CTI

Community Training Initiative

support: forums.ni.com
add-ons: vipm.io
resources: gcentral.org

Simple Grove Examples

Version v1.0, 2026-04-09

Table of Contents

Overview

Course Objective

This material is aimed at providing examples employing the CTI firmware, mainly implemented using the Grove platform.

The focus is to provide enough toys to enable a prospective engineer or hobbyist to start building things.

Target Audience

Some Electronics knowledge would be useful, but hopefully the wealth of material available for these platforms can fill in the gaps. Videos will be provided to offer some real-world experience.

Set-up Instructions

Hardware Requirements

- Raspberry Pi Pico 2040
  - Pihut UK
  - Farnell UK#
- Grove Shield For Pico
  - PiHut UK
  - Farnell UK
- Radxa X4
  - Radxa
- Grove Shield for Radxa X4
  - github Page for KiCAD files

Download the LabVIEW Drivers

Download the latest drivers

- cti-drivers-lv-visa

Download and install the CTI Pico firmware (UF2)

1. Download the latest CTI Pico firmware (UF2)
   - UF2 Releases
2. Load firmware on the Pico by pressing the bootsel button and powering on the Pico.

> On a Radxa you could run this script instead (mainly because the Bootsel button is a bit fiddly!)...
>
> *Example 1. Reset Bash Script*
>
> ```
> #! /bin/bash
>
> gpioset gpiochip0 17=1
>
> gpioset gpiochip0 7=1
>
> sleep 1
>
> gpioset gpiochip0 17=0
> ```

```
gpioset gpiochip0 7=0
```

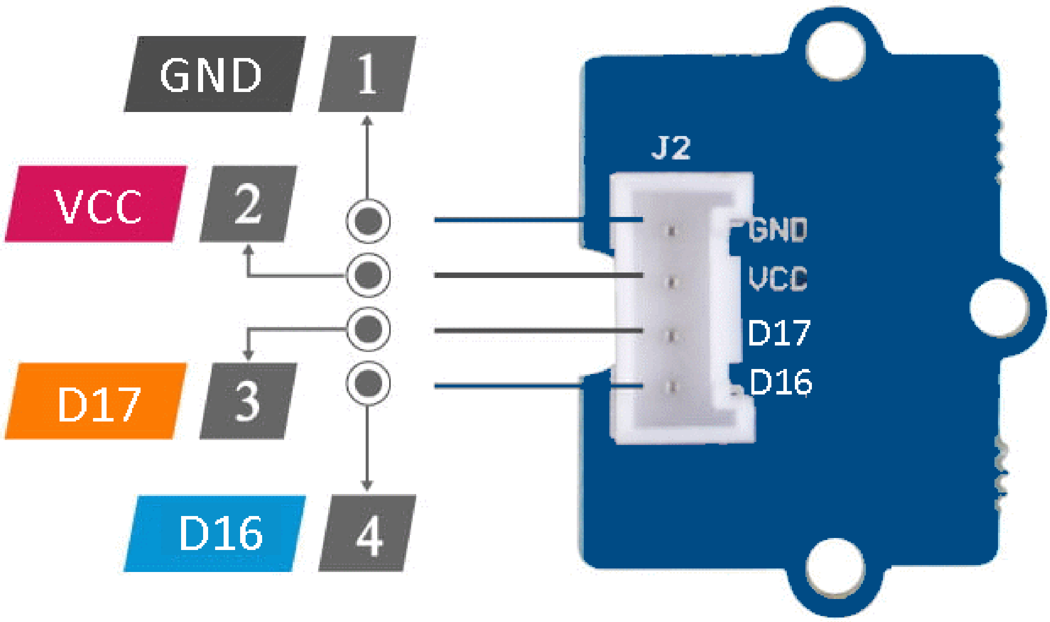
What is Grove

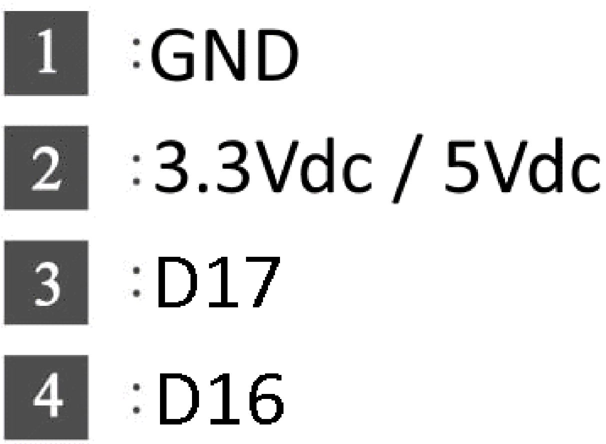
Overview

Grove is an inexpensive connection system for hobbyist micro-controllers (Pico, RPi and Arduino) and comes with a whole eco-system of boards and hats. This makes it very convenient for trainers and hobbyist engineers.

SEEED Grove Website

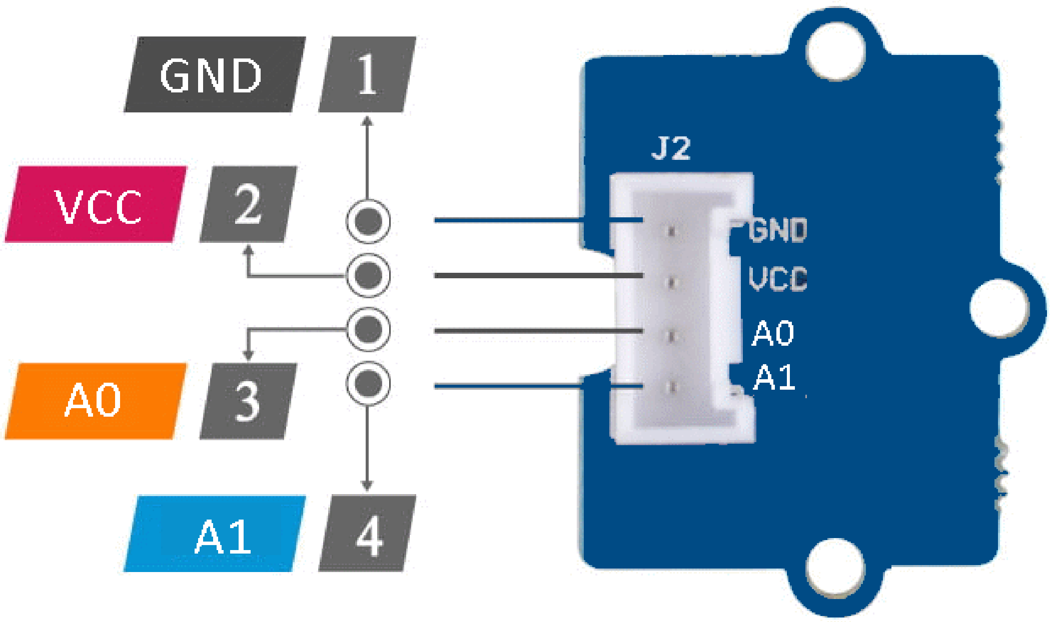
Digital Connectors

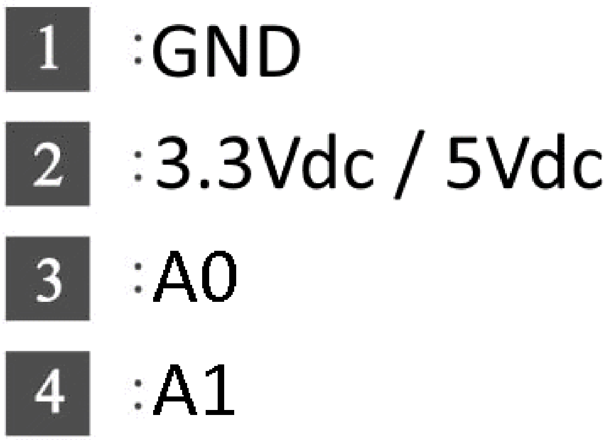




Connectors D16,D18,D20 are used for our examples and are wired as shown in Figure 1

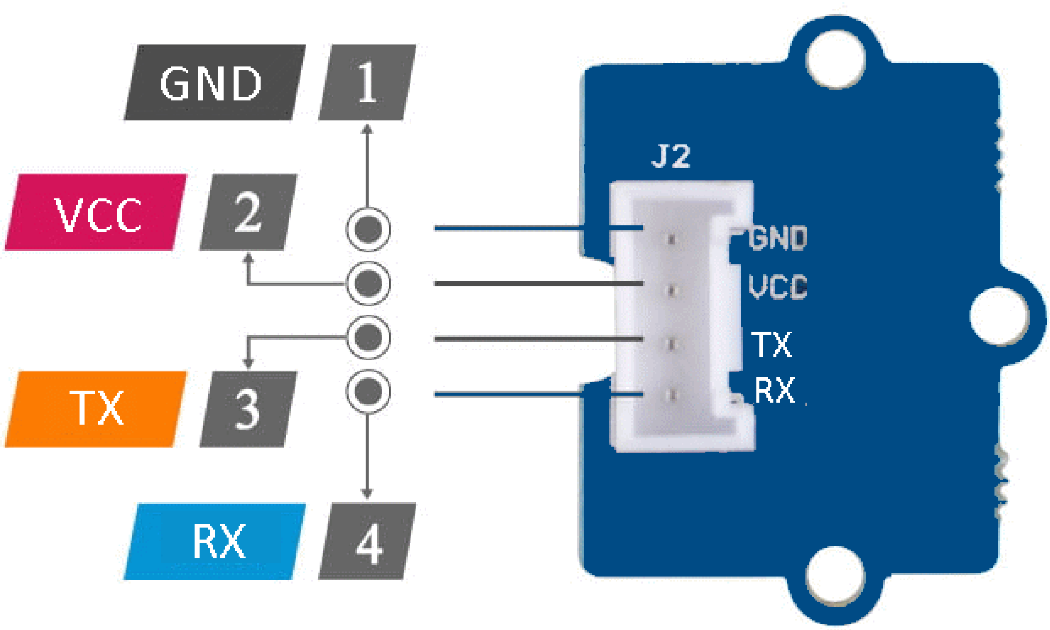
Analog Connectors

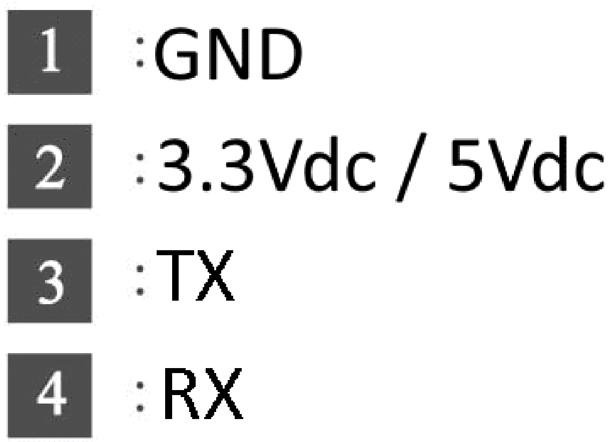




Connectors A0,A1,A2 are used for our examples and are wired similar to Figure 2

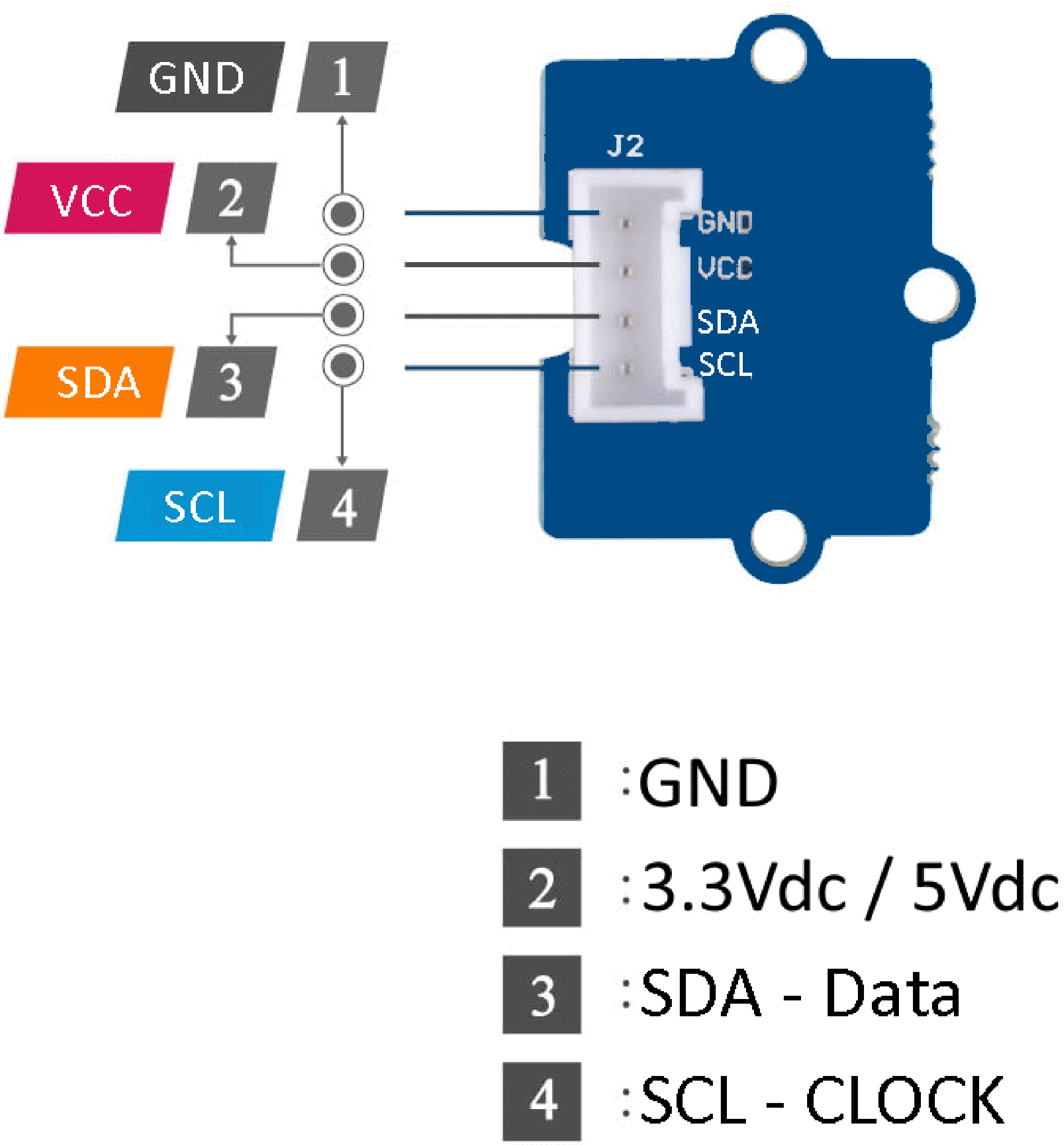
UART Connectors





Connectors UART0 and UART1 are used for our examples and are wired as shown in Figure 3

I2C Connectors



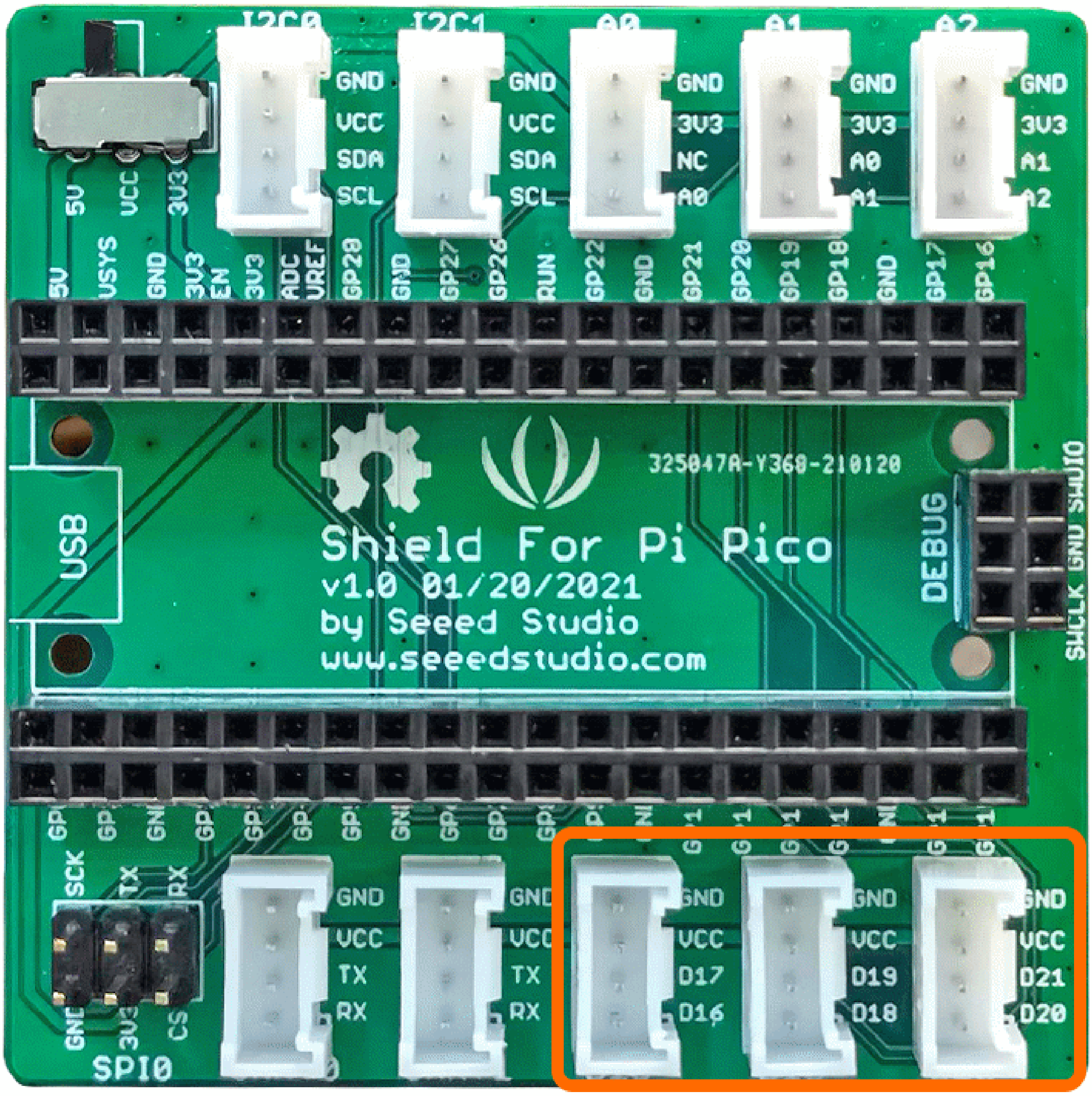
Connectors I2C0 and I2C1 are used for our examples and are wired as shown in Figure 4

Digital Output Examples

Standard Digital Output

Overview

A digital output can be thought of as the signal to an LED or a relay. These examples use the GPIO Pins [D16,D17],[D18,D19],[D20,D21] that correspond to connectors D16,D18 and D20 on the grove shield.



