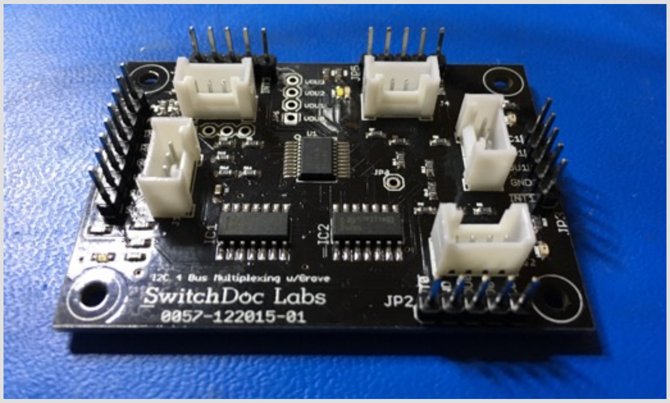
Product Specification

SwitchDoc Labs

# I2CMux Grove Board

**0057-GRV4I2CMux-DSBT/SF**



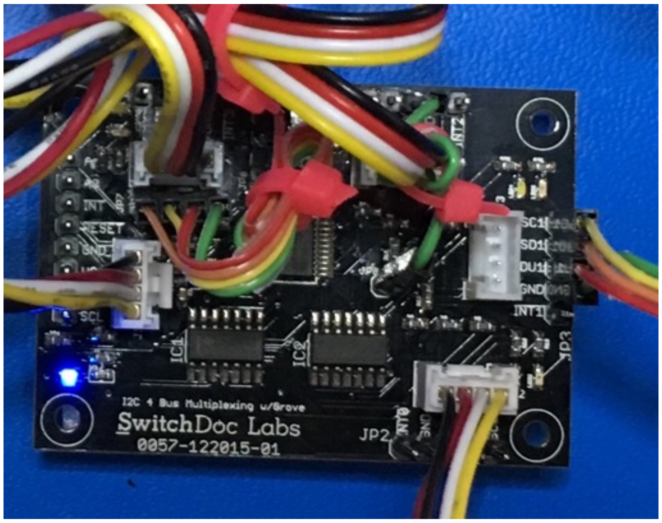
The **I2CMux Grove Board** is an easy to use 4 Channel I2C Multiplexer.

The I2C Mux Breakout Board is a quad bidirectional translating switch controlled via the I2C bus. The SCL/SDA controlling fans out to four downstream channels. Any individual channel or combination of channels can be selected via I2C. It is based on the TCA9545A. It can be used with Grove and non-Grove I2C devices.

Features and Benefits:

- Converts one I2C bus (on Pi or Arduino) to 4 separate I2C buses
- Allows using same I2C addresses for many sensors
- Status LEDs for each I2C channel
- Works with Arduino and Raspberry Pi
- Grove Compatible Connectors
- Pin Headers for Non-Grove Devices
- Four Active-Low Interrupt Inputs
- Two I2C Address Selector
- Channel Selection via I2C Bus, in Any Combination
- Allows Voltage-Level Translation Between 1.8-V, 2.5-V, 3.3-V, and 5-V Buses
- No Glitch on Power-Up
- Power-Supply Range of 1.65 V to 5.5 V
- Software Drivers for Arduino and Raspberry Pi
- Low Cost
- Quantity Discounts Available
- Immediate Availability

## Introduction



The I2C Mux Breakout Board is a quad bidirectional translating switch controlled via the I2C bus with Grove connectors. The host SCL/SDA controls fans out to four downstream channels.

At SwitchDoc Labs, we love data. And we love I2C devices.

