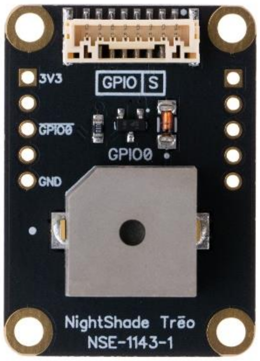
Active Buzzer - AST501Q - Trēo™ Module

Module Features

- Mallory AST501Q
- RoHS Compliant
- Software Library
- NightShade Trēo™ Compatible
- Breakout Headers



AST501Q Features

(from Mallory)

- 2.4kHz Frequency
- 87dB Sound Pressure @ 10cm

Applications

- Audible interface feedback
- System Alert/Alarm

Trēo™ Compatibility

Electrical

Communication	GPIO
Max Current, 3.3V	18mA
Max Current, 5V	0mA

Mechanical

- 25mm x 35mm Outline
- 20mm x 30mm Hole Pattern
- M2.5 Mounting Holes

Description

The AST501Q Trēo™ Module is an Active Buzzer module that features Mallory's AST501Q Active Buzzer. It generates its own oscillating frequency of 2.4kHz, so that it must only be turned on and off. The host does not need to generate a waveform. 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

This module is operated using the NightShade_Treo_DigitalOutput library. The module operated with the on(), off(), toggle(), and set() methods. The current state can be found with the read() method.

2 What is Trēo™?

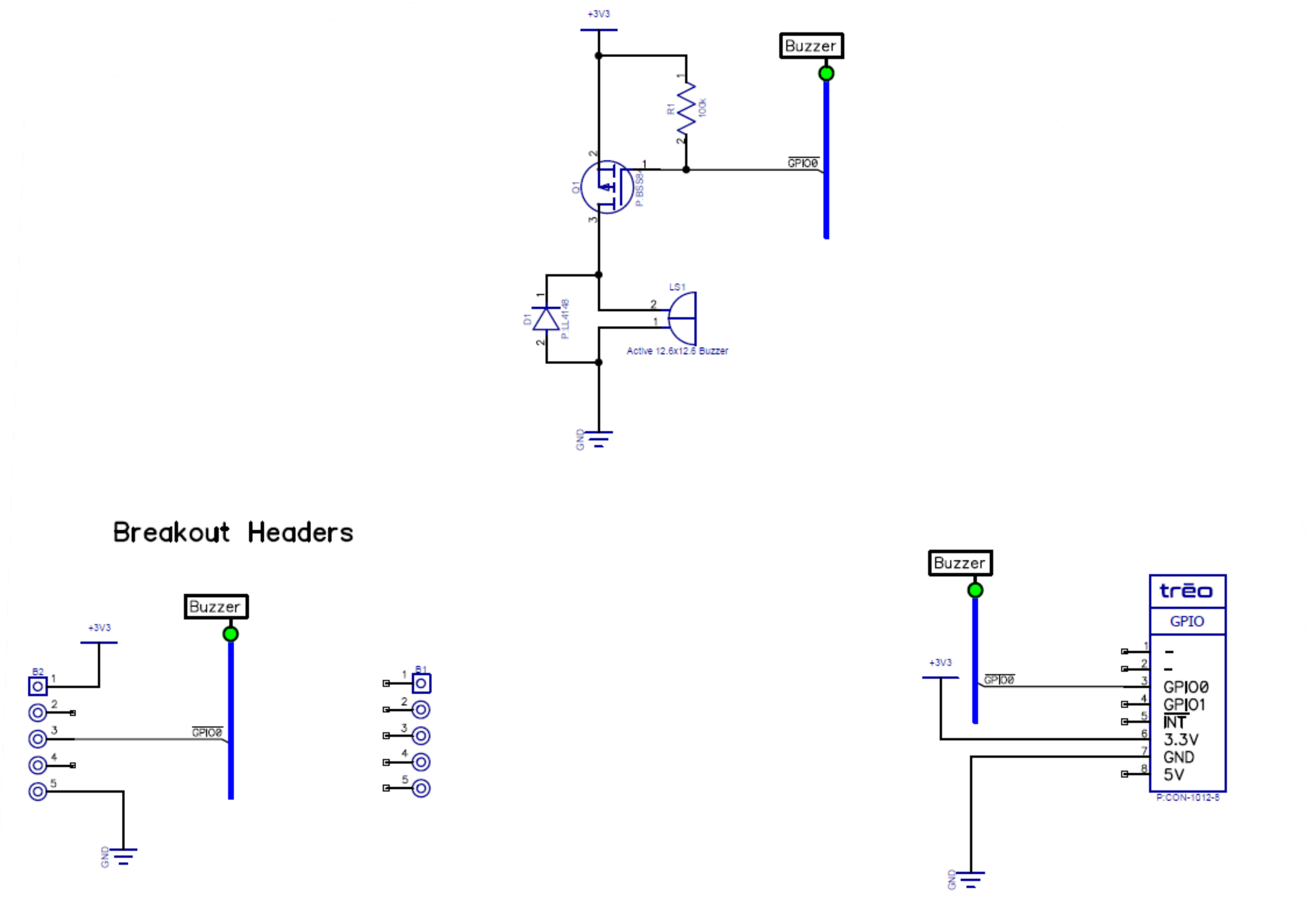
NightShade Trēo is a system of electronic modules that have standardized mechanical, electrical, and software interfaces. It provides you with a way to quickly develop electronic systems around microprocessor development boards. The grid attachment system, common connector/cabling, and extensive cross-platform 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.