Hardware Details

Grove LED

Grove Relay

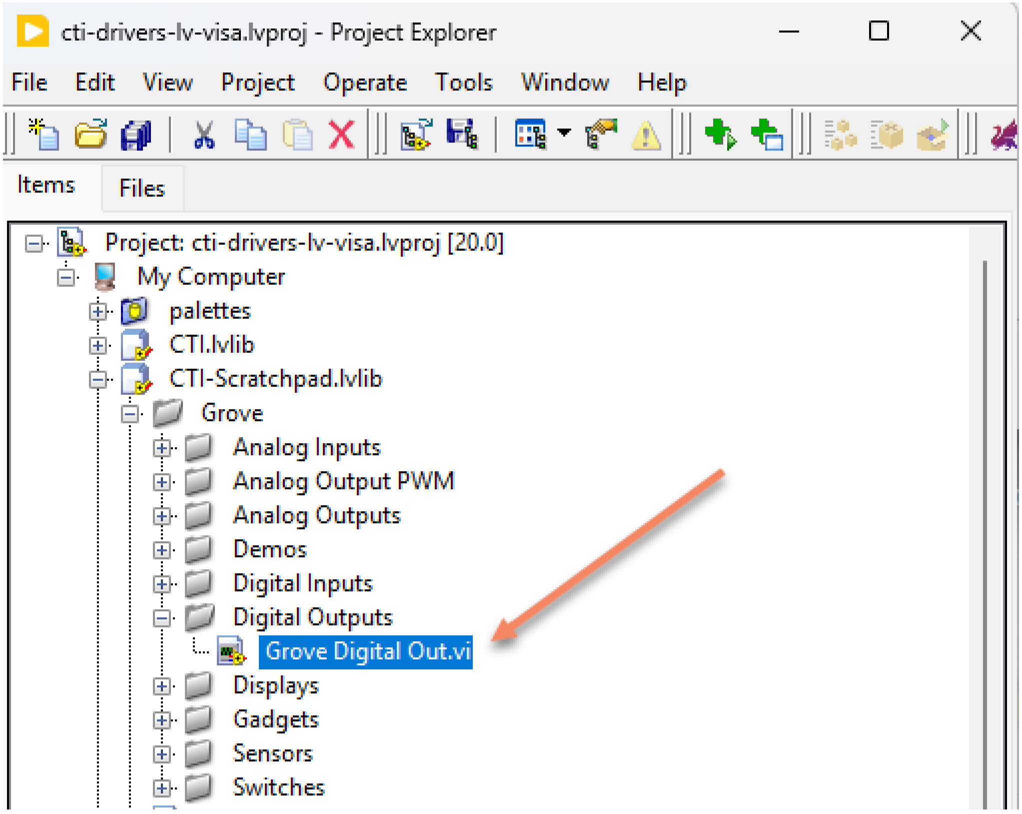
Demo Video

Here is a video that shows the set-up and running of the LED and Relay Examples

▶ https://www.youtube.com/watch?v=w0F2TlSiCmE *(YouTube video)*

Example Code

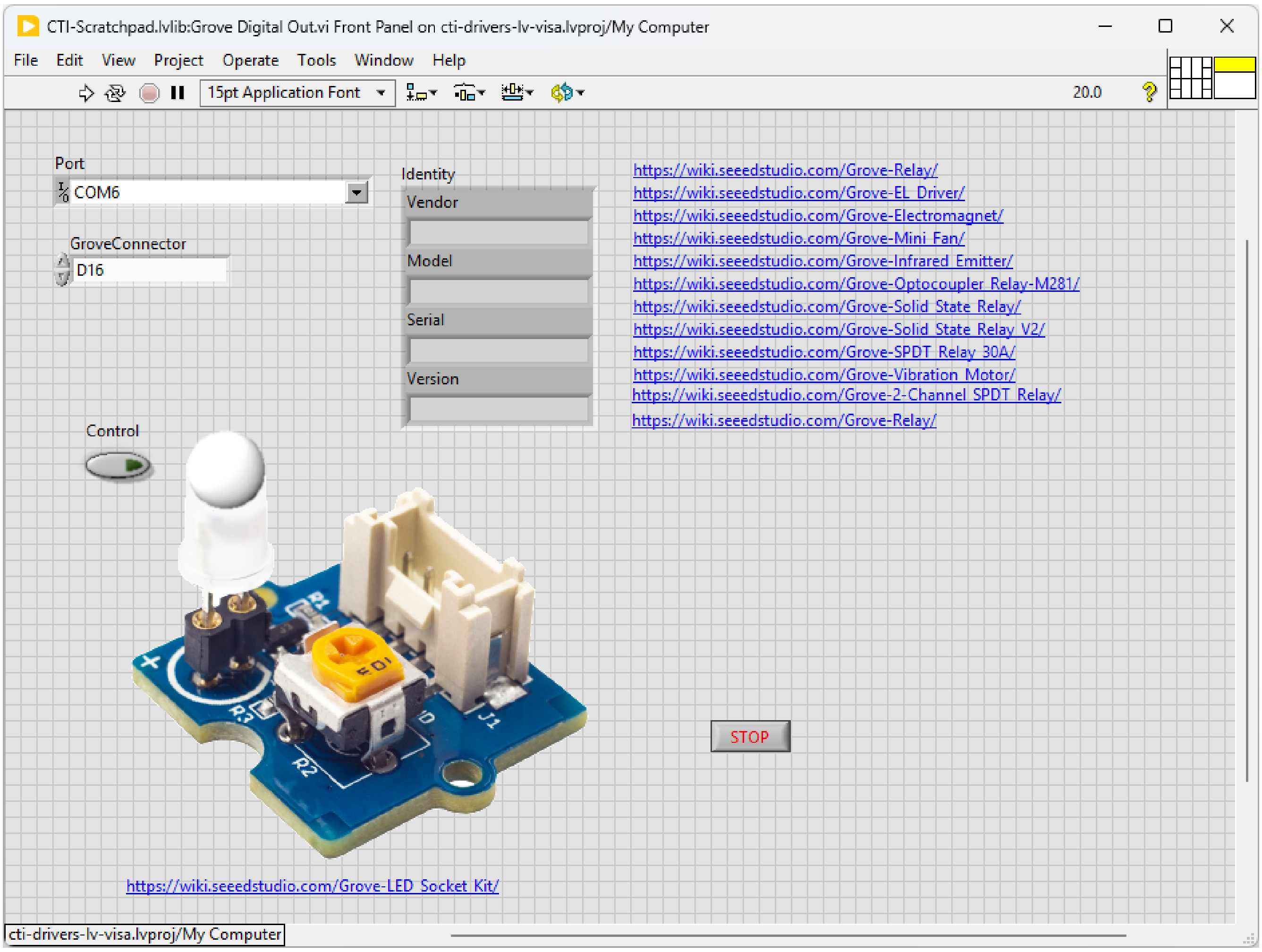
Navigate to >>Scratchpad>>Grove>>Digital Inputs>>**Grove Digital Out.vi**



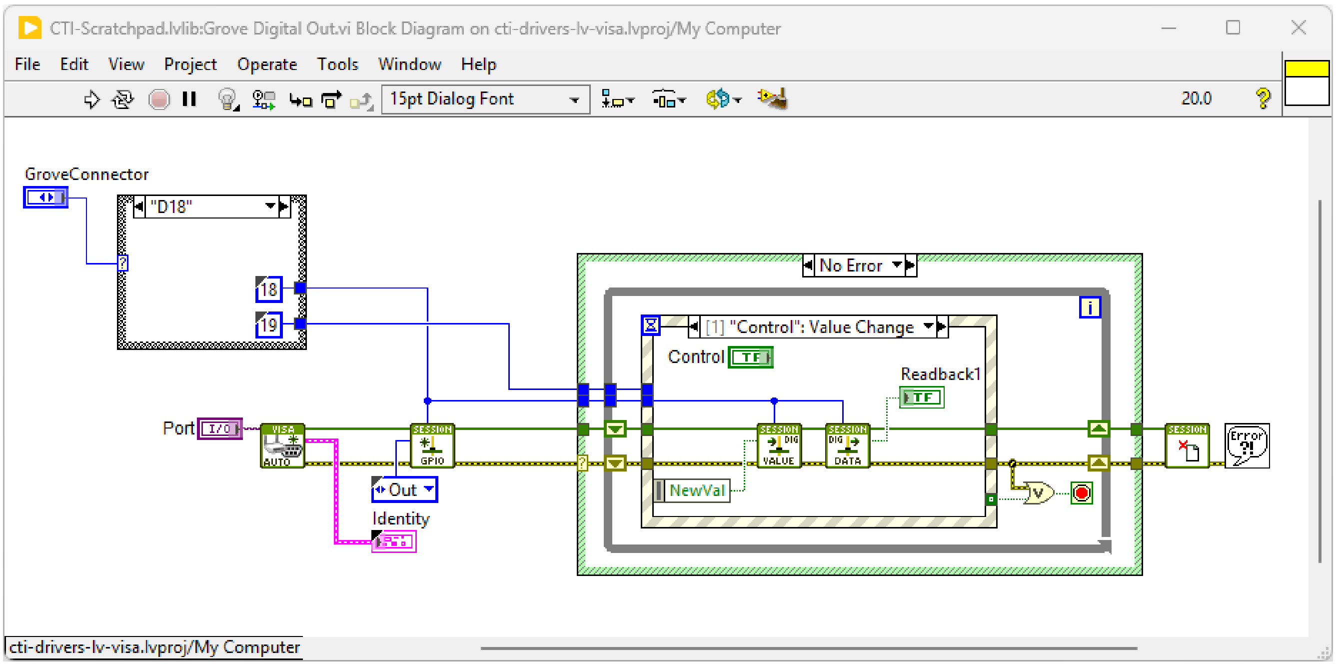
This particular VI uses the Green LED. Select the port for the connected Pico and the Grove connector that the board is plugged into. Press the run arrow.

When you press the [Control Button] the LED should toggle on and off.

Links to various other similar boards are also on the Front Panel.



On the block diagram you can see that the selected Grove connector dictates the GPIO Pin and we then set the pin to [Out]. Next we loop round and event structure and use the Control Value Change Event to set the LED. Pressing Stop will fire the Stop event and exit the loop. The video also shows the same VI being used to switch a relay on and off.



SPST Relay Board / MOSFET Switch

Overview

This is essentially the same VI as the LED output VI, but with different graphics. There are some demo videos just to show wiring and operation.

Hardware Details

Grove Relay Board

Grove MOSFET

Demo Videos

Here is a video that shows the set-up and running of the Grove Relay Board

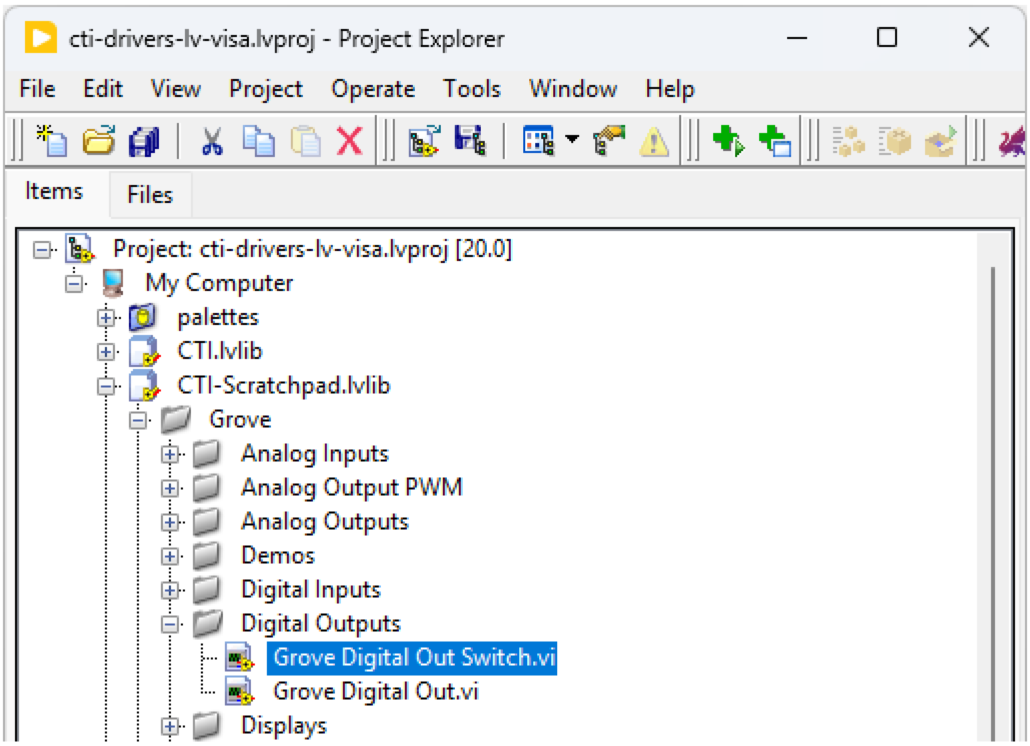
▶ https://www.youtube.com/watch?v=-c-kp2Pila4 *(YouTube video)*

Here is a video that shows the set-up and running of the Grove MOSFet Board

▶ https://www.youtube.com/watch?v=T9XEYUTOUW0 *(YouTube video)*

Example Code

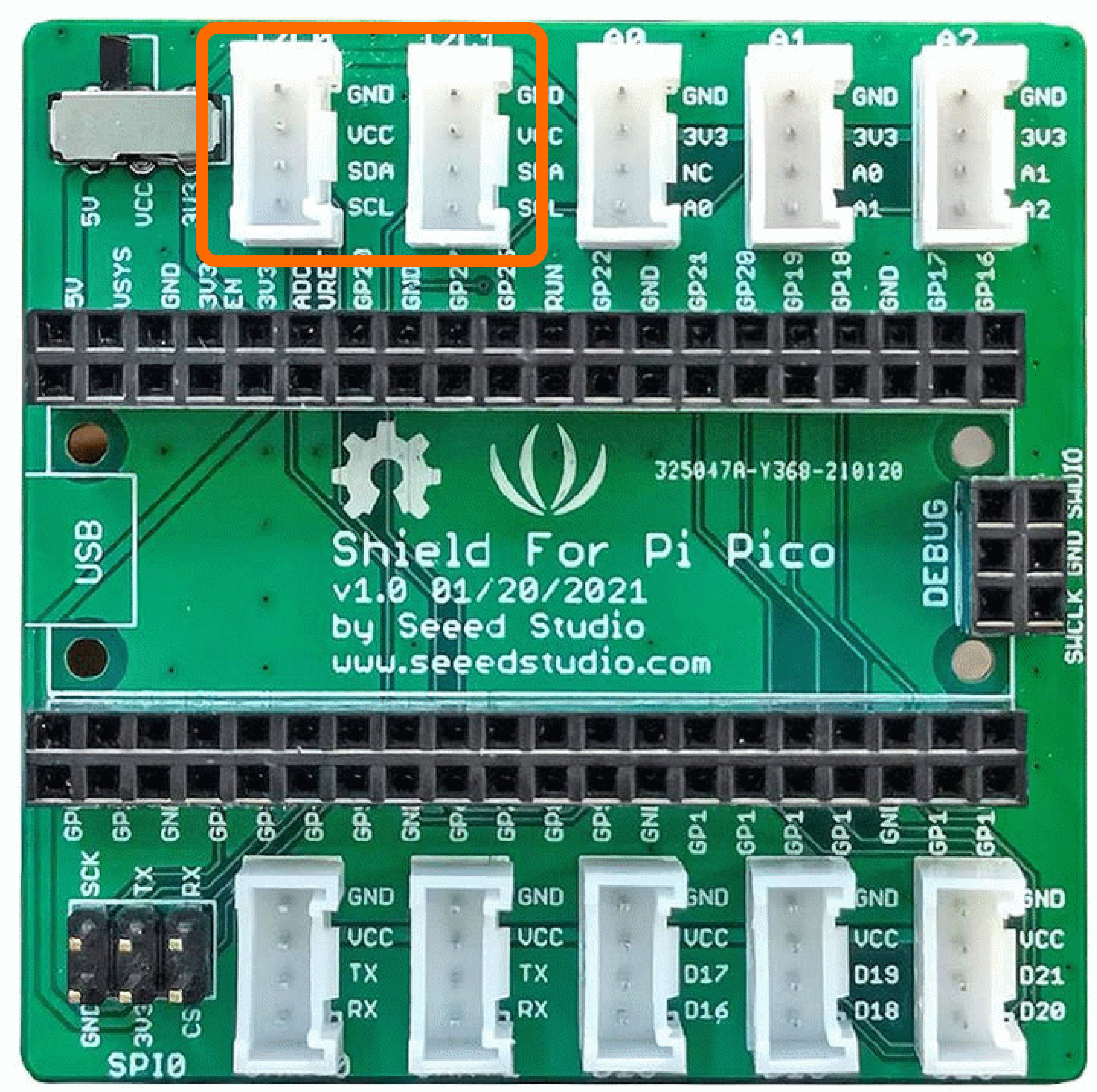
Navigate to >>Scratchpad>>Grove>>Digital Outputs>>**Grove Digital Out Switch.vi**



I2C 4 Channel SPDT Relay Board

Overview

There are also options that communicate using I2C. These will use connectors I2C0 and I2C1 as shown.



This example uses the grove 4 Channel Relay module, there are other similar modules that may work with this code.

The Single Pole Double Throw SPDT relay is quite useful in certain applications because it has one common terminal and 2 contacts which are great for selecting between two options. The Grove - 4-Channel SPDT Relay has four single pole - double throw (SPDT) switches. It only requires low-voltage and low current signals to control those switches. Specifically, you can use 5V DC to control max.250V AC or 110V DC. The I2C address is changeable so that you can use multiple relay modules in the same project.

The Grove - 4-Channel SPDT Relay has four single pole - double throw (SPDT) switches. It only requires low-voltage and low current signals to control those switches. Specifically, you can use 5V DC to control max.250V AC or 110V DC.

SEEED uses an on-board STM32F030F4P6 to control the channels separately. The command from Arduino or other boards is transmit via the I2C interface, the on-board STM32F030F4P6 will parse the command, so that you can control the switch you want.

Hardware Details

Grove I2C 4 Channel SPDT Relay Board

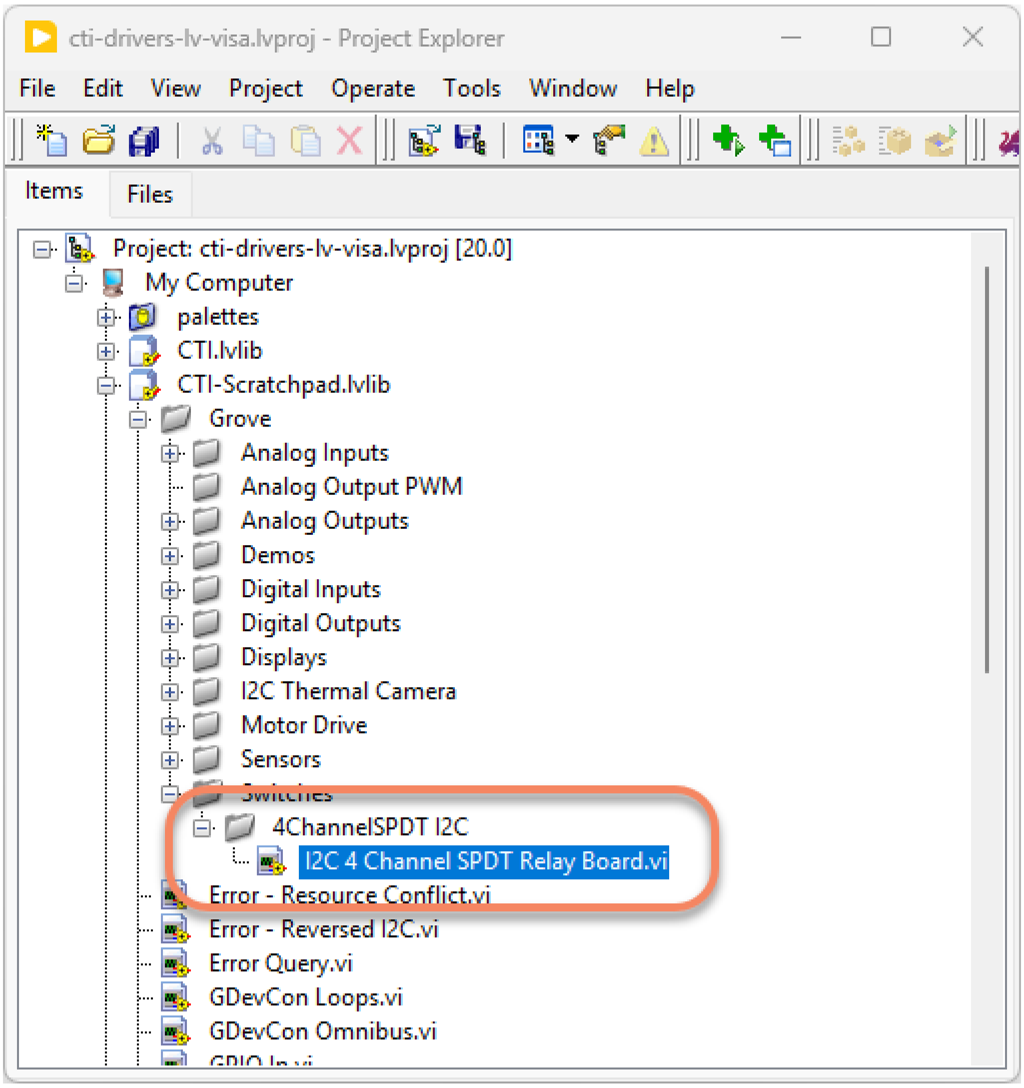
Demo Video

Here is a video that shows the set-up and running of the 4 Channel Relay Board

▶ https://www.youtube.com/watch?v=0Jp3GYXE-z4 *(YouTube video)*

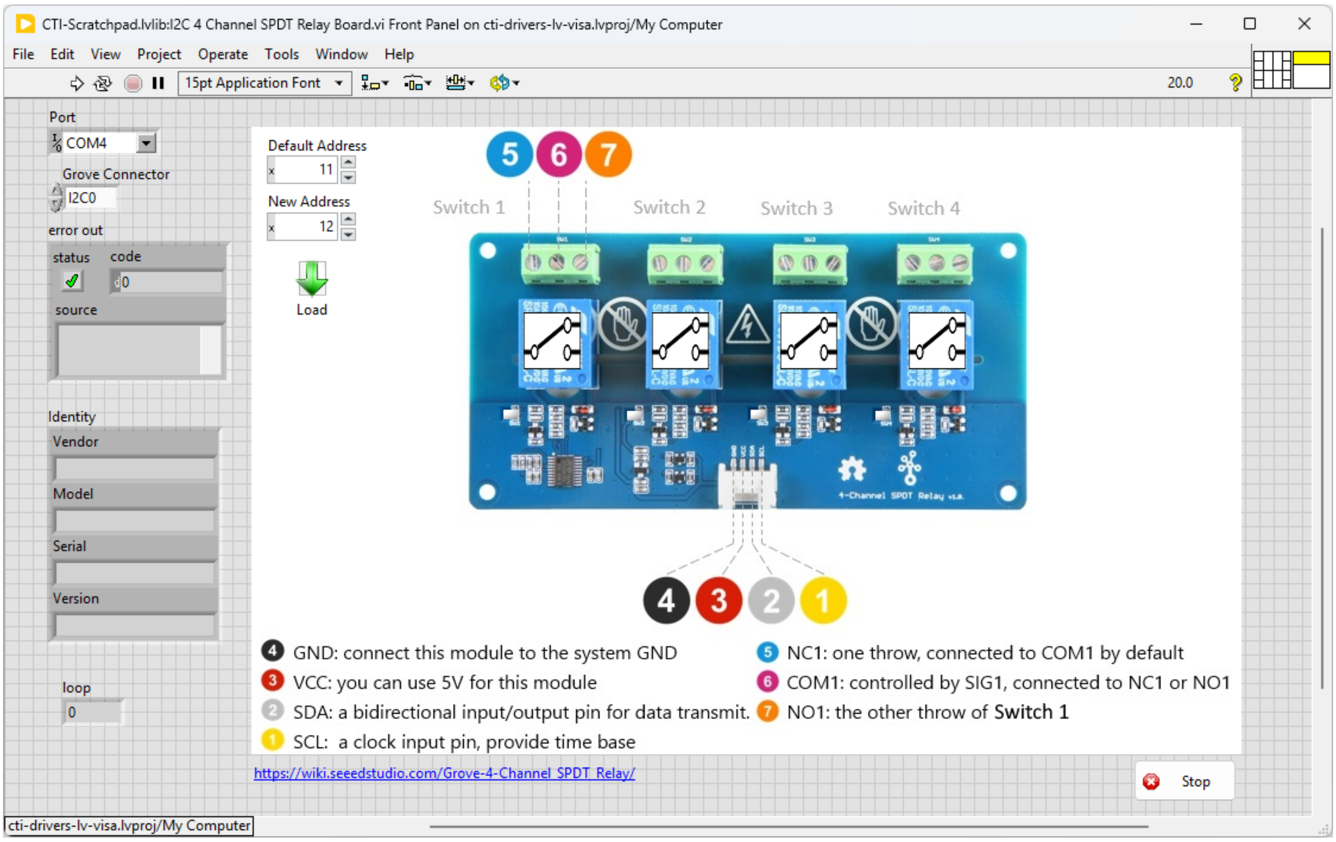
Example Code

Navigate to >>Scratchpad>>Grove>>Switches>>**I2C 4 Channel SPDT Relay Board.vi**

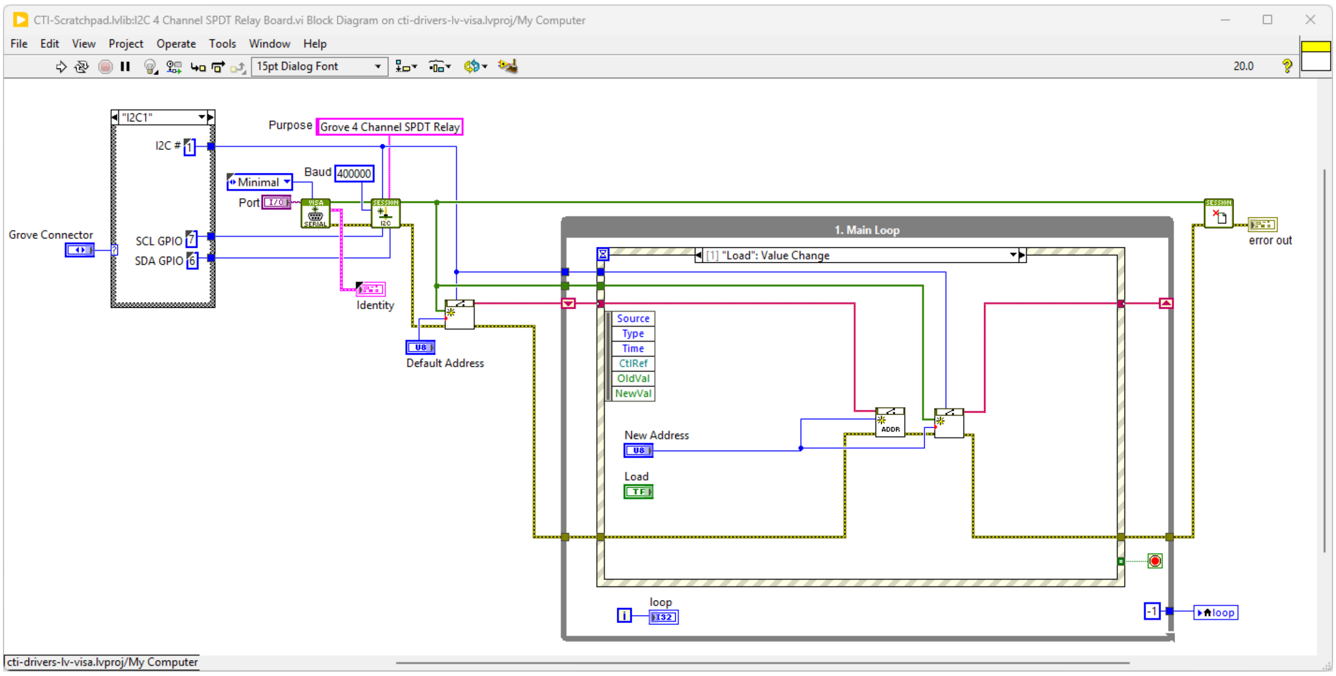


This particular VI uses the 4 Channel Relay Board. Select the port for the connected Pico and the Grove connector that the board is plugged into. Press the run arrow.

You can now click on the switch symbols and the corresponding relay will toggle matching the symbol. If you want to change the I2C address for multiple boards, you can modify [New Address] and press the load button. It's probably a good idea to mark the board with the new address, although it's fairly easy to ping the addresses of attached modules.



On the block diagram you can see that the selected Grove connector dictates the GPIO Pin for the IC2 Port. We set up the IC2 port for the device in **Init.vi**. Next we loop round the event structure and wait for a button on the Front Panel to be pressed. When a switch is detected (event:Switch1,Switch2,Switch3 or Switch4) **Set Switch.vi** recieves the required settings. If the Load button is pressed the new board address is entered and the system re-initialise. Pressing Stop will fire the Stop event and exit the loop.

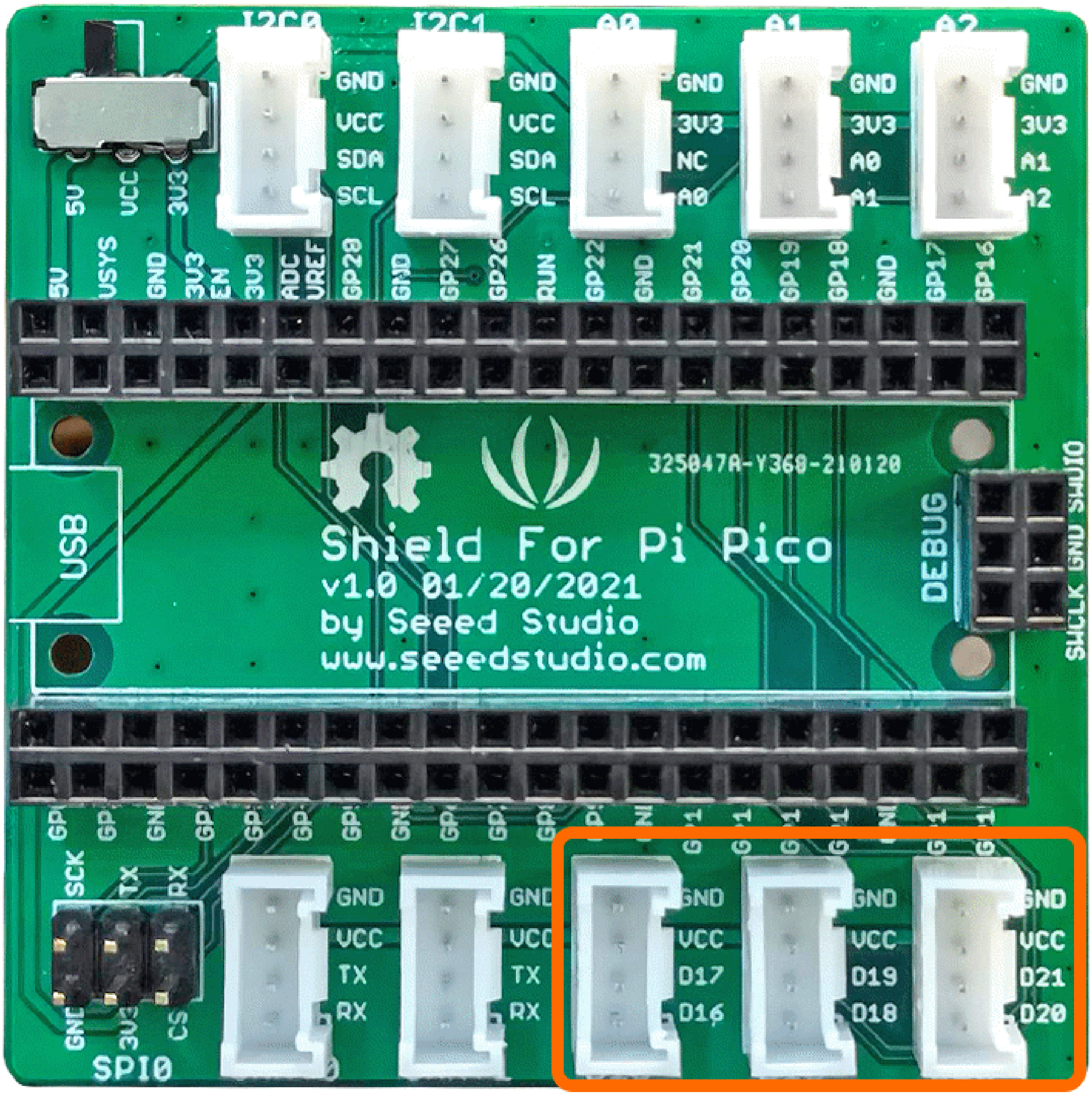


Digital Input Examples

Standard Digital Input

Overview

A digital input can be thought of as the reading from a button, a switch or a digital sensor. These examples use the GPIO Pins [D16,D17],[D18,D19],[D20,D21] that correspond to connectors D16,D18 and D20 on the grove hat.



This example uses the grove button module but the source code is also applicable for other similar Grove Modules.

Hardware Details

Grove Dual Button

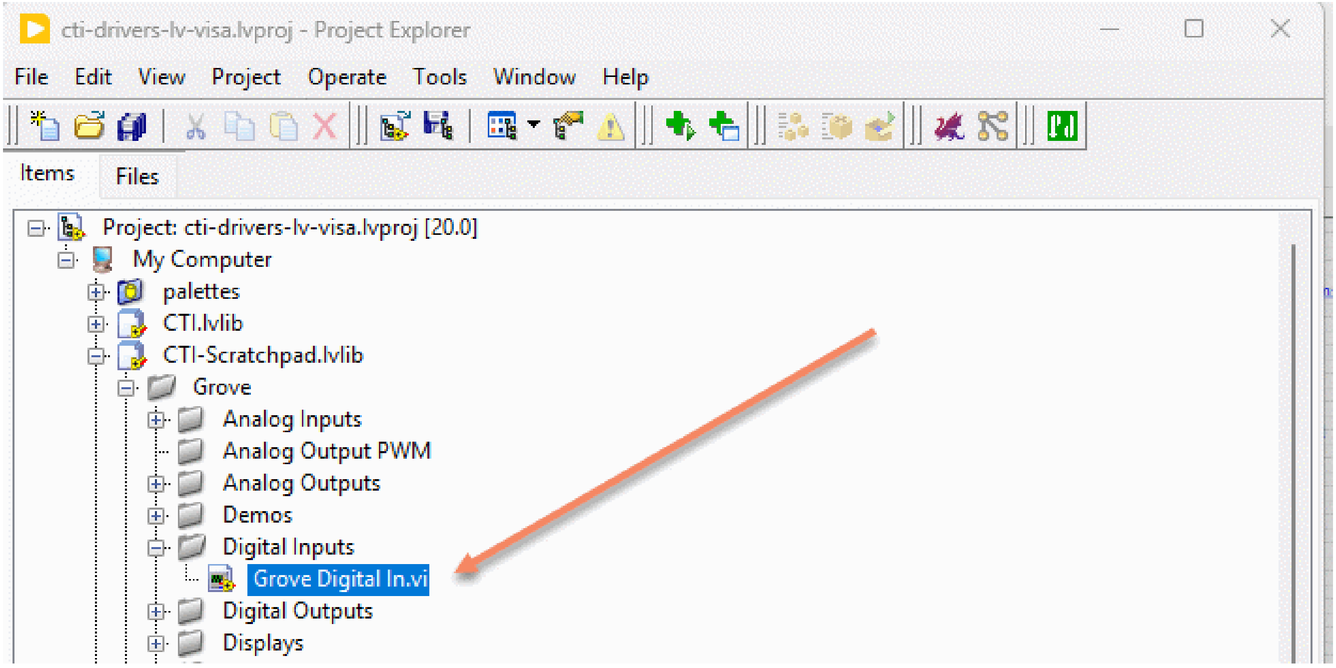
Demo Video

Here is a video that shows the set-up and running of the Dual Button Digital Inputs

▶ https://www.youtube.com/watch?v=ViA9pmmxI2Q *(YouTube video)*

Example Code

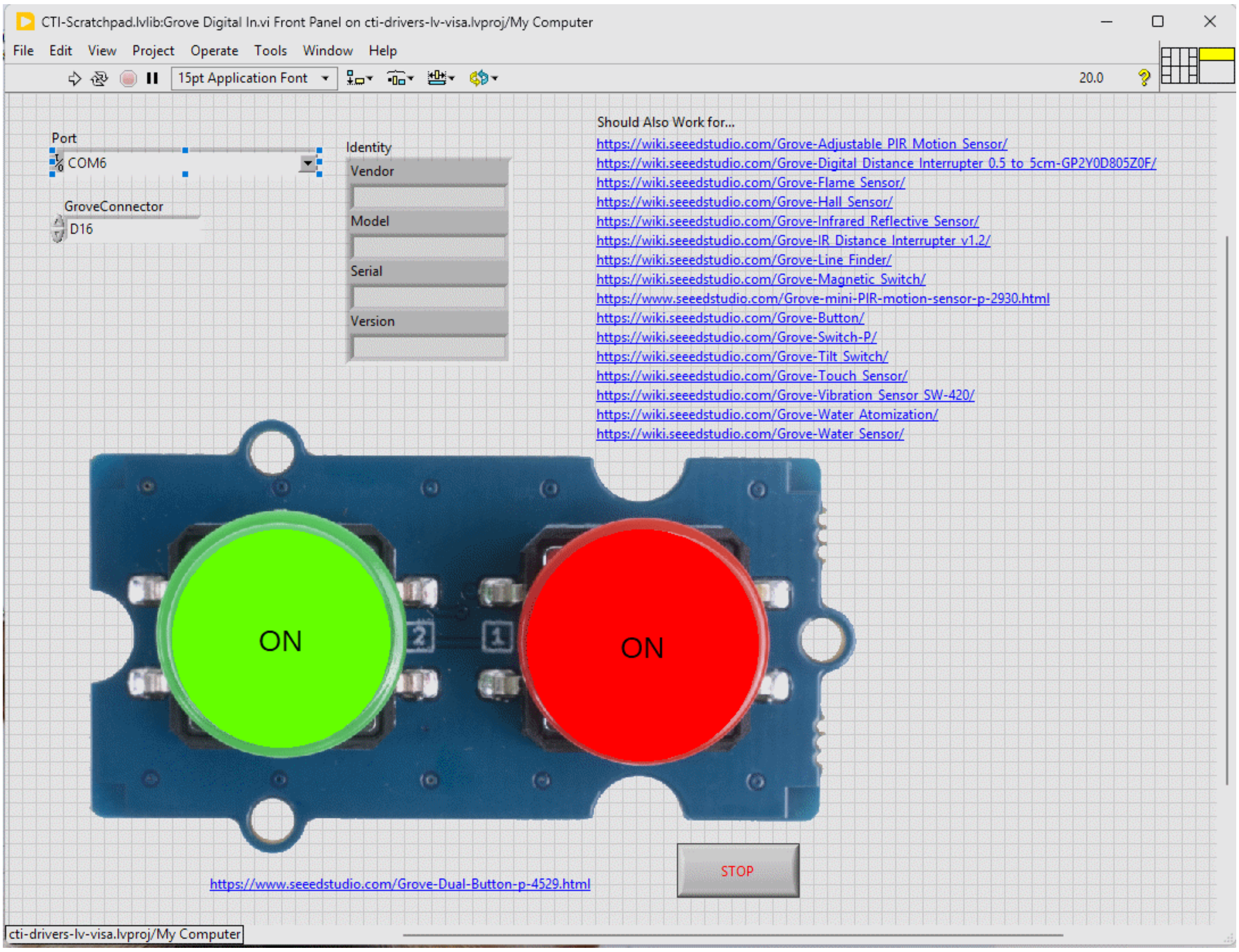
Navigate to **Scratchpad › Grove › Digital Inputs › Grove Digital In.vi**



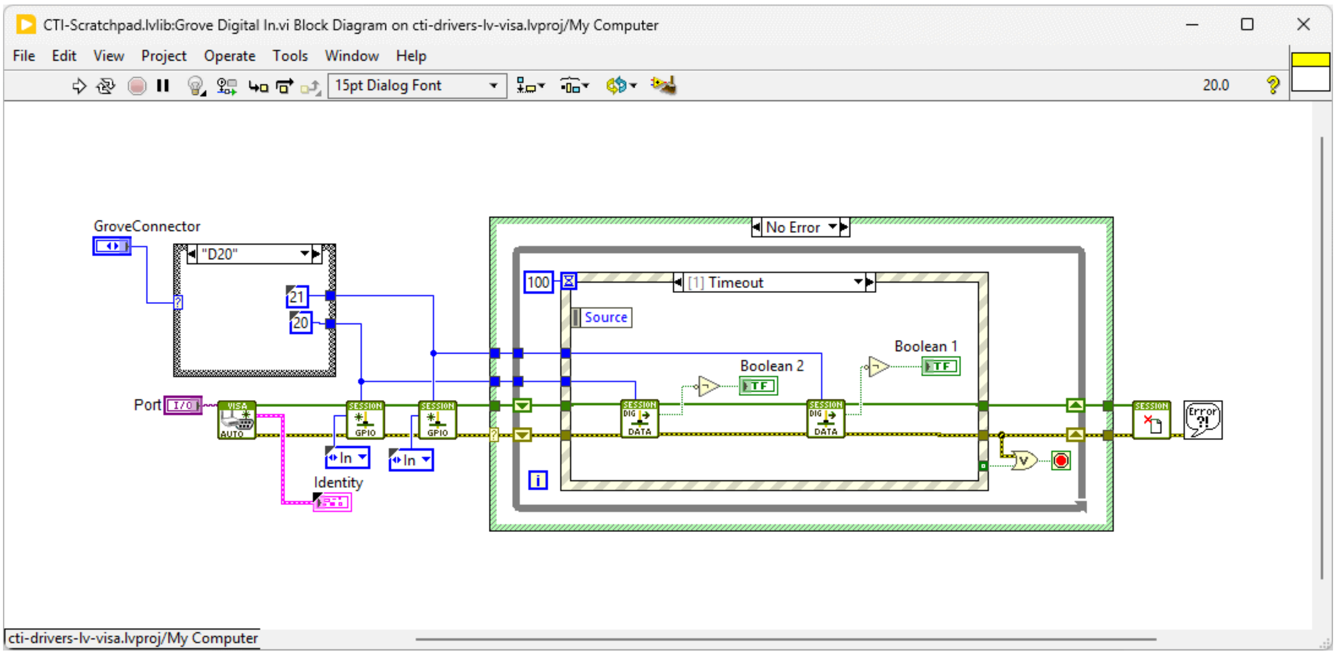
This particular VI uses the dual buttons. Select the port for the connected Pico and the Grove connector that the board is plugged into. Press the run arrow.