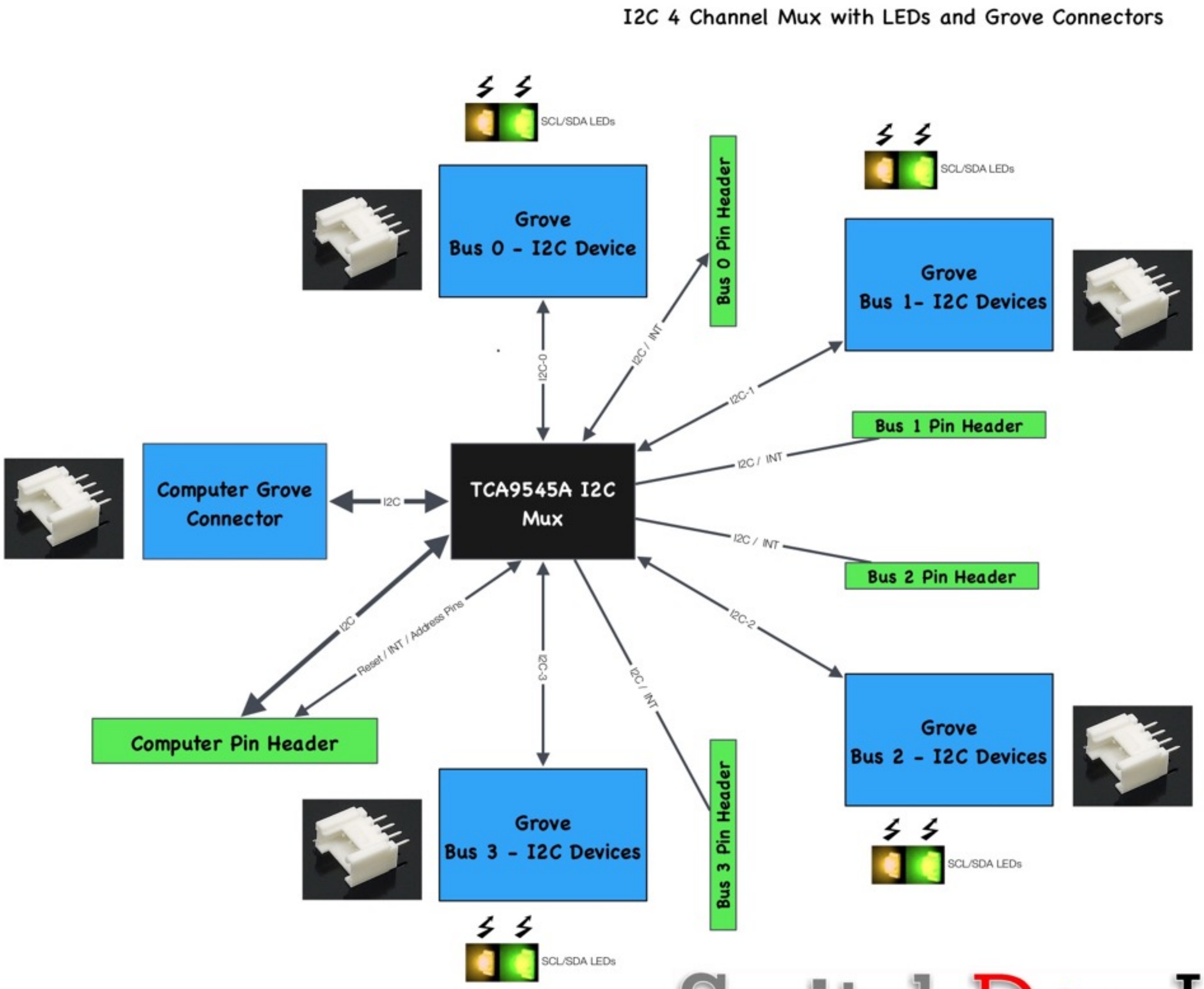
We like to gather the data using lots of I2C devices on our computers and projects. Project Curacao has a total of 12 devices and SunRover will have over 20 and will require one I2C bus just for controlling the motors. In addition, we run into conflicts with addressing on the I2C device. Since there are no standards, sometimes multiple devices will have the same address, such as

0x70 and you are just out of luck in running both of them on the same I2C bus without a lot of jimmy rigging.

What is the solution for this? **It's an I2C controlled 4 I2C bus multiplexer!**

## How To Use

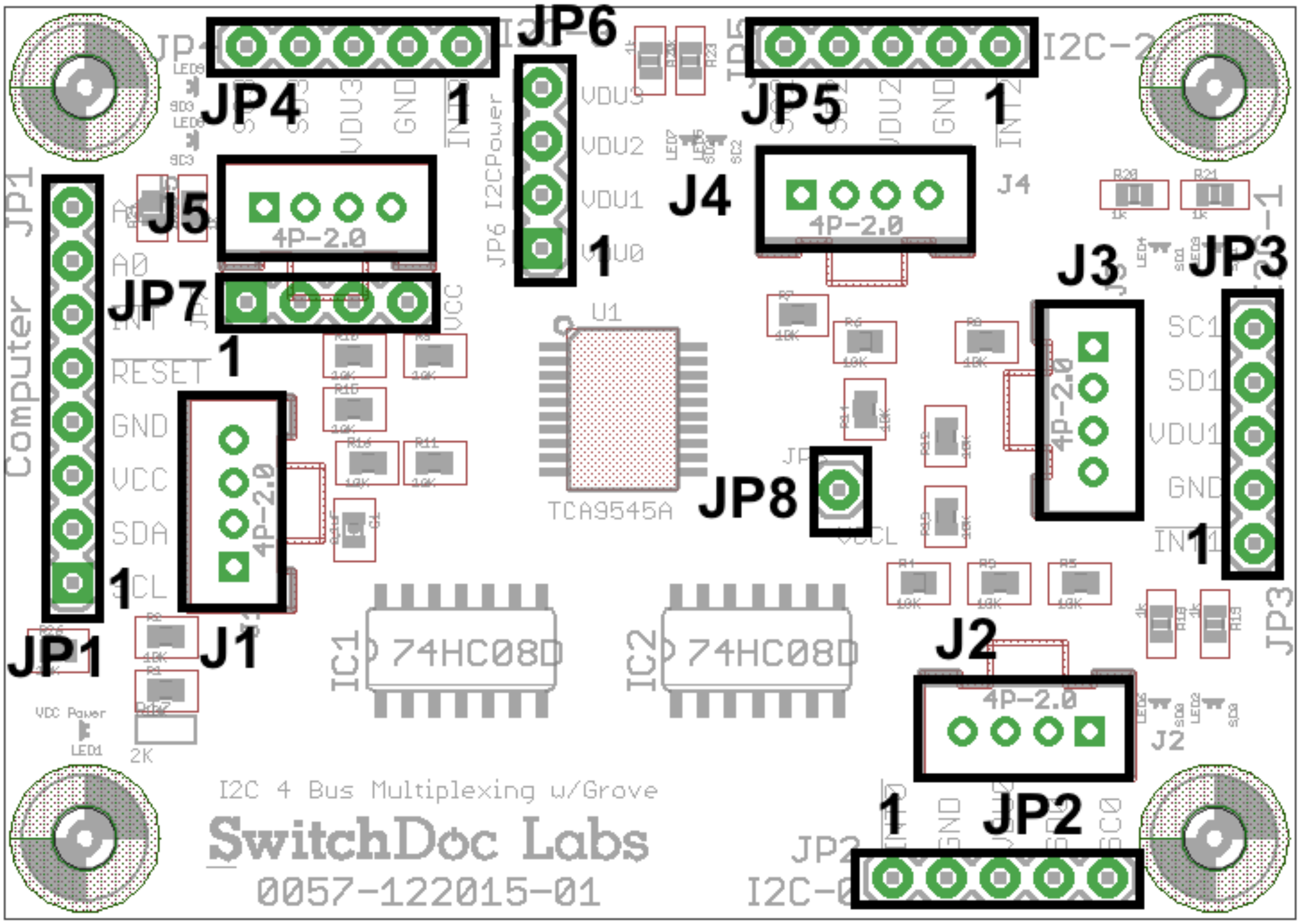
The I2C Mux Breakout Board is a quad bidirectional translating switch controlled via the I2C bus. The SCL/SDA controlling fans out to four downstream channels.



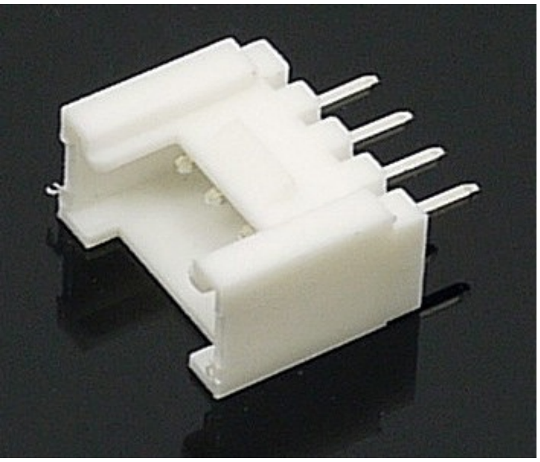
To use the Grove I2C Mux Breakout Board, you connect the I2C bus up to an Arduino or Raspberry Pi and then connect the additional I2C busses as shown below. The main I2C bus can be 3.3V or 5.0V as well as each of the multiplexed bus can be individually selected as 3.3V or 5.0V I2C busses. There are three things to note when you wire up your I2C Mux Breakout Board.

- Each of the four multiplexed I2C busses must be connected to either 3.3V or 5.0V. Each multiplexed bus is individually powered. See JP6 below.
- Unlike other SwitchDoc Labs Breakout Boards, there is a 10K Ohm Pullup on the main I2C lines to VCC and 10K Pullups on each of the Multiplexed I2C busses to their respective power supplies.
- Connect JP8 to the highest voltage you are using on the host or mux'd busses. 5V will always work.

## I2CMux Grove Board Pinout



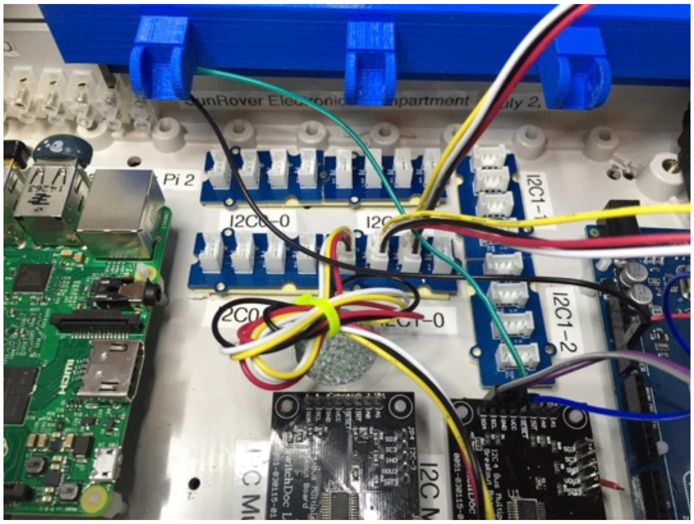
## What is a Grove Connector?

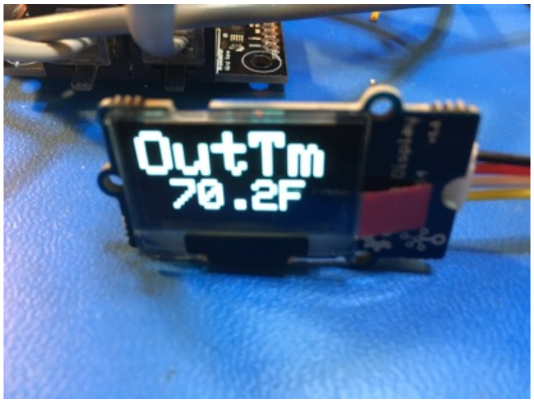


The way we have been wiring I2C connections before just didn't work for large projects. Basically, we used to put the I2C bus to screw terminals or snap down connectors and then ran wires to each device. This would not work for complex projects. Because of this, we moved to Grove connectors [ http://www.seeedstudio.com/wiki/Grove_System].

There are dozens of Grove I2C sensors out now.   Many different kinds of cables and I2C Hubs.

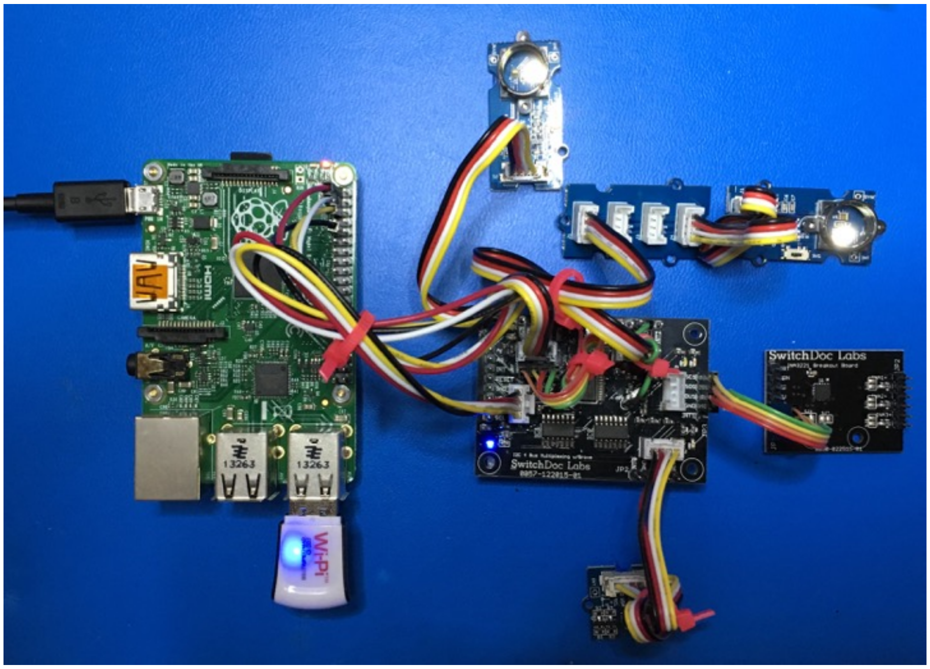
We quickly found the connectors and their respective cables very useful. With the large selection of Grove I2C devices available, we decided to include a Grove connector on all our future I2C boards.   The white connectors on the 4 Channel I2C Mux board picture at the top are Grove connectors for easy, non-soldered connections to the I2C bus. The picture below shows  the SunRover robot built using Grove connectors for the 8 different I2C busses in the robot.  A Grove OLED display is shown underneath the picture.





## Connecting to Grove Connectors

There are a number of Grove shields and Hats for Raspberry Pi and Arduino devices.
Grove I2C Connectors are keyed so they can not be plugged in incorrectly. Below is the I2CMux board hooked up with both Grove and non-Grove devices.



## Wiring List

For non Grove devices (you can buy Grove shields and Hats for Raspberry Pi and Arduino devices), I2CMux can be used with the following wiring lists. With all of the below devices, you need to wire up power supplies for the individual I2C busses (for example, if they are all 3.3V and you are connecting VCC to 3.3V on the Pi, all of the connections on JP6 should be connected to VCC - the section on JP6 below).

## Raspberry Pi (A/B/A+/B+/Pi 2/Zero)

| Signal Name | Raspberry Pi A/B/A+/B+/Pi 2 | I2CMux Grove Board |
|---|---|---|
| **SDA** | I2C1_SDA (GPIO/3) | SDA (JP1/2) |
| **SCL** | I2C1_SCL (GPIO/5) | SCL (JP1/1) |
| **GND** | GND (GPIO/6) | GND (JP1/4) |
| **Power** | 3.3V (GPIO/1) | VCC (JP1/3) |
| **Reset** | N/C | RESET' (JP1/5) - on board pullup |
| **Int** | GPIO 18 (GPIO/12) | INT' (JP1/6) |
| **Address Select 0** | N/C | A0 (JP1/7) |
| **Address Select 1** | N/C | A1 (JP1/8) |

## Arduino Uno

| Signal Name | Arduino Uno | I2CMux Grove Board |
|---|---|---|
| **SDA** | ADC4/SDA (ANALOG IN/A4) | SDA (JP1/2) |
| **SCL** | ADC5/SCL (ANALOG IN/A5) | SCL (JP1/1) |
| **GND** | GND (POWER/GND) | GND (JP1/4) |
| **Power** | 5.0V (POWER/5V) | VCC (JP1/3) |
| **Reset** | N/C | RESET' (JP1/5) -on board pullup |
| **Int** | INT0 (DIGITAL/2) | INT' (JP1/6) |
| **Address Select 0** | N/C | A0 (JP1/7) |
| **Address Select 1** | N/C | A1 (JP1/8) |

## Ardunio Mega 2560

| Signal Name | Arduino Mega 2560 | I2CMux Grove Board |
|---|---|---|
| **SDA** | SDA (COMMUNICATIONS 20) | SDA (JP1/2) |
| **SCL** | SCL (COMMUNICATIONS 21) | SCL (JP1/1) |
| **GND** | GND (POWER/GND) | GND (JP1/4) |
| **Power** | 5.0V (POWER/5V) | VCC (JP1/3) |
| **Reset** | 5.0V (POWER/5V) | RESET’ (JP1/5) |
| **Int** | INT4 PWM/2) | INT’ (JP1/6) |
| **Address Select 0** | N/C | A0 (JP1/7) |
| **Address Select 1** | N/C | A1 (JP1/8) |