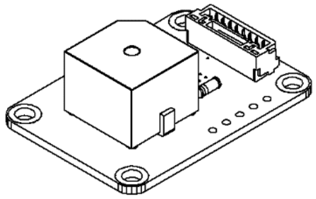
3 Electrical Characteristics

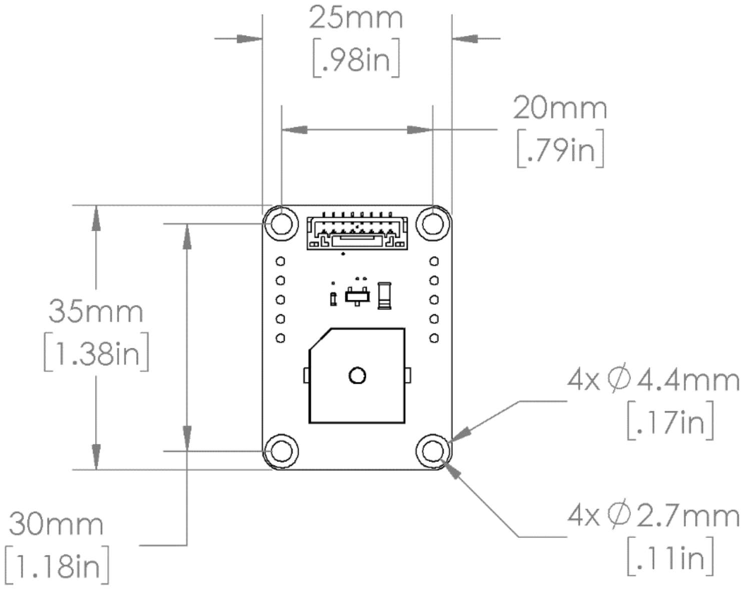
	Minimum	**Nominal**	**Maximum**
Voltages			
$V_{i/o}$ (GPIO)	-0.3V	-	3.6V
$V_{3.3V}$	3.1V	3.3V	3.5V
Specifications			
Resonant Frequency	-	2400Hz	-
SPL @ 10cm	-	87dB	-
Operating Temperature	-20°C	-	+70°C

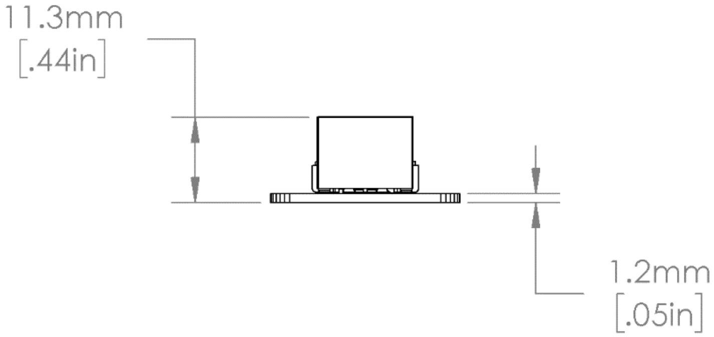
4 Electrical Schematic



5 Mechanical Outline







6 Example Arduino Program

```
/**********************************************************
  DigitalOutput - NightShade_Treo by NightShade Electronics

  This sketch demonstrates the functionality of the
  NightShade Trēo digital output modules like LEDs and
  relays.

  Created by Aaron D. Liebold
  on February 15, 2021

  Links:
  NightShade Trēo System: https://nightshade.net/treo

  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 (Dual Output Device - GPIO0: D5, GPIO1: D4)
NightShade_Treo_DigitalOutput out0(5);
NightShade_Treo_DigitalOutput out1(4);

void setup() {

}

void loop() {
  out0.on();
  delay(500);
  out0.off();
  delay(500);
  out1.on();
  delay(500);
  out1.off();
  delay(500);
}
```

7 Library Overview (C++ & Python)

C++ Class

```
NightShade_Treo_DigitalOutput <classObject>();
```

Python Module

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

7.1 Constructors

NightShade_Treo_DigitalOutput(int gpioPin)

Creates a DigitalOutput object.

Arguments:

gpioPin	Integer of the GPIO pin

Returns:

Nothing

7.2 Methods

on()

Turns the GPIO output to the ON state.

Arguments:

None

Returns:

Error	0 = Success

off()

Turns the GPIO output to the OFF state.

Arguments:

None

Returns:

Error	0 = Success

toggle()

Toggles the GPIO output state.

Arguments:

None

Returns:

Error	0 = Success

set(int enable)

Sets the GPIO to the ON (enabled) or OFF (disabled) state.

Arguments:

enable	true/false

Returns:

Error	0 = Success

read()

Returns the current GPIO state.

Arguments:

None

Returns:

GPIO state (int)	0: Off 1: On