	-	+85°C

## 4 Electrical Schematic


**Breakout Headers**

**Address Configuration**


## 5 Mechanical Outline





## 6 Example Arduino Program

```
/******  
NumberPad - NightShade_Treo by NightShade Electronics  
  
This sketch demonstrates the functionality of the  
NightShade Trēo Number Padmodule. (NSE-1142-1) It prints  
each pressed character to Serial at 115200 baudrate.  
  
Created by Aaron D. Liebold  
on February 17, 2021  
  
Links:  
NightShade Trēo System: https://nightshade.net/treo  
Product Page: https://nightshade.net/product/treo-number-pad-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_NumberPad keypad(1);  
  
void setup() {  
  Serial.begin(115200);  
  keypad.begin();  
}  
  
void loop() {  
  if (keypad.buttonPressed()) {  
    char key = keypad.read();  
    if (key == '*') Serial.write('\n'); // '*' is interpreted as a carriage return  
    else Serial.write(key);  
    while (keypad.buttonPressed());  
  }  
}
```



## 7 Library Overview (C++ & Python)

### C++ Class

```
NightShade_Treo_NumberPad <classObject>();
```

### Python Module

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

### 7.1 Constructors

#### NightShade\_Treo\_NumberPad(int port, uint8\_t slaveAddress, uint32\_t clockSpeed)

Creates a NumberPad 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\_NumberPad(int port)

Creates a NumberPad 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 number pad module. Returns the ASCII value of the key being pressed or -1 is no key is being pressed. Use the

Arguments:

None

Returns:

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



**read()**

Returns the current keypress.

Arguments:

None

Returns:

Keypress (ASCII byte)

**buttonPressed()**

Returns a Boolean value indicating if a button is being pressed. This method should be used for polling operations as it is faster than the read() method.

Arguments:

None

Returns:

Error                      0 = Success

**connectorFlipped(int isFlipped)**

The keypad connector is not polarized and therefore can be plugged into the connector two ways. Use the method to flip the connector wiring if the output does not match the keys being pressed.

Arguments:

isFlipped                      true/false

Returns:

Error                      0 = Success