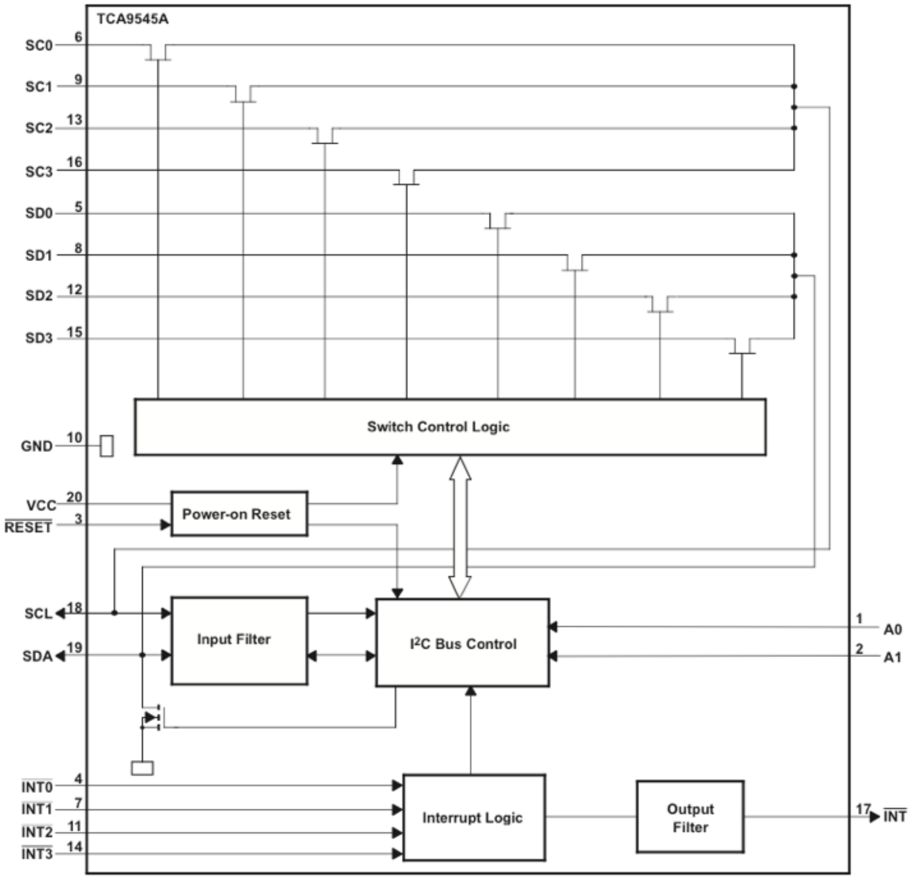
# Theory of Operation

## TCA9545A IC

For more information, see the full TCA9545A Specification at: http://www.ti.com/lit/ds/symlink/tca9545a.pdf

The TCA9545A is a quad bidirectional translating switch controlled via the I2C bus. The SCL/SDA upstream pair fans out to four downstream pairs, or channels. Any individual SCn/SDn channel or combination of channels can be selected, determined by the contents of the programmable control register. Four interrupt inputs (INT3’–INT0’), one for each of the downstream pairs, are provided. One interrupt (INT’) output acts as an AND of the four interrupt inputs.

An active-low reset (RESET’) input allows the TCA9545A to recover from a situation in which one of the downstream I2C buses is stuck in a low state. Pulling RESET low resets the I2C state machine and causes all the channels to be deselected, as does the internal power-on reset function.



The pass gates of the switches are constructed such that the VCC terminal can be used to limit the maximum high voltage, which will be passed by the TCA9545A. This allows the use of different bus voltages on each pair, so that 1.8-V, 2.5-V, or 3.3-V parts can communicate with 5-V parts, without any additional protection. External pull-up resistors (to the IC, they are already present on the I2CMux Grove Board) pull the bus up to the desired voltage level for each channel. All I/O terminals are 5.5 V tolerant.

## Operating Values

| | | Min | Normal | Max | Unit |
|---|---|---|---|---|---|

| | | | | | |
|---|---|---|---|---|---|
| VCC | Supply Voltage | 2.7 | | 5.5 | V |
| VDU0, VDU1, VDU2, VDU3 | Multiplexed I2C Bus Supply Voltages | 1.8V | | 5.5 | V |

## Pin Locations

Physical dimensions of board:  62mm x 43mm x 12mm(max).

**I/O Key:**

**I - Digital Input**
**O - Digital Output**
**A - Analog**

## Pin Functions

### JP1 - Computer Side Pins

Input / Output Control Lines for I2CMux Grove Board

| NAME | PIN | I/O | DESCRIPTION |
|---|---|---|---|
| SCL | JP1 / 1 | I | Serial bus clock line; open-drain input.  Tied to VCC with 10K Ohm Pullup |
| SDA | JP1 / 2 | I/O | Serial bus data line; open-drain input/output.  Tied to VCC with 10K Ohm Pullup |
| VCC | JP1 / 3 | A | Power for the I2CMux Grove Board.  Use 3.3V with Raspberry Pi and 5.0V with Arduino |
| GND | JP1 / 4 | A | GND |
| RESET' | JP1 / 5 | I | Reset Board; open-drain input.  Tied to VCC with 10K Ohm Pullup |
| INT' | JP1 / 6 | O | Interrupt from Board; open-drain output. |
| A0 | JP1 / 7 | I | A0 Address Input.  10K Pullup to VCC.  I2C Address range from 0x70 - 0x73.  Default 0x73 |
| A1 | JP1 / 8 | I | A1 Address Input.  10K Pullup to VCC.   I2C Address range from 0x70 - 0x73.  Default 0x73 |