You should now see the button indicators light up when the corresponding button is pressed. The Indicators use a transparent fill for the off state and a brighter Red or Green for the on state.

Links to various other similar boards are also on the Front Panel.



On the block diagram you can see that the selected Grove connector dictates the GPIO Pin and we then set the pin to [In]. Next we loop round and event structure and use the timeout to poll the digital inputs. Pressing Stop will fire the Stop event and exit the loop.



Analog Output Examples

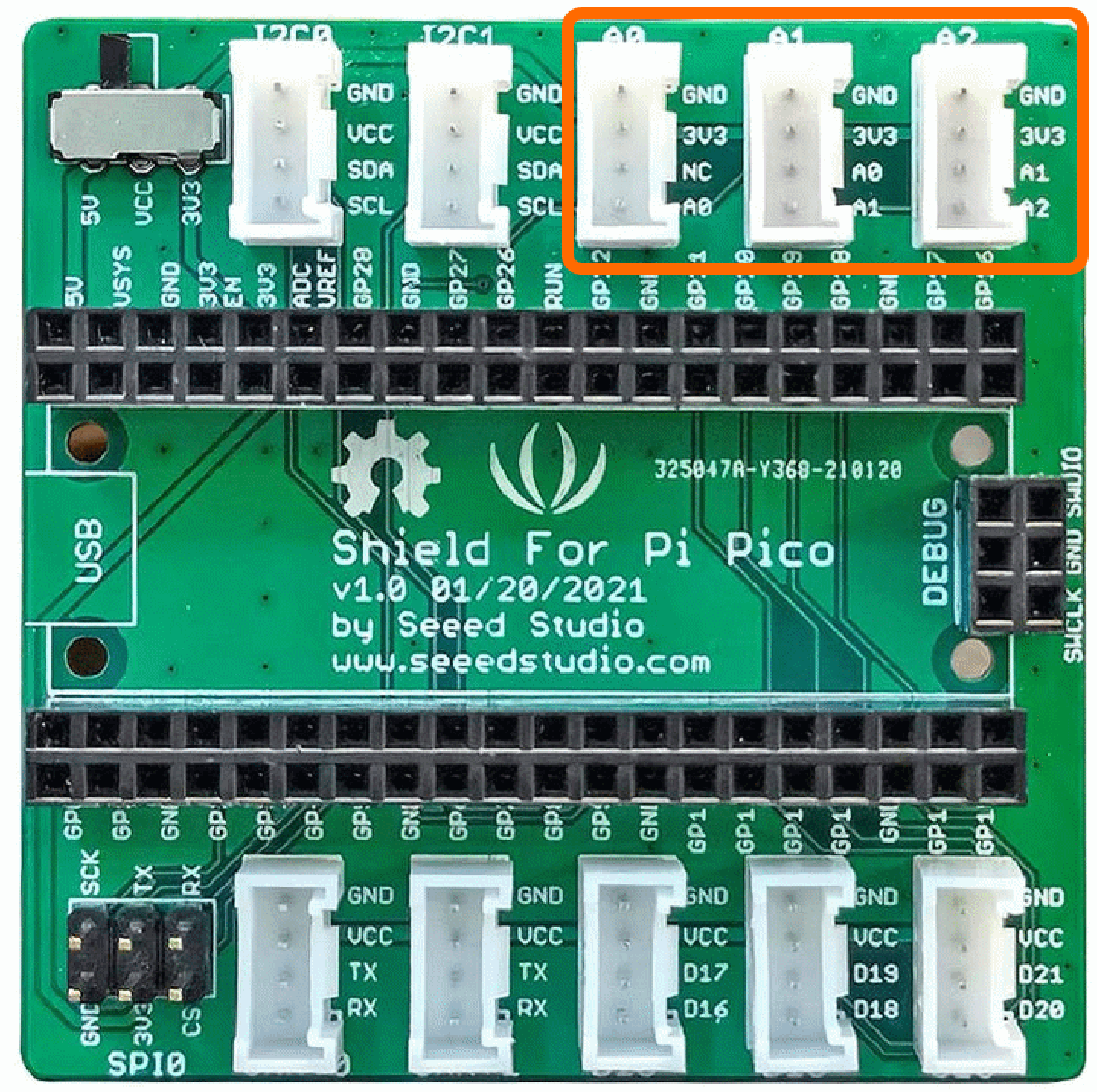
Standard Analog Output PWM

Overview

An analog output can be thought of as a way of generating a signal, the standard way this is done on a Pico is using a PWM square-wave.

The Connections used are [A0,NC],[A1,A2],[A2,A1] and these correspond to GPIO Pins A0=GPIO26, A1=GPIO27, A1=GPIO28

The connectors used are [A0],[A1],[A2] as shown below



This example uses the passive buzzer module but the source code is also applicable for other similar Grove Modules.

Hardware Details

Grove Passive Buzzer

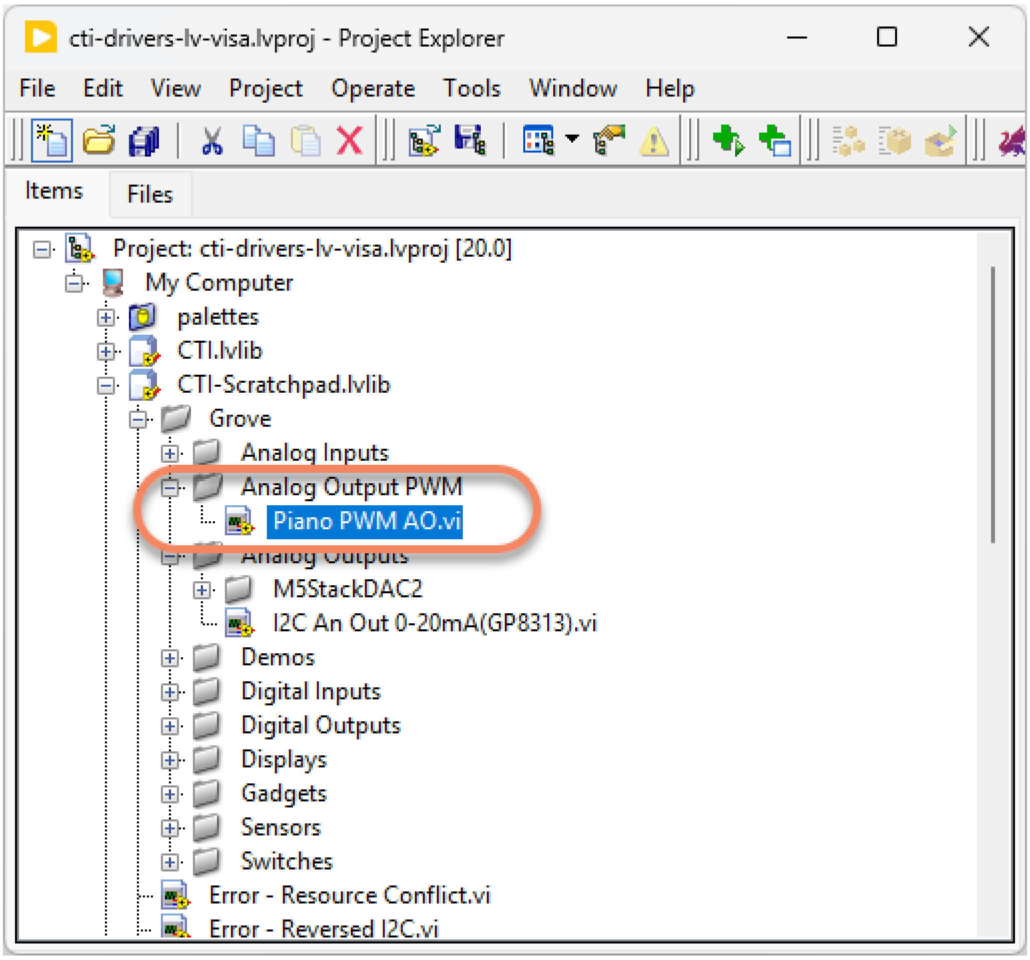
Demo Video

Here is a video that shows the set-up and running of the Passive buzzer using a piano scale to demonstrate Frequency and Duty Cycle

▶ https://www.youtube.com/watch?v=QSPMtF8XWTE *(YouTube video)*

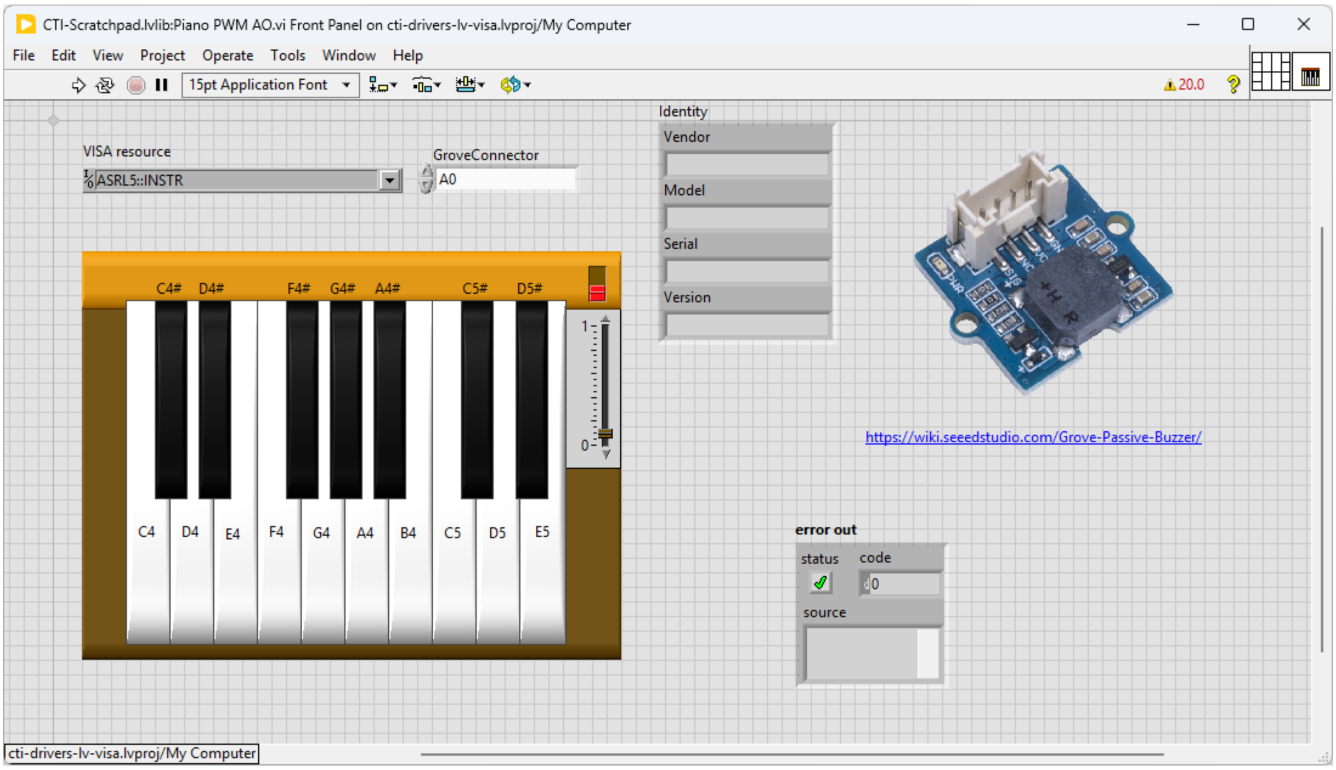
Example Code

Navigate to >>Scratchpad>>Grove>>Analog Output PWM>>**Piano PWM AO.vi**

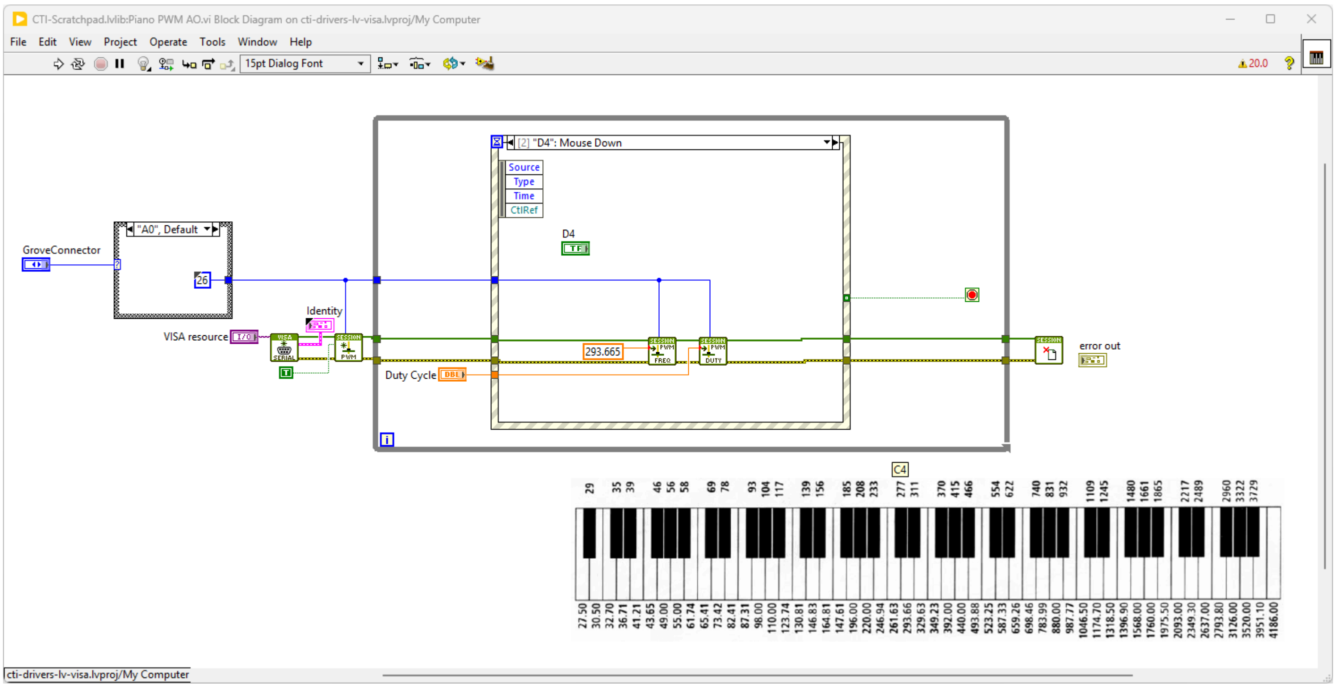


This particular VI uses the passive buzzer to make the sound. Select the port for the connected Pico and the Grove connector that the board is plugged into. Press the run arrow.

Links to various other similar boards are also on the Front Panel.

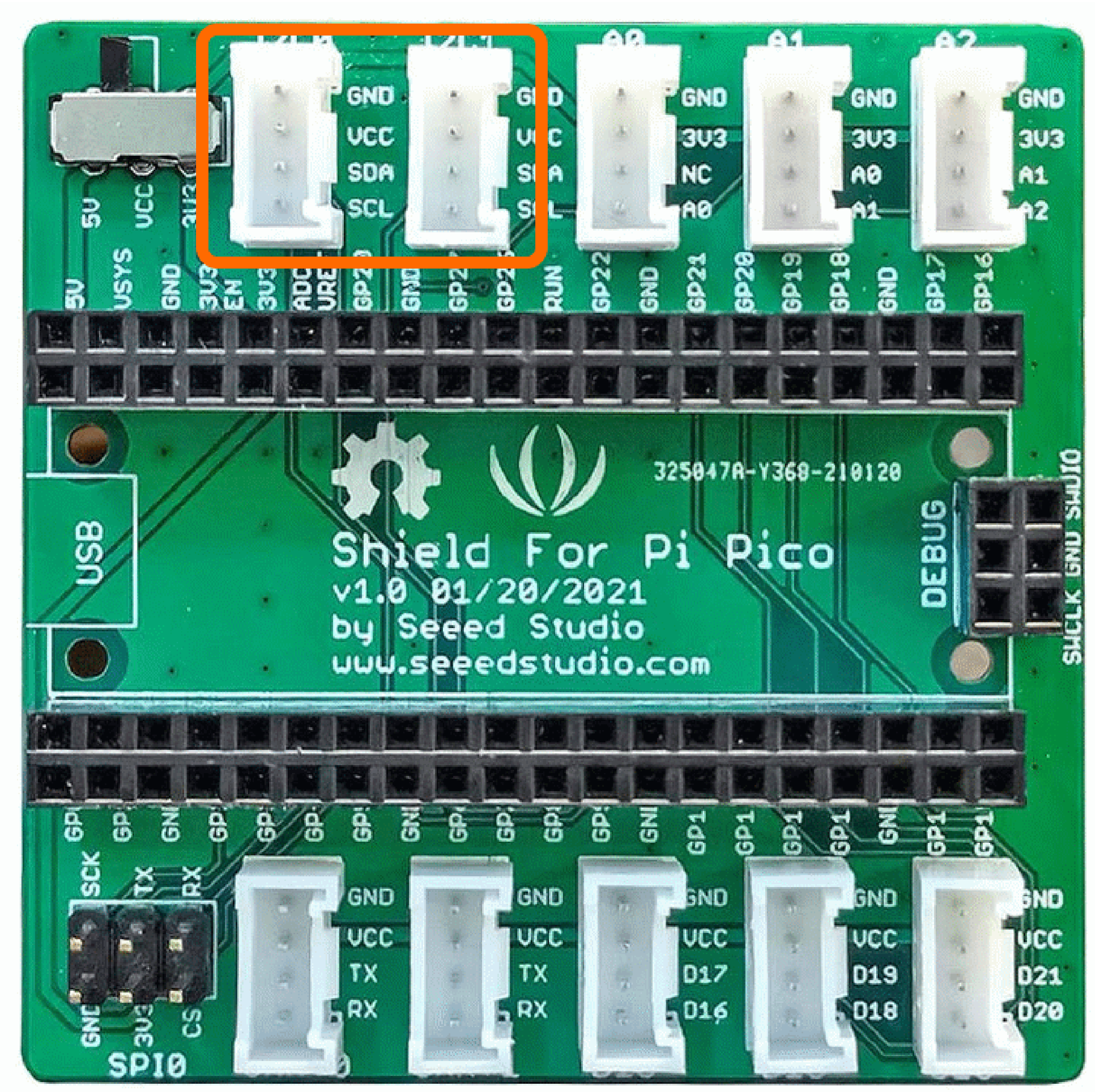


On the block diagram you can see that the selected Grove connector dictates the GPIO Pin and we then wire it to the **Configure PWM.vi** . Next we loop round the event structure waiting for button press events. If a piano key is pressed the corresponding frequency will be entered into the **PWM Write Frequency.vi**. If the duty cycle is high enough a sound of the right key should be heard. Pressing Stop will fire the Stop event and exit the loop.



M5Stack DAC2 I2C Analog Outputs

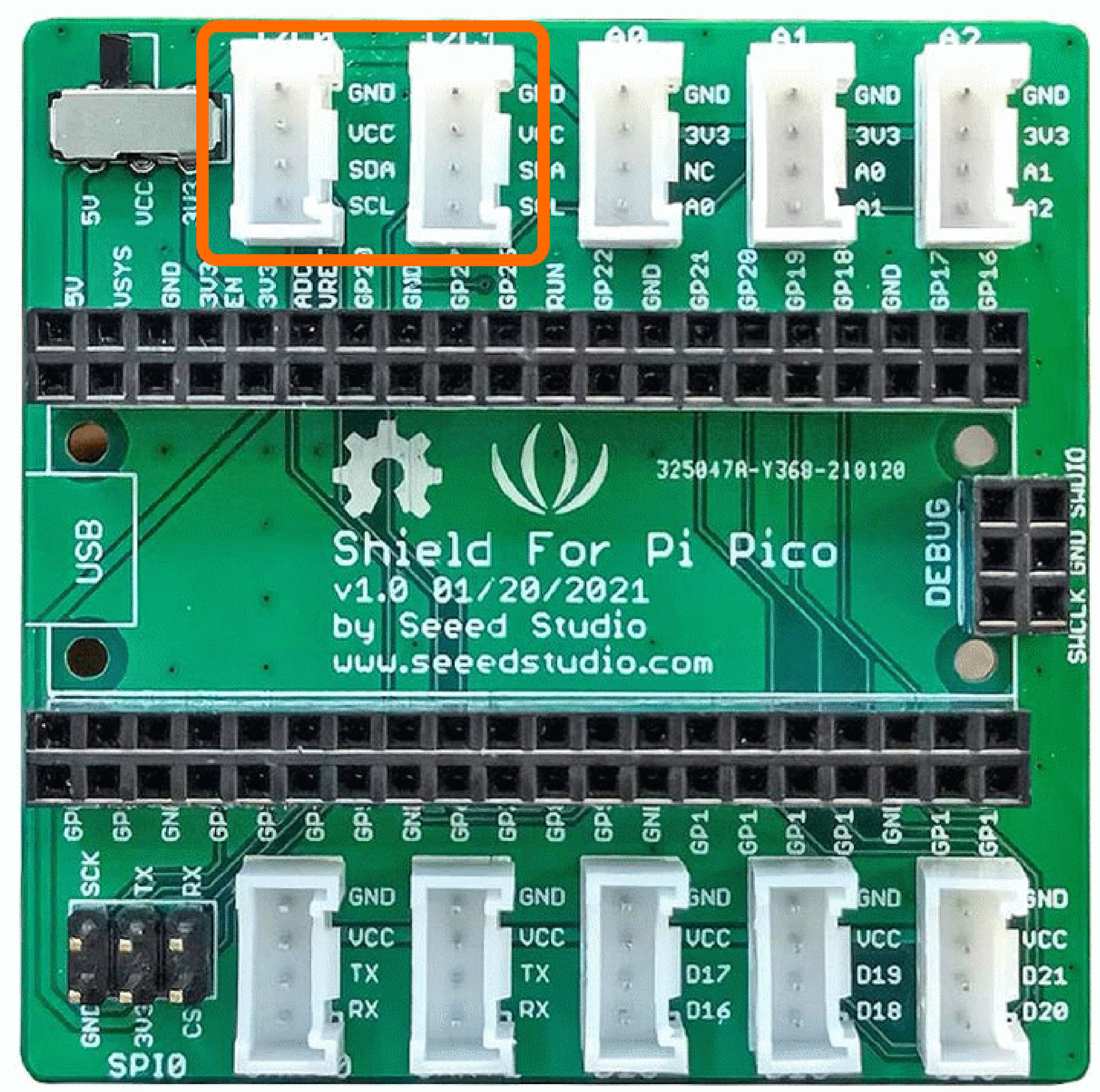
There are also options that communicate using I2C. These will use connectors I2C0 and I2C1 as shown. Commonly these will offer improved resolution e.g. 16 bits so 65536 levels. They may also offer amplifiers, multiplexers or other enhanced functionality.



Overview

This example uses the M5Stack DAC 2 Module, there are other similar modules that may work with this code as it uses the GP8313 integrated circuit. These are Grove compatible modules.

The GPIO Pins used are [I2C0] or [I2C1]



Hardware Details

[M5Stack DAC2 Module]

[Schematic]

[DFRobot boards may also work]

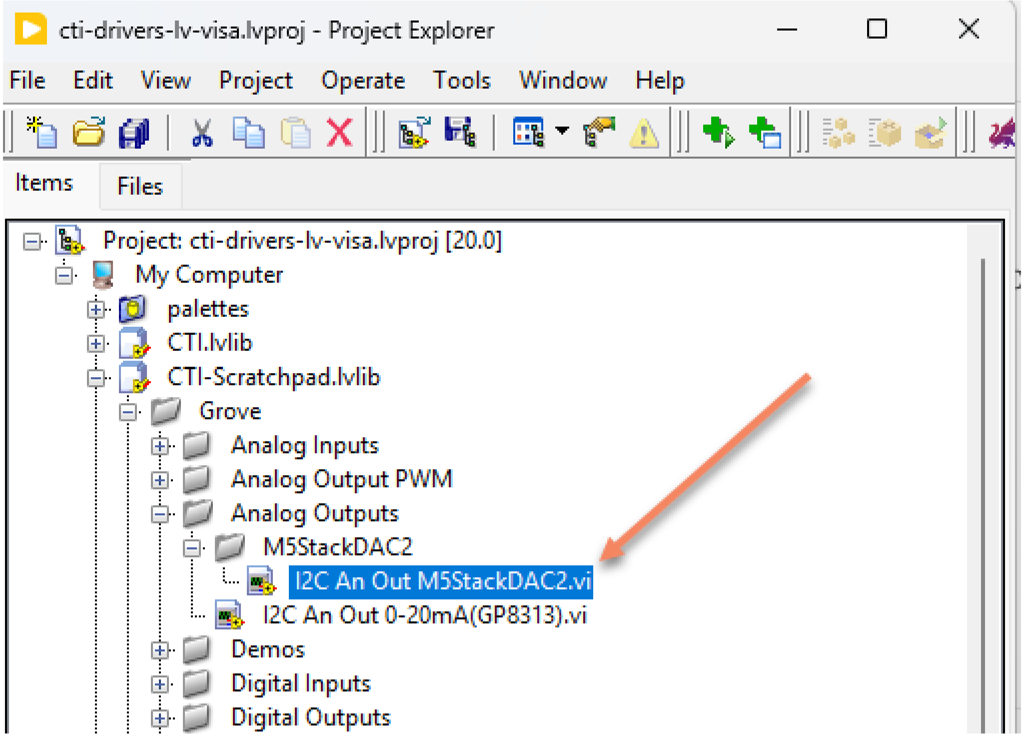
Demo Video

Here is a video that shows the set-up and running of the M5Stack DAC2

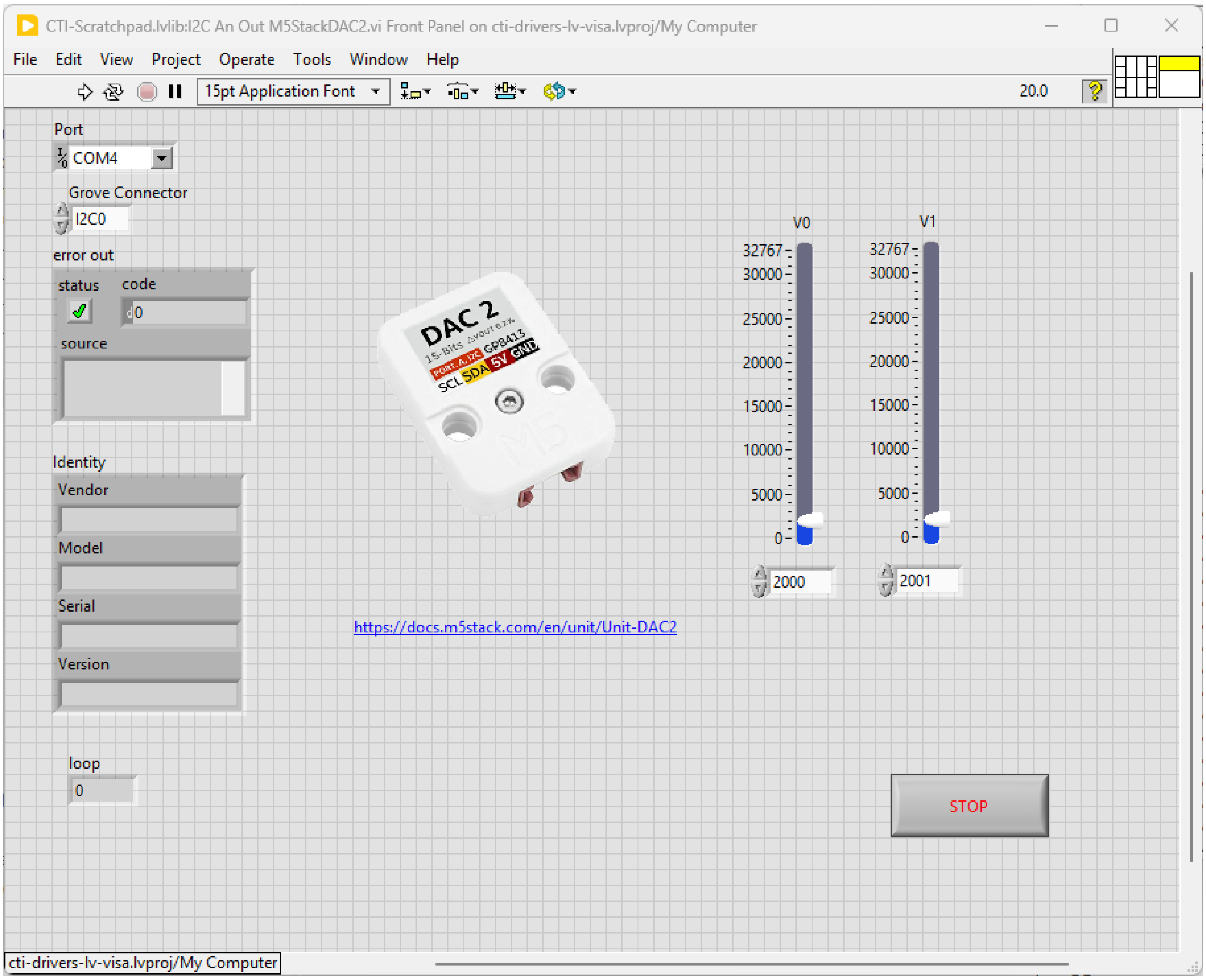
▶ https://www.youtube.com/watch?v=KhWh9I7JJ1Y *(YouTube video)*

Example Code

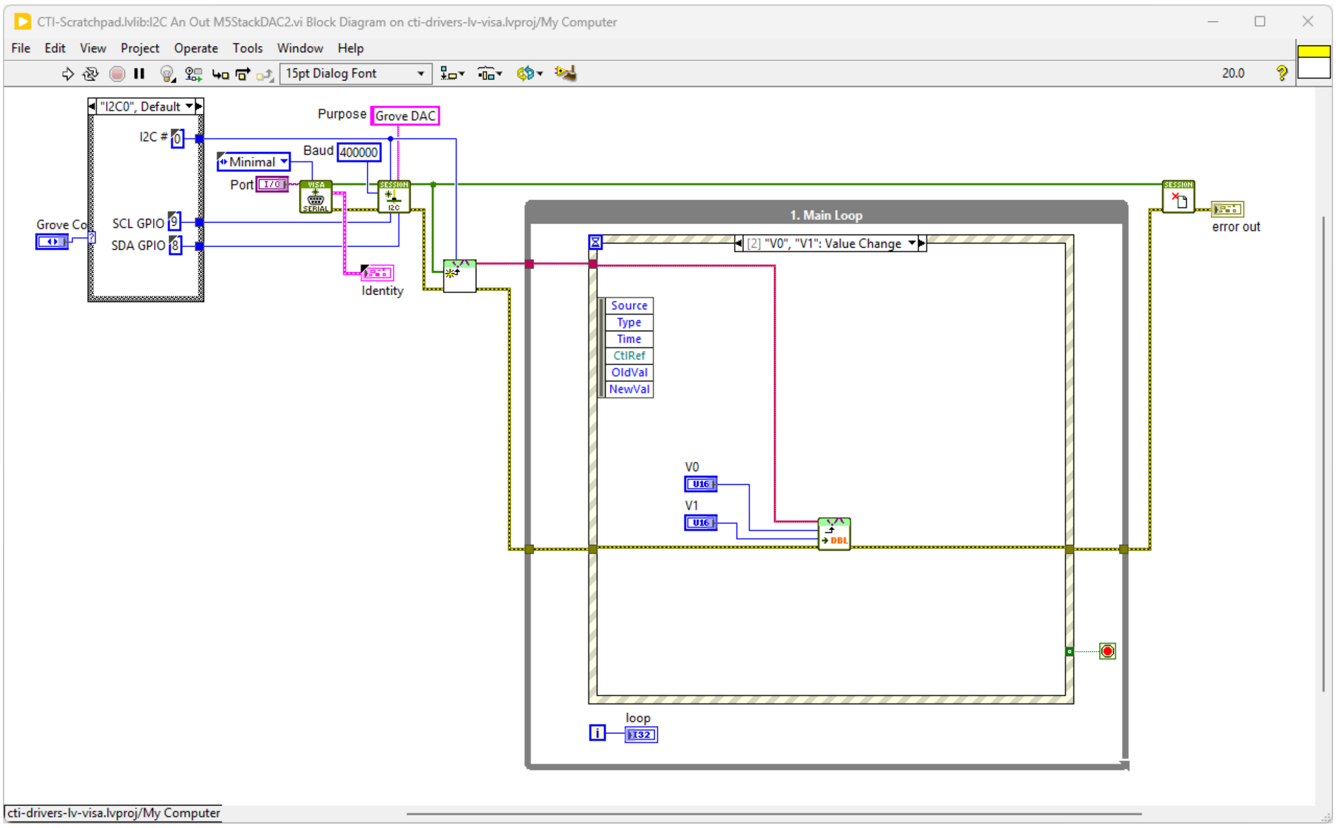
Navigate to >>Scratchpad>>Grove>>Analog Outputs>>**I2C An Out M5StackDAC2.vi**



This particular VI uses the M5Stack DAC2 and this gives us 2 channels of 16 bit Analog Output. Select the port for the connected Pico and the Grove connector that the board is plugged into. Press the run arrow.



On the block diagram you can see that the selected Grove connector dictates the GPIO Pin for the IC2 Port. We set up the IC2 port for the device in **Init.vi**. Next we loop round the event structure and wait for a Sliders V0 or V1 to be changed. When change has been detected the new values are input into the **Write DAC.vi**. Pressing Stop will fire the Stop event and exit the loop.


CTI-Scratchpad.lvlib:I2C An Out M5StackDAC2.vi Block Diagram on cti-drivers-lv-visa.lvproj/My Computer
File Edit View Project Operate Tools Window Help
15pt Dialog Font
"I2C0", Default
I2C #
Purpose
Grove DAC
Baud
400000
Minimal
Port
Grove Co
SCL GPIO
SDA GPIO
Identity
1. Main Loop
[2] "V0", "V1": Value Change
Source
Type
Time
CtlRef
OldVal
NewVal
V0
V1
loop
error out
cti-drivers-lv-visa.lvproj/My Computer

Analog Input Examples

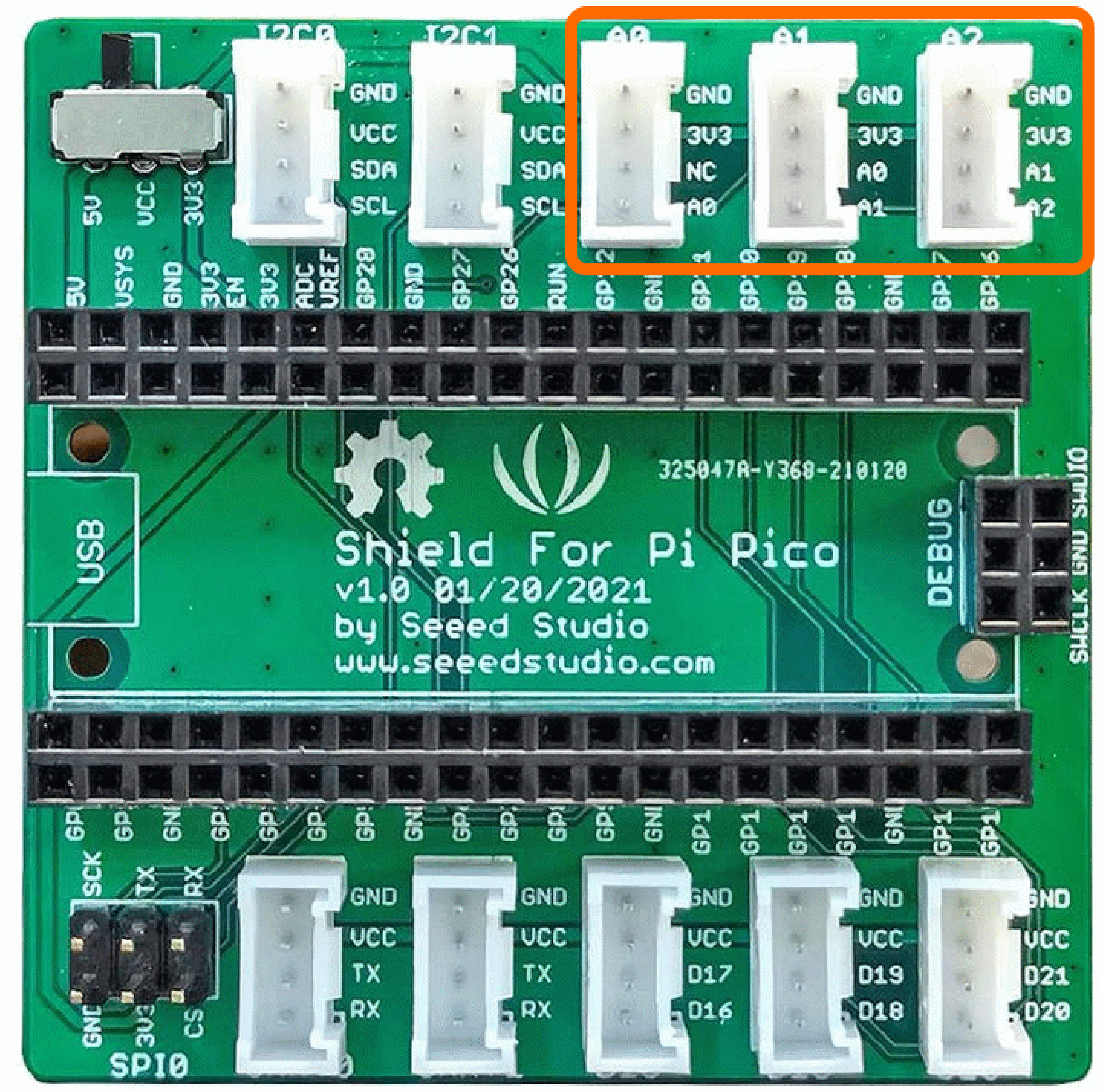
Grove Standard Analog Inputs

Overview

An analog input can be thought of as a measurement or sensor reading. These will natively have a resolution of 12 bits, which can be thought of as a sensitivity across the range of the input. 12 bits= 4096 levels across the range.

The Connections used are [A0,NC],[A1,A2],[A2,A1] and these correspond to GPIO Pins A0=GPIO26, A1=GPIO27, A1=GPIO28

The connectors used are [A0],[A1],[A2] as shown below



Hardware Details

Rotary Angle Sensor

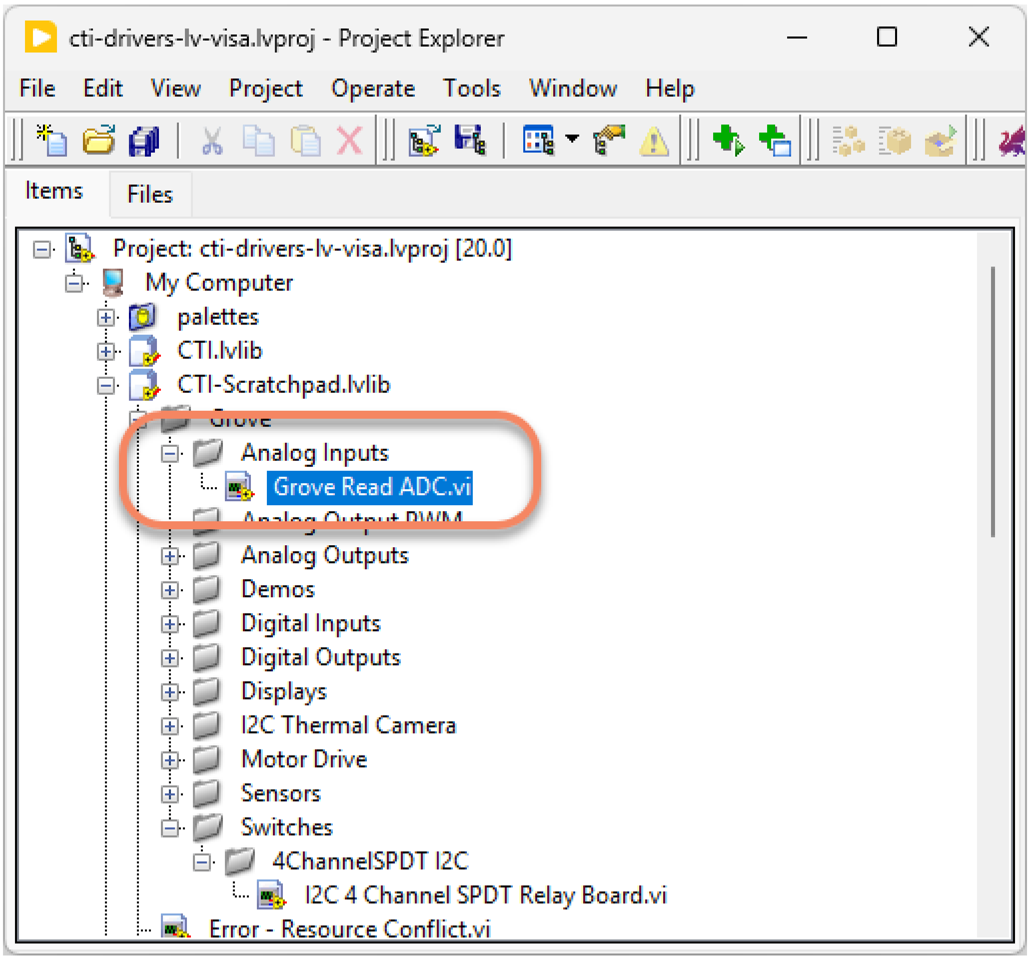
Demo Video

Here is a video that shows the set-up and running of the Rotary Angle Sensor

▶ https://www.youtube.com/watch?v=0-suR8HejsY *(YouTube video)*

Example Code

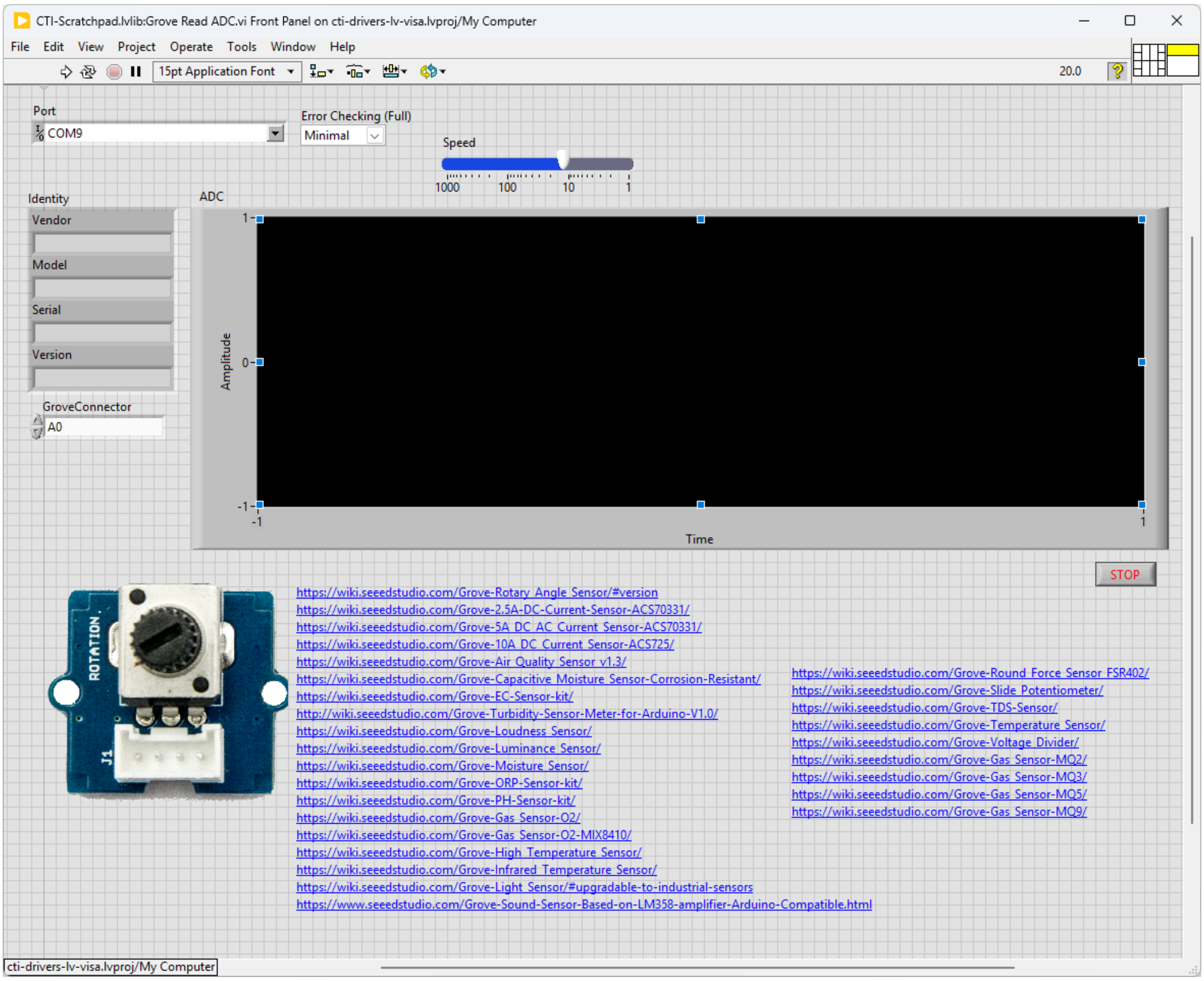
Navigate to >>Scratchpad>>Grove>>Analog Inputs>>Grove Read ADC.vi



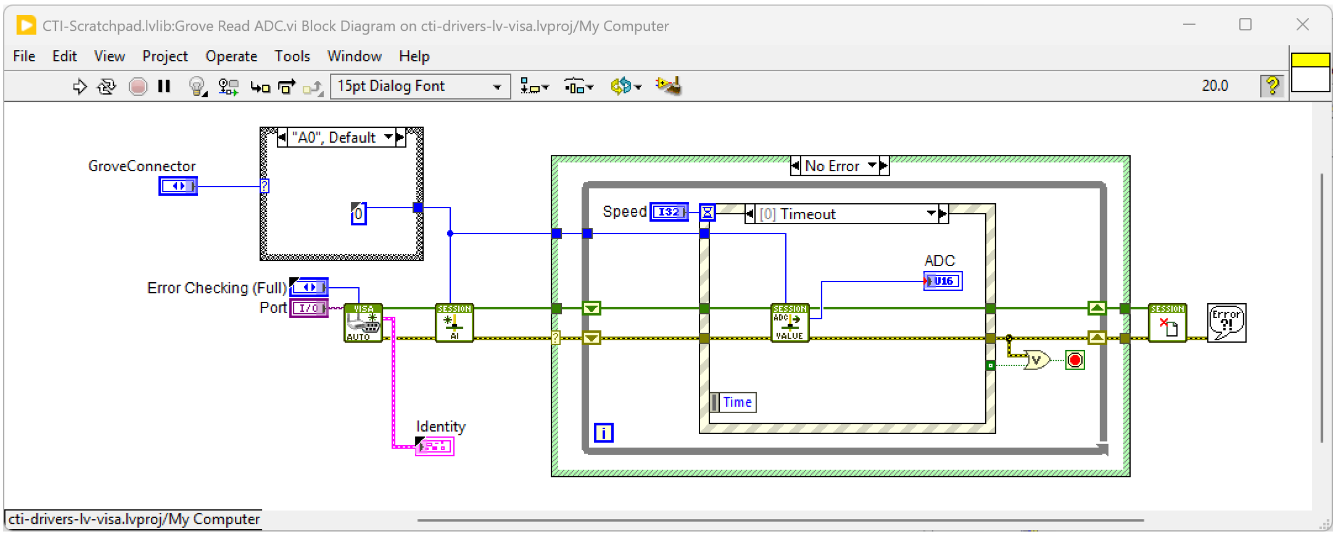
This particular VI uses a rotary angle sensor (potentiometer). Select the port for the connected Pico and the Grove connector that the board is plugged into. Press the run arrow.

You should be able to see the graph change corresponding to how much you change the rotary sensor.

Links to various other similar boards are also on the Front Panel.

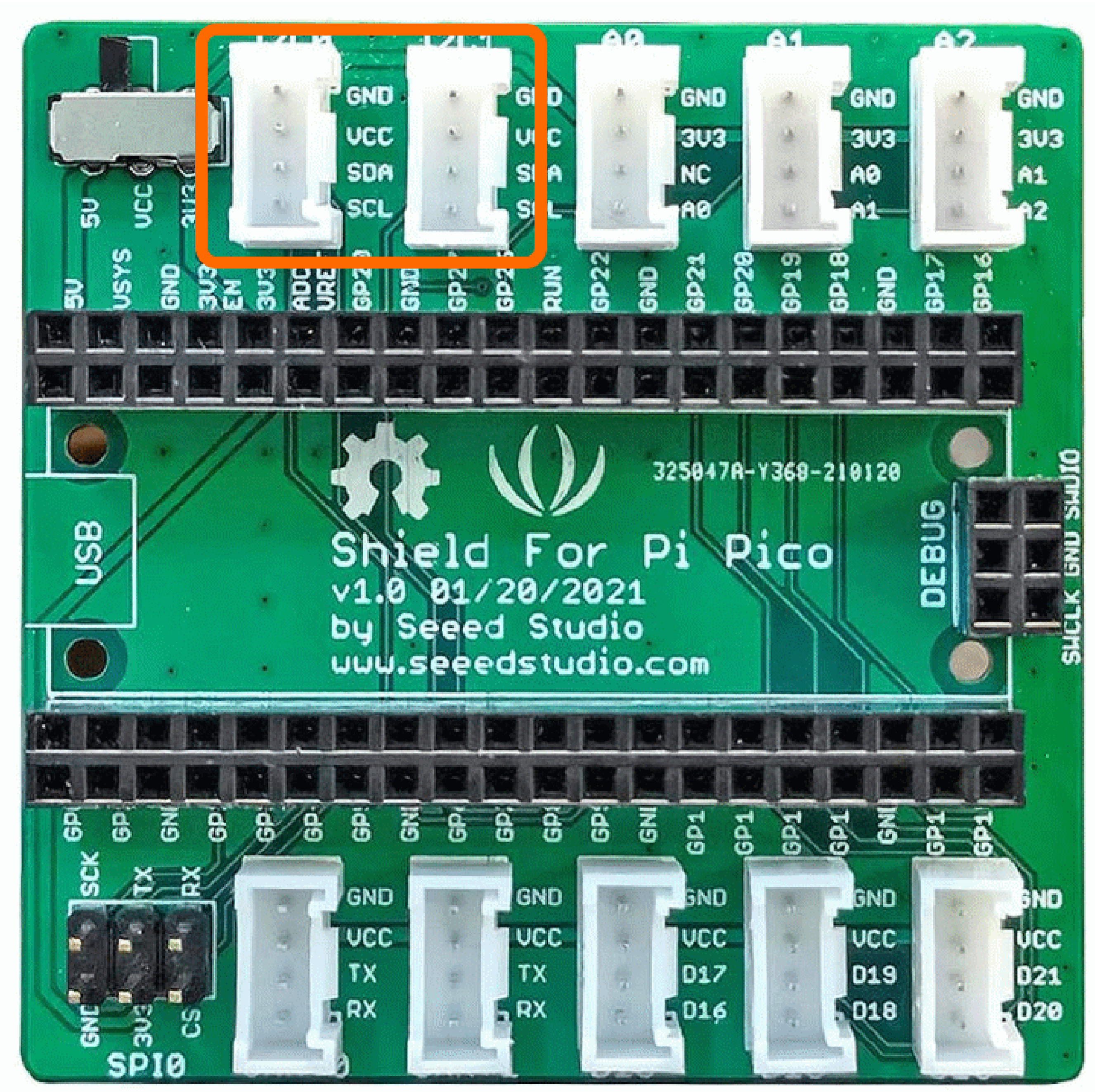


On the block diagram you can see that the selected Grove connector is input into **Configure AI.vi**. Next we loop round the event structure and use the timeout to poll the selected analog input using **AI Read Value.vi**. Pressing Stop will fire the Stop event and exit the loop.



Grove I2C Analog Inputs

There are also options that communicate using I2C. These will use connectors I2C0 and I2C1 as shown. Commonly these will offer improved resolution e.g. 16 bits or 24 bits, so 65536 levels or 16,777,215 level respectively. They may also offer amplifiers, multiplexers or other enhanced functionality.



Grove I2C 4 Channel 16 Bit Analog Input

Overview

This module provides 4 channels with 16 bit resolution and programmable gain.

Hardware Details

Grove I2C 4 Channel 16 Bit Analog Input

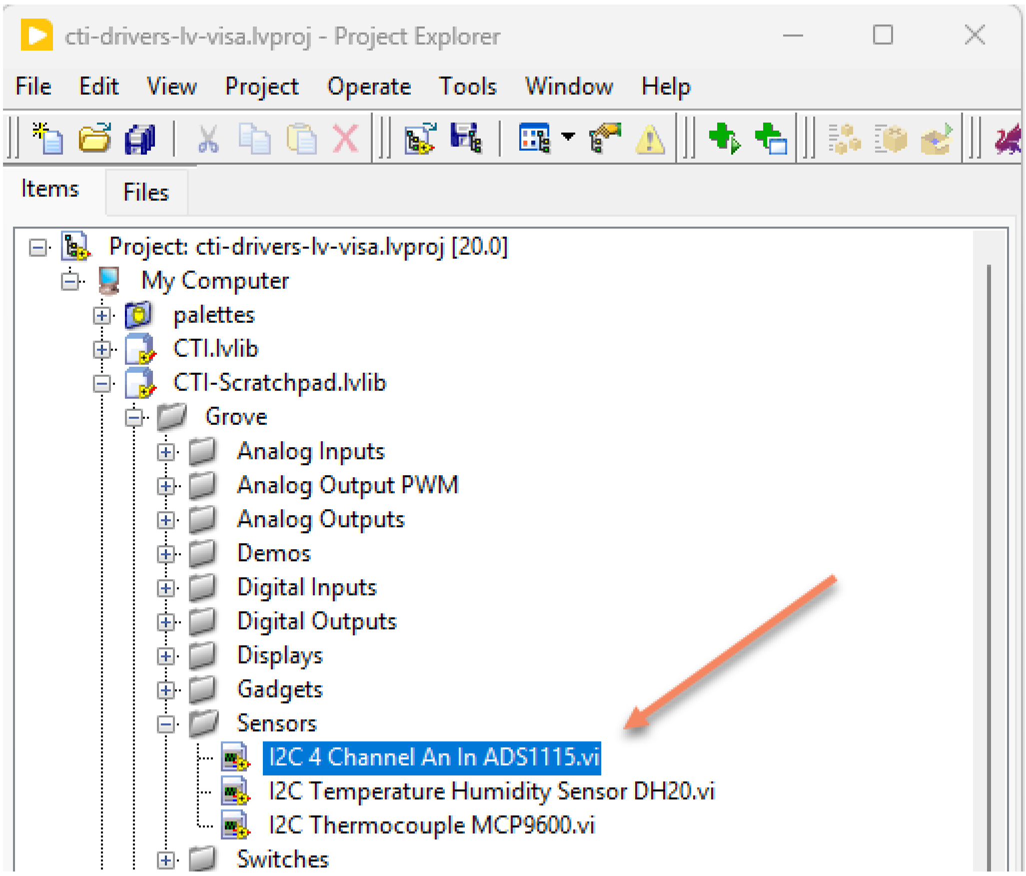
Demo Video

Here is a video that shows the set-up and running of the Grove I2C 4 Channel 16 Bit Analog Input module

▶ https://www.youtube.com/watch?v=_xpF_2K6F94 *(YouTube video)*

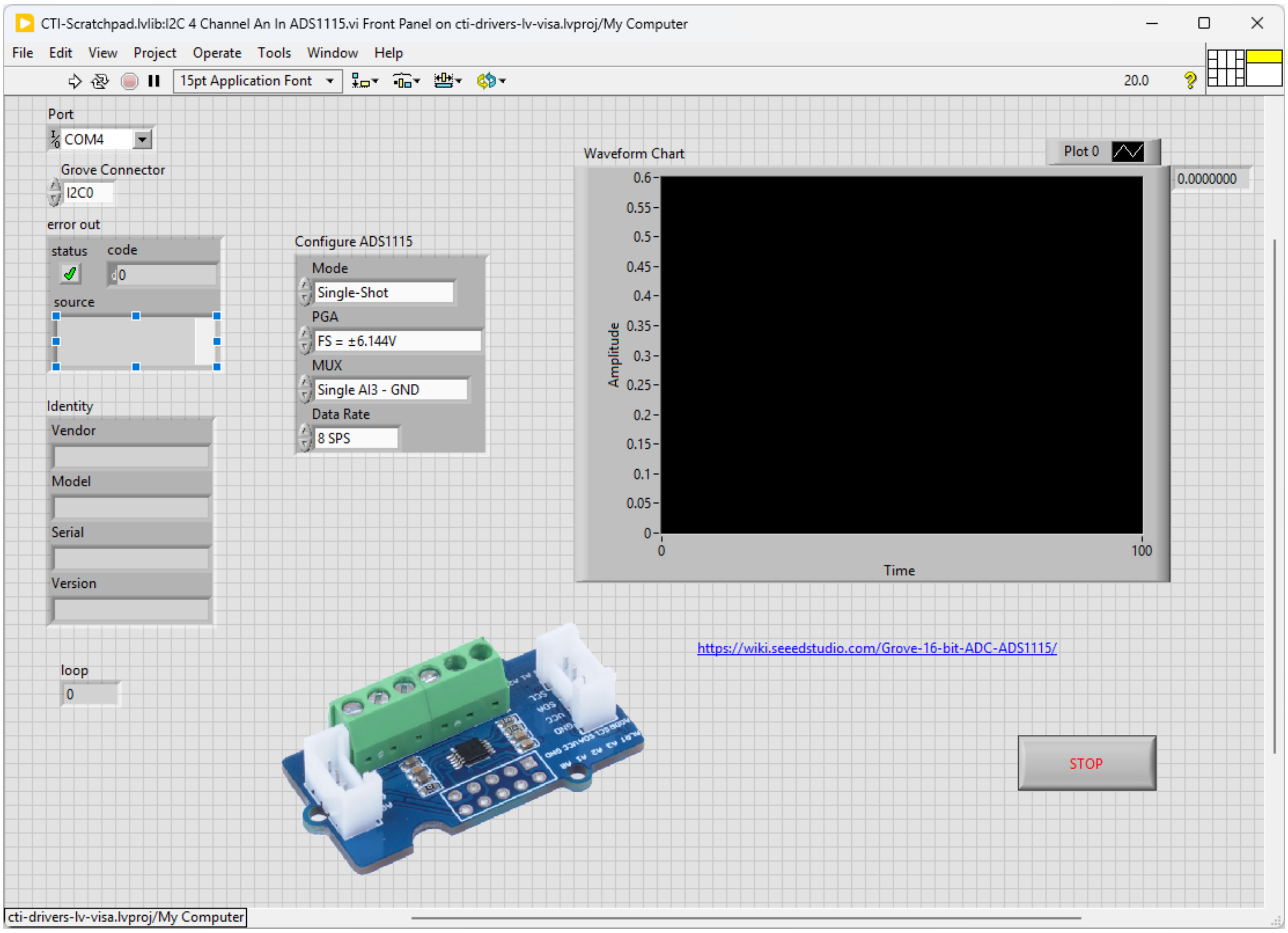
Example Code

Navigate to >>Scratchpad>>Grove>>Sensors>>**I2C 4 Channel An In ADS1115.vi**

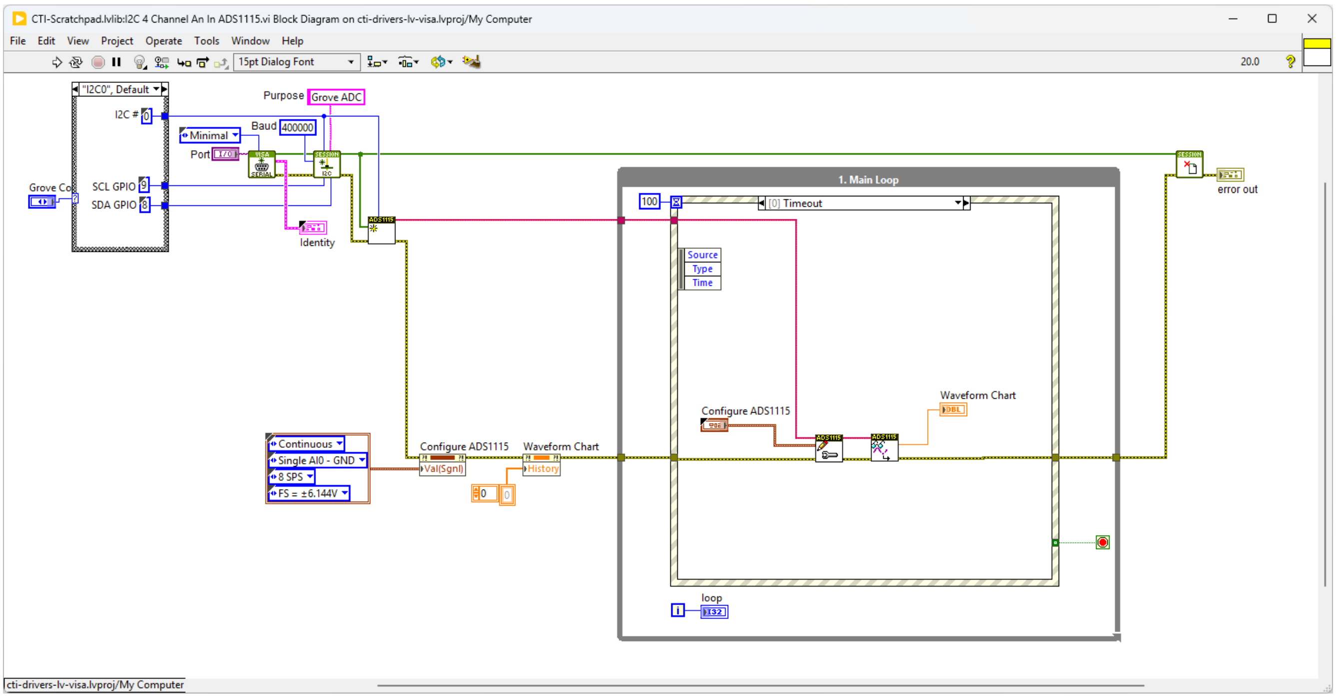


In this example we will need to have a voltage source to show a change in signal. Here I've just wired a potentiometer to the 5V and Ground on the board.

This VI uses ADS1115 4 Channels Analog Input Module. Select the port for the connected Pico and the Grove connector that the board is plugged into. Press the run arrow.



On the block diagram you can see that the selected Grove connector is input into **Configure AI.vi**. Next we loop round the event structure and use the timeout to poll the selected analog input using **AI Read Value.vi**. Pressing Stop will fire the Stop event and exit the loop.

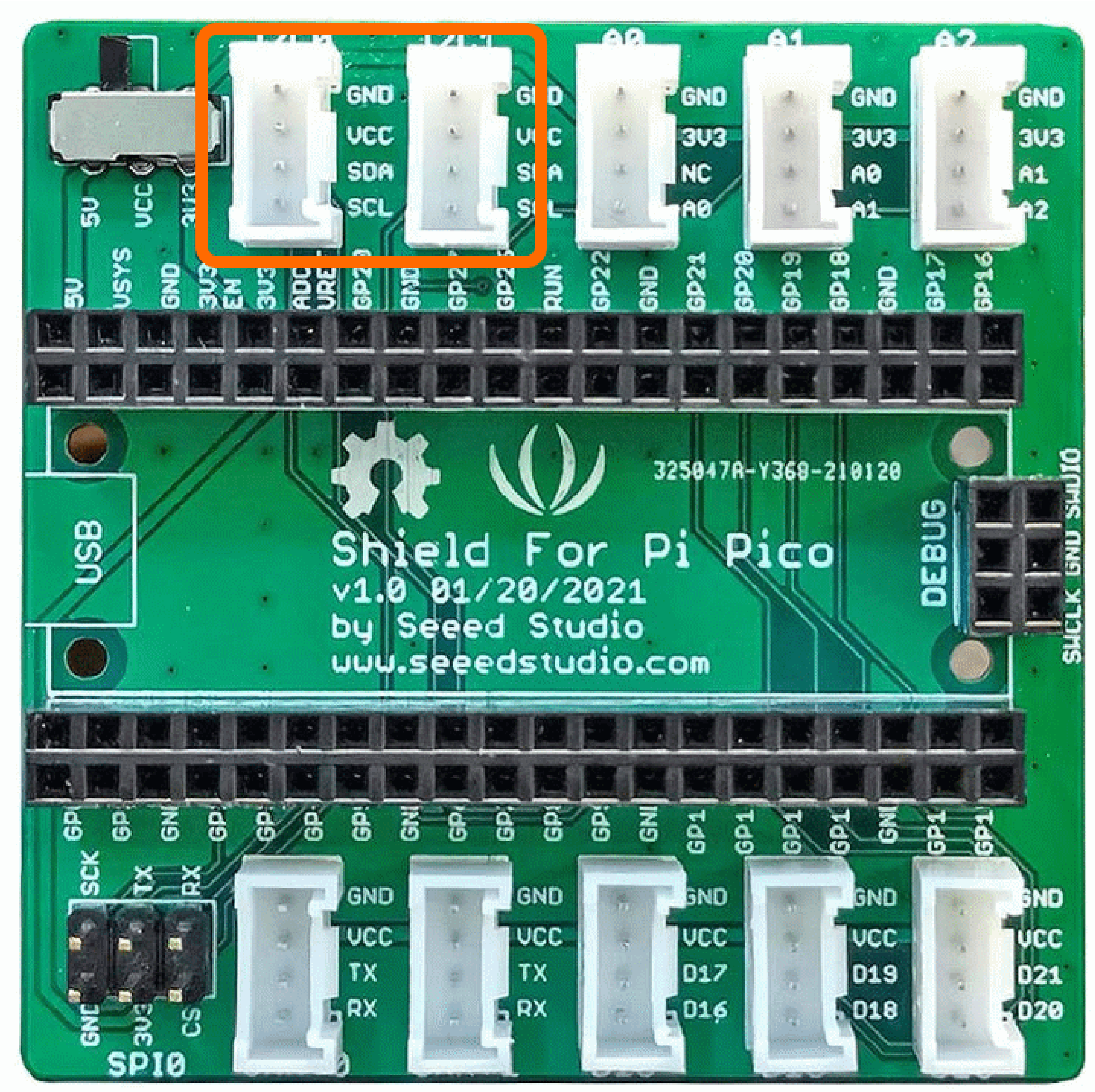


Various Sensors

Grove has a wide variety of sensors that can be communicated to using I2C.

Grove I2C Sensors

There are also options that communicate using I2C. These will use connectors I2C0 and I2C1 as shown. Commonly these will offer improved resolution e.g. 16 bits or 24 bits, so 65536 levels or 16,777,215 level respectively. They may also offer amplifiers, multiplexers or other enhanced functionality.



Grove I2C Temperature Humidity Sensor DH20

Overview

The Grove - Temperature & Humidity Sensor is based on the DHT20 sensor. The DH20 is an upgraded version of the DHT11, compared with the previous version, the temperature and humidity measurement accuracy are higher, and the measurement range is larger. It features I2C output which means it is easier to use.

Hardware Details

Grove I2C Temperature Humidity Sensor DH20

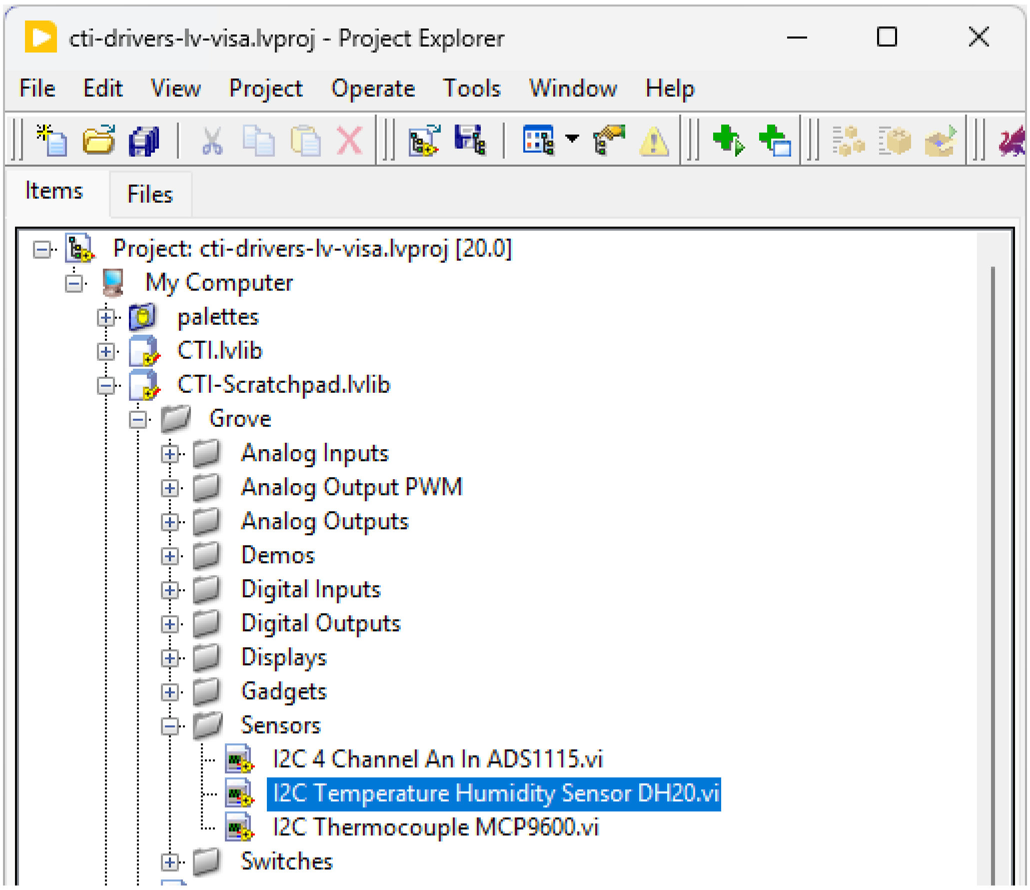
Demo Video

Here is a video that shows the set-up and running of the Grove I2C Temperature Humidity Sensor

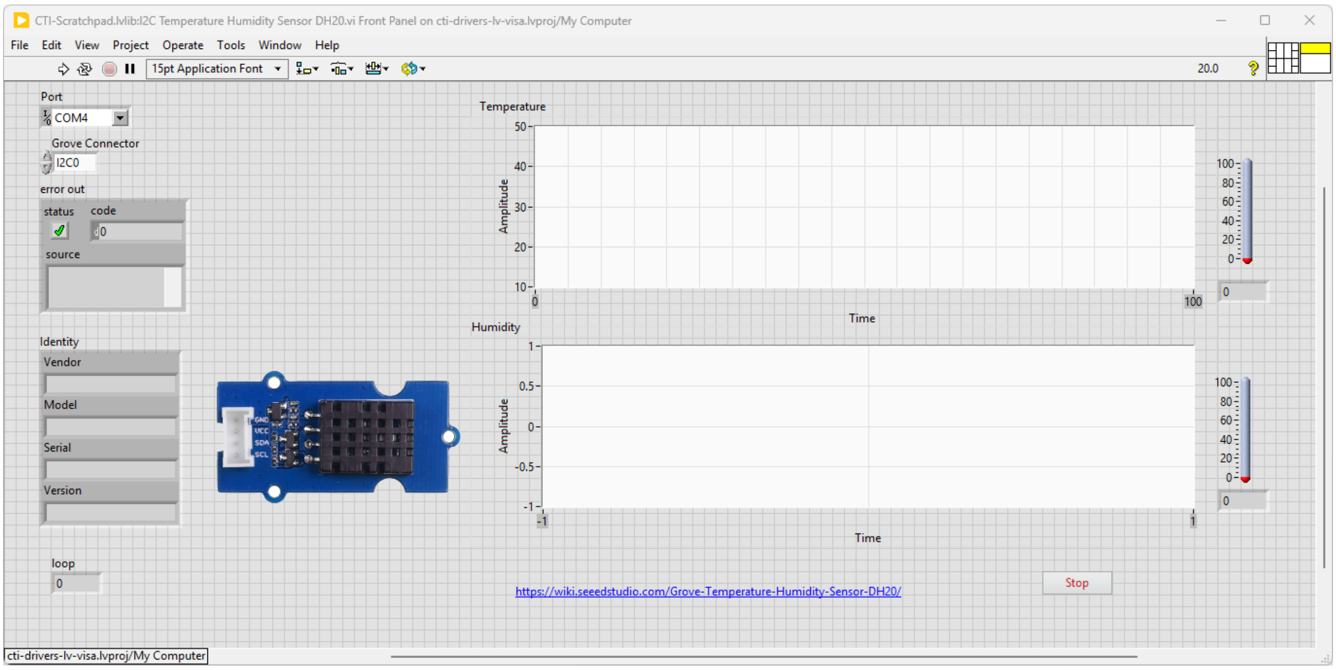
▶ https://www.youtube.com/watch?v=ENfYFbsgunE *(YouTube video)*

Example Code

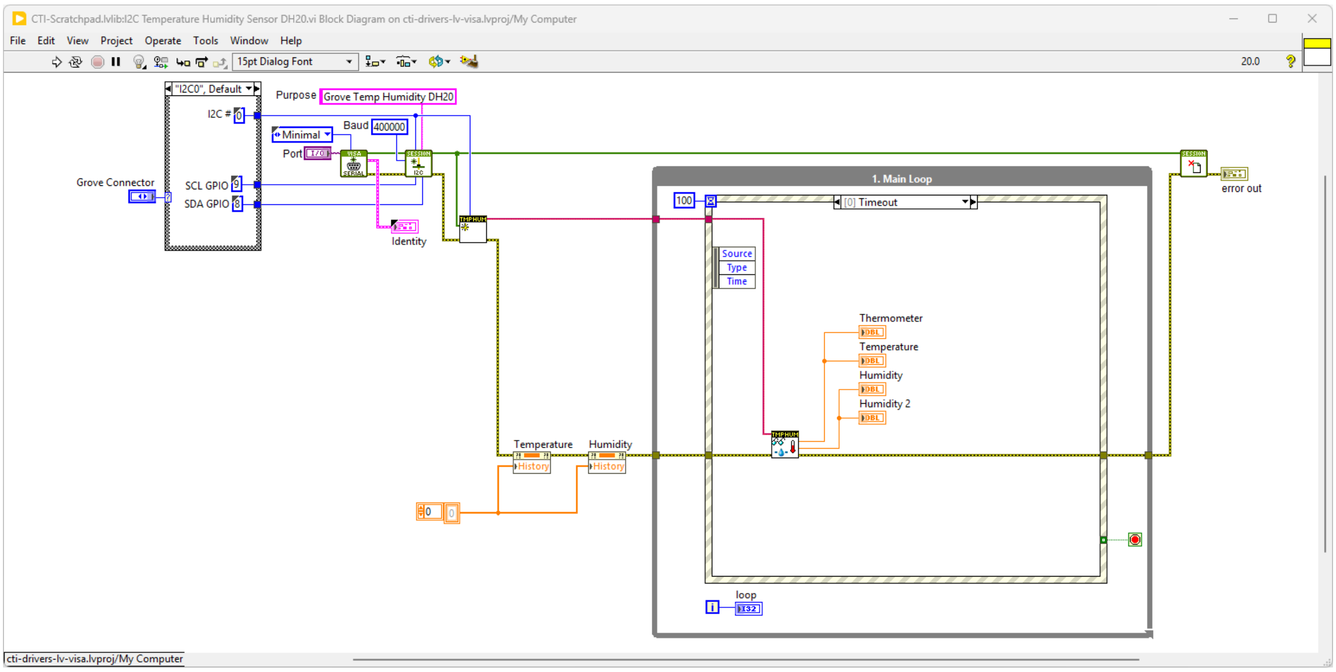
Navigate to >>Scratchpad>>Grove>>Sensors>>**I2C Temperature Humidity Sensor DH20.vi**



This VI uses Grove I2C Temperature Humidity DH20 Sensor. Select the port for the connected Pico and the Grove connector that the board is plugged into. Press the run arrow.



On the block diagram you can see that the selected Grove connector is input into **Grove Temperature Humidity Sensor DH20.lvclass:Init.vi**. Next we loop round the event structure and use the timeout to poll the Sensors using **Grove Temperature Humidity Sensor DH20.lvclass:Read Temp Humidity.vi**. Pressing Stop will fire the Stop event and exit the loop.



Grove I2C Thermocouple Sensor

Overview

The Grove - I2C Thermocouple Amplifier (MCP9600) is a thermocouple-to-digital converter with integrated cold-junction and I2C communication protocol. This module is designed to be used in conjunction with a k-type thermocouple. The thermocouples have a much larger measurement range than thermistors.

Hardware Details

Grove I2C Thermocouple Amplifier MCP9600

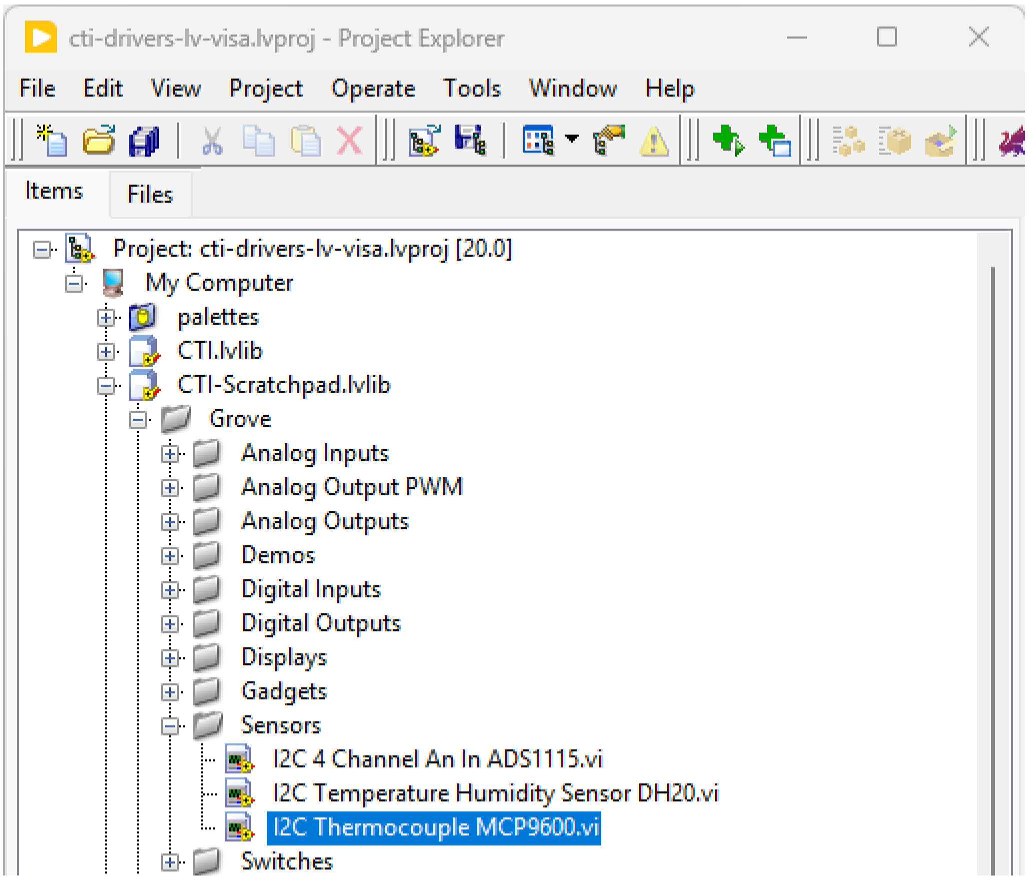
Demo Video

Here is a video that shows the set-up and running of the Grove I2C Thermocouple Amplifier MCP9600

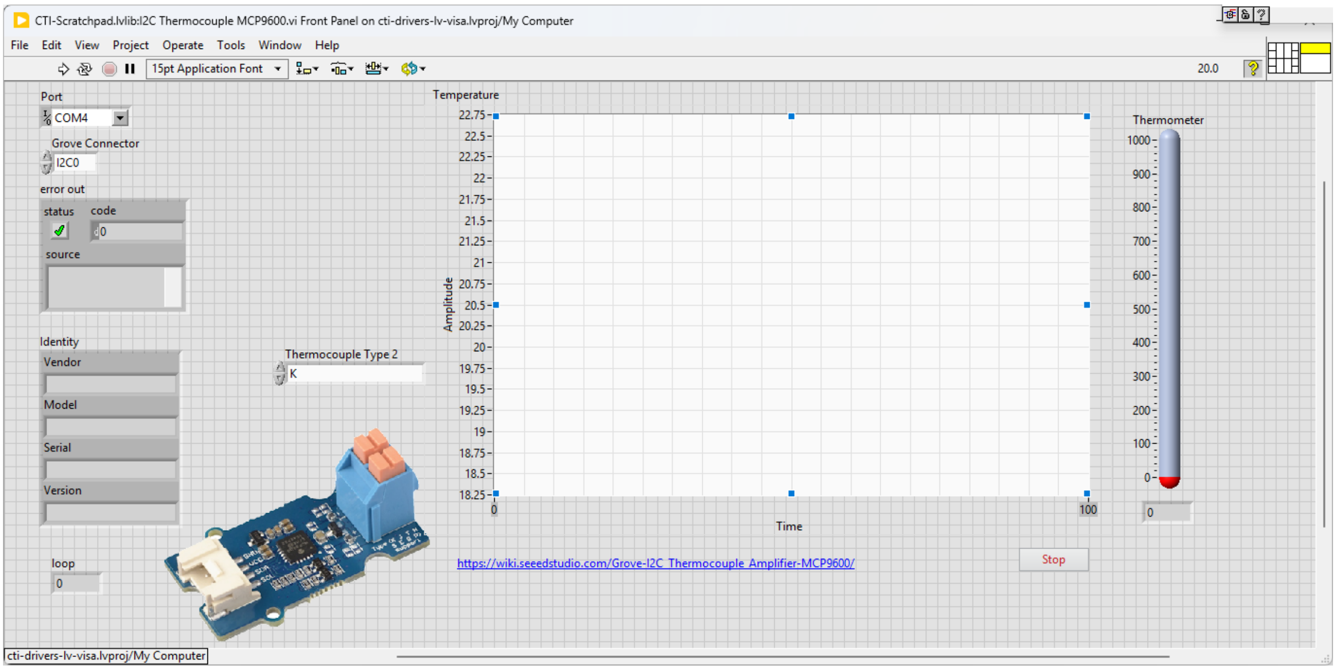
▶ https://www.youtube.com/watch?v=BdaQqxTZXwk *(YouTube video)*

Example Code

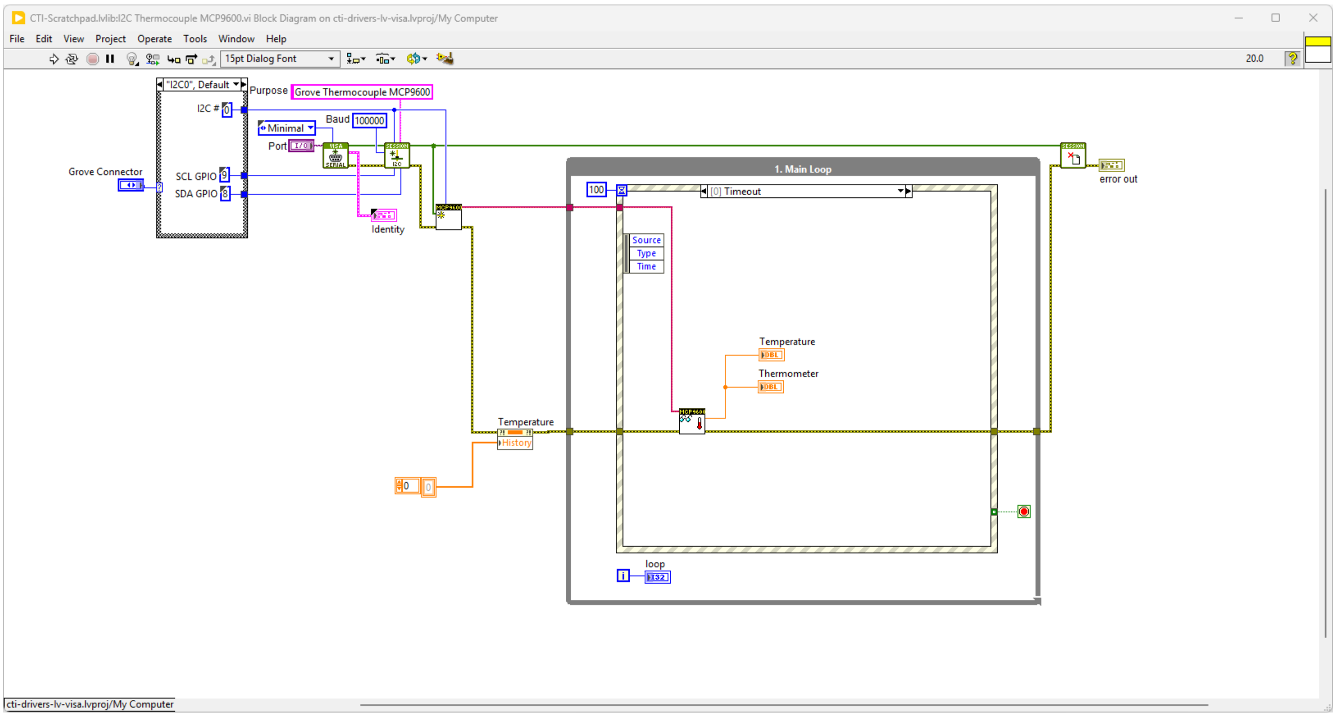
Navigate to >>Scratchpad>>Grove>>Sensors>>**I2C Thermocouple MCP9600.vi**



This VI uses Grove I2C Thermocouple Amplifier MCP9600. Select the port for the connected Pico and the Grove connector that the board is plugged into. Press the run arrow.



On the block diagram you can see that the selected Grove connector is input into **Grove Thermocouple MCP9600.lvclass:Init.vi**. We then can set the thermocouple type from the Thermocouple Type Value Change event and **Grove Thermocouple MCP9600.lvclass:Set Thermocouple Type.vi**. Next we loop round the event structure and use the timeout to poll the Sensors using **Grove Thermocouple MCP9600.lvclass:Read Temperature.vi**. Pressing Stop will fire the Stop event and exit the loop.



Adafruit NAU7802 I2C Strain Gauge Module

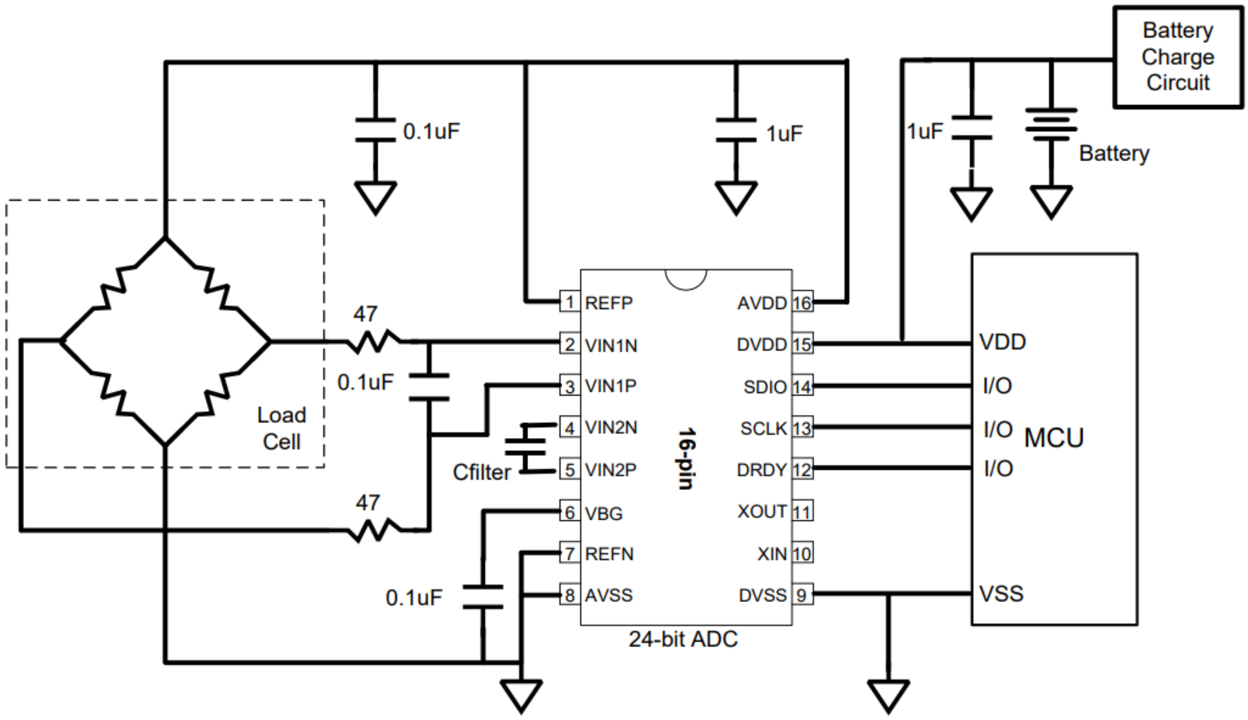
Overview

A strain gauge (also spelled strain gage) is a device used to measure strain on an object. Invented by Edward E. Simmons and Arthur C. Ruge in 1938, the most common type of strain gauge consists of an insulating flexible backing which supports a metallic foil pattern. As the object is deformed, the foil is deformed, causing its electrical resistance to change. This resistance change, usually measured using a Wheatstone bridge, is related to the strain by the quantity known as the gauge factor.

From Wikipedia

Hardware Details

The NAU7802 module contains a super-high-resolution 24-Bit differential ADC with extra gain and calibration circuitry that makes it perfect for measuring strain gauges / load cells or other sensors that have four wires that are connected in a Wheatstone bridge arrangement.



Each breakout comes with a NAU7802 ADC chip, plus some support circuitry, and 4 terminal blocks that can be used to connect a 4-wire sensor.

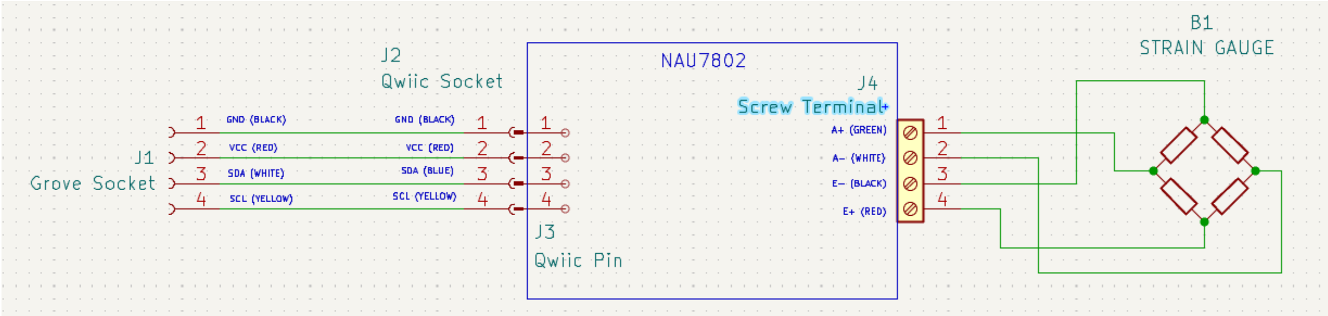
NAU7802 Datasheet

NAU 7802 Strain Gauge Module

Strain Gauge

This board uses the Qwiic connector, so we will need to use a Qwiic to Grove lead.. These can be bought directly or made to the circuit diagram below.

The wiring diagram for the board will therefore look like this.



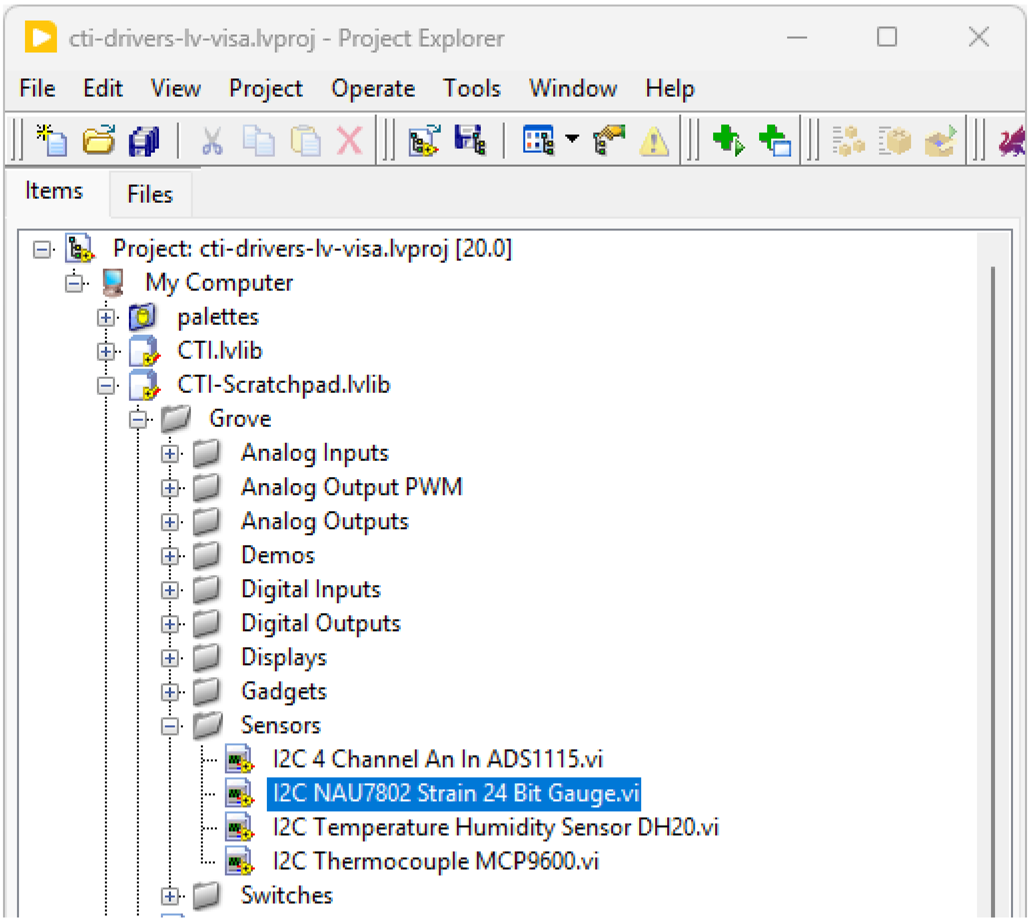
Demo Video

Here is a video that shows the set-up and running of the Adafruit NAU7802 I2C Strain Gauge Module

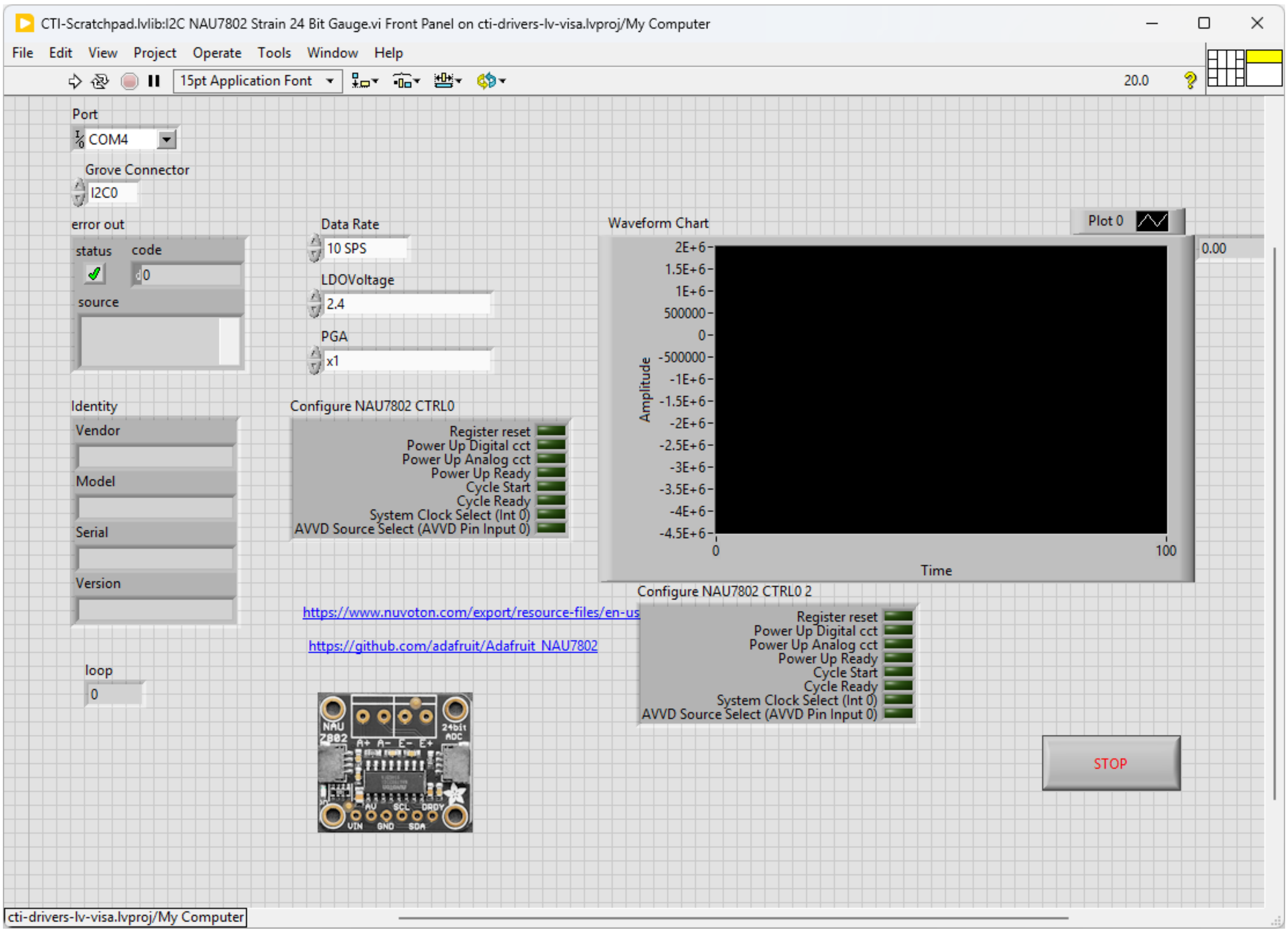
▶ https://www.youtube.com/watch?v=dy8eHkbQ16g *(YouTube video)*

Example Code

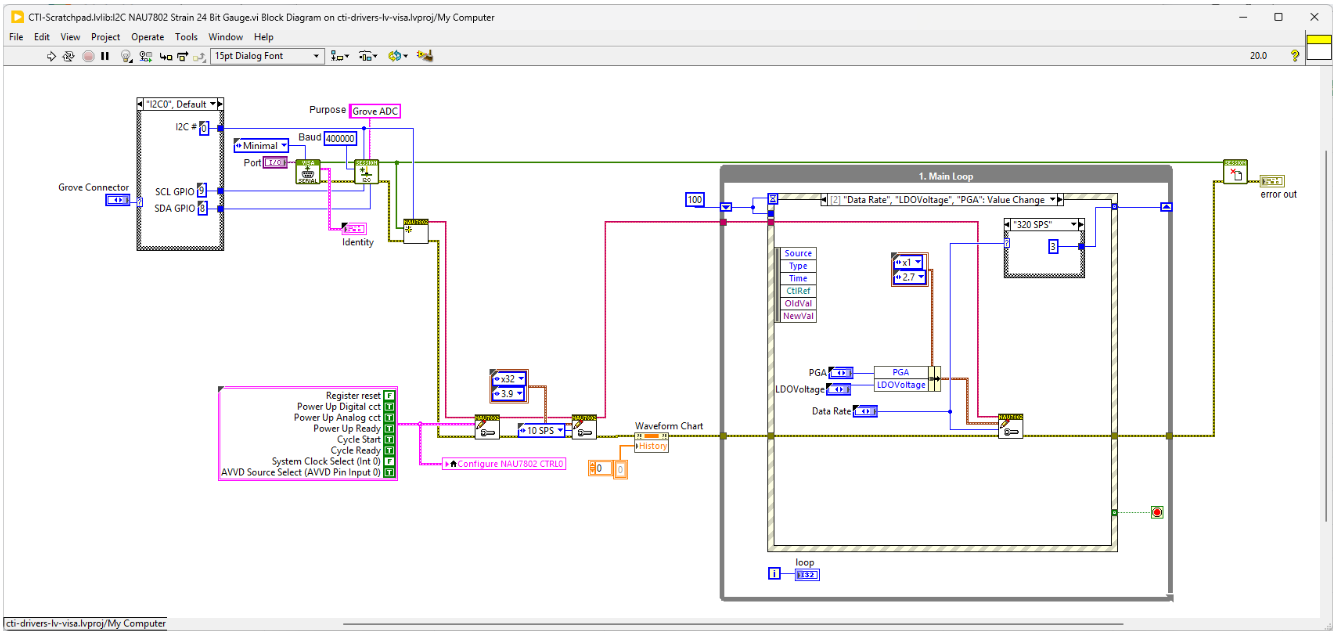
Navigate to >>Scratchpad>>Grove>>Sensors>>**I2C NAU7802 Strain 24 Bit Gauge.vi**



This VI uses NAU7802 Strain Gauge IC. Select the port for the connected Pico and the Grove connector that the board is plugged into. Press the run arrow.