### JP2 - Multiplexed I2C Bus 0

Multiplexed I2C Bus 0.  All buses can have different power supplies

| NAME | PIN | I/O | DESCRIPTION |
|---|---|---|---|
| INT0' | JP2 / 1 | I | Open Drain Input for I2C Bus 0 Input.  Tied to VDU0 with 10K Pullup |
| GND | JP2 / 2 | A | GND |
| VDU0 | JP2 / 3 | A | Power Supply for I2C Bus 0.  Tie to 3.3V or 5.0V. |
| SD0 | JP2 / 4 | I/O | Serial bus data line.  open-drain input/output.  Tied to VDU0 with 10K Pullup |
| SC0 | JP2 / 5 | O | Serial bus clock line; open-drain output.  Tied to VDU0 with 10K Pullup |

## JP3 - Multiplexed I2C Bus 1

Multiplexed I2C Bus 1.  All buses can have different power supplies

| NAME | PIN | I/O | DESCRIPTION |
|---|---|---|---|
| INT1’ | JP3 / 1 | I | Open Drain Input for I2C Bus 1 Input.  Tied to VDU1 with 10K Pullup |
| GND | JP3 / 2 | A | GND |
| VDU1 | JP3 / 3 | A | Power Supply for I2C Bus 1.  Tie to 3.3V or 5.0V. |
| SD1 | JP3 / 4 | I/O | Serial bus data line.  open-drain input/output.  Tied to VDU1 with 10K Pullup |
| SC1 | JP3 / 5 | O | Serial bus clock line; open-drain output.  Tied to VDU1 with 10K Pullup |

## JP5 - Multiplexed I2C Bus 2   (NOTE:  JP5 IS BUS 2!)

Multiplexed I2C Bus 2.  All buses can have different power supplies

| NAME | PIN | I/O | DESCRIPTION |
|---|---|---|---|
| INT2’ | JP5 / 1 | I | Open Drain Input for I2C Bus 2 Input.  Tied to VDU2 with 10K Pullup |
| GND | JP5 / 2 | A | GND |
| VDU2 | JP5 / 3 | A | Power Supply for I2C Bus 2.  Tie to 3.3V or 5.0V. |
| SD2 | JP5 / 4 | I/O | Serial bus data line.  open-drain input/output.  Tied to VDU2 with 10K Pullup |
| SC2 | JP5 / 5 | O | Serial bus clock line; open-drain output.  Tied to VDU2 with 10K Pullup |

## JP4 - Multiplexed I2C Bus 3 (NOTE:  JP4 is BUS 3!)

Multiplexed I2C Bus 3.  All buses can have different power supplies

| NAME | PIN | I/O | DESCRIPTION |
|---|---|---|---|
| INT3' | JP4 / 1 | I | Open Drain Input for I2C Bus 3 Input.  Tied to VDU3 with 10K Pullup |
| GND | JP4 / 2 | A | GND |
| VDU3 | JP4 / 3 | A | Power Supply for I2C Bus 3.  Tie to 3.3V or 5.0V. |
| SD3 | JP4 / 4 | I/O | Serial bus data line.  open-drain input/output.  Tied to VDU3 with 10K Pullup |
| SC3 | JP4 / 5 | O | Serial bus clock line; open-drain output.  Tied to VDU3 with 10K Pullup |

## JP6 - I2C Auxiliary Power Pads

These pads allow you to hook up power supplies for each of the 4 I2C Muxed busses.   Depending on your design, you may be powering the I2C devices on each bus from another power supply.  In that case, you would leave these pads empty.   Otherwise, you connect the appropriate power supply for each bus.  For example, if you want all busses at 3.3V and the host is a Raspberry Pi, you  would hook up VCC (from the Pi @ 3.3V) .  You can mix 3.3V and 5.0V I2C busses by connecting the pads on JP6 to their respective power supplies.

| NAME | PIN | I/O | DESCRIPTION |
|---|---|---|---|
| VDU0 | JP6 / 1 | A | Tied to VDU0 on JP2 and power on Grove J2 |
| VDU1 | JP6 / 2 | A | Tied to VDU1 on JP3 and power on Grove J3 |
| VDU2 | JP6 / 3 | A | Tied to VDU2 on JP5 and power on Grove J4 |
| VDU3 | JP6 / 4 | A | Tied to VDU3 on JP4 and power on Grove J5 |

## JP7 - Auxiliary VCC connections

These connectors are all tied to VCC  on JP1 / 3, the VCC from the host interface (usually the Raspberry Pi or Arduino)

| NAME | PIN | I/O | DESCRIPTION |
|---|---|---|---|
| VCC | JP7 / 1 | A | Tied to VCC on Host Interface (JP1 / 3) |
| VCC | JP7 / 2 | A | Tied to VCC on Host Interface (JP1 / 3) |
| VCC | JP7 / 3 | A | Tied to VCC on Host Interface (JP1 / 3) |
| VCC | JP7 / 4 | A | Tied to VCC on Host Interface (JP1 / 3) |

## JP8 - Status LED Power Supplies - VCCL

| NAME | PIN | I/O | DESCRIPTION |
|---|---|---|---|
| VCCL | JP8 / 1 | A | Power Supply for buffers to all of the Status LEDs on each I2C Channel.   Connect to the highest power supply being used on your busses (5.0V will always work).   Leave unconnected to greatly reduce the current draw of the Status LEDs. |

# Grove Connectors

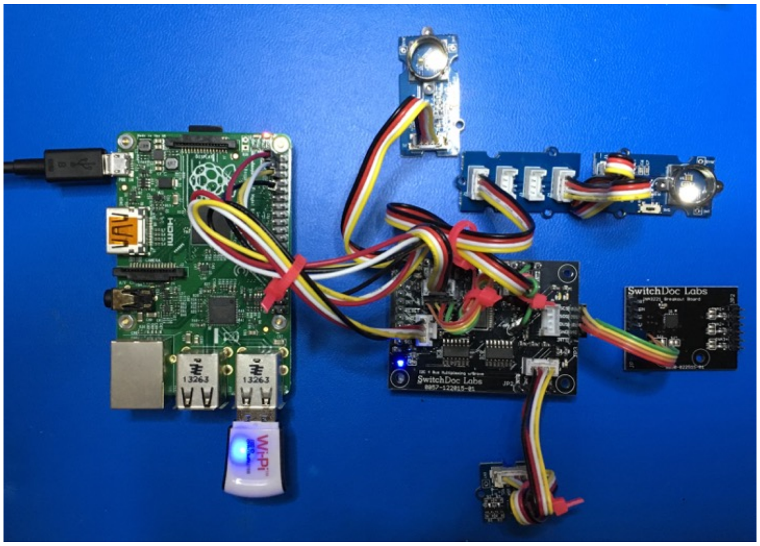
## J1 - Host Interface Grove Connector (Usually Arduino or Raspberry Pi)

## J2 - I2C Bus 0 Grove Connector

## J3 - I2C Bus 1 Grove Connector

## J4 - I2C Bus 2 Grove Connector

## J5 - I2C Bus 3 Grove Connector



## Software for Arduino and Raspberry Pi

SwitchDoc Labs developed this pure Python TCA9545A I2CMux Raspberry Pi library and the Arduino Library and have posted them on the SwitchDoc Labs Repository on github.com

The Raspberry Pi Pure Python software is here:  https://github.com/switchdoclabs/SDL_Pi_TCA9545

The Arduino Software is here:  https://github.com/switchdoclabs/SDL_Arduino_TCA9545A

Using the Arduino libraries and the test software show the following result.  The test setup is to connect an additional I2C device to Bus0 - in this case a SwitchDoc Labs INA3221 Breakout Board at address 0x40 on Bus0.

```
-----------------------------
------------------------------
SDA_Arduino_TCA9545_Test
Reading all four I2C Buses
------------------------------
------------------------------

------------------------------
------------------------------
Bus 0 Control Register:1
Scanning...
I2C device found at address 0x40  !
I2C device found at address 0x73  !
done

------------------------------
Bus 1 Control Register:2
Scanning...
I2C device found at address 0x73  !
done

------------------------------
Bus 2 Control Register:4
```

```
Scanning...
I2C device found at address 0x73  !
done

------------------------------
Bus 3 Control Register:8
Scanning...
I2C device found at address 0x73  !
done
```

Repeat the above test connecting the I2C Device to Bus1, Bus2 and Bus3

The I2C device (the INA3221 in this case) will move from bus to bus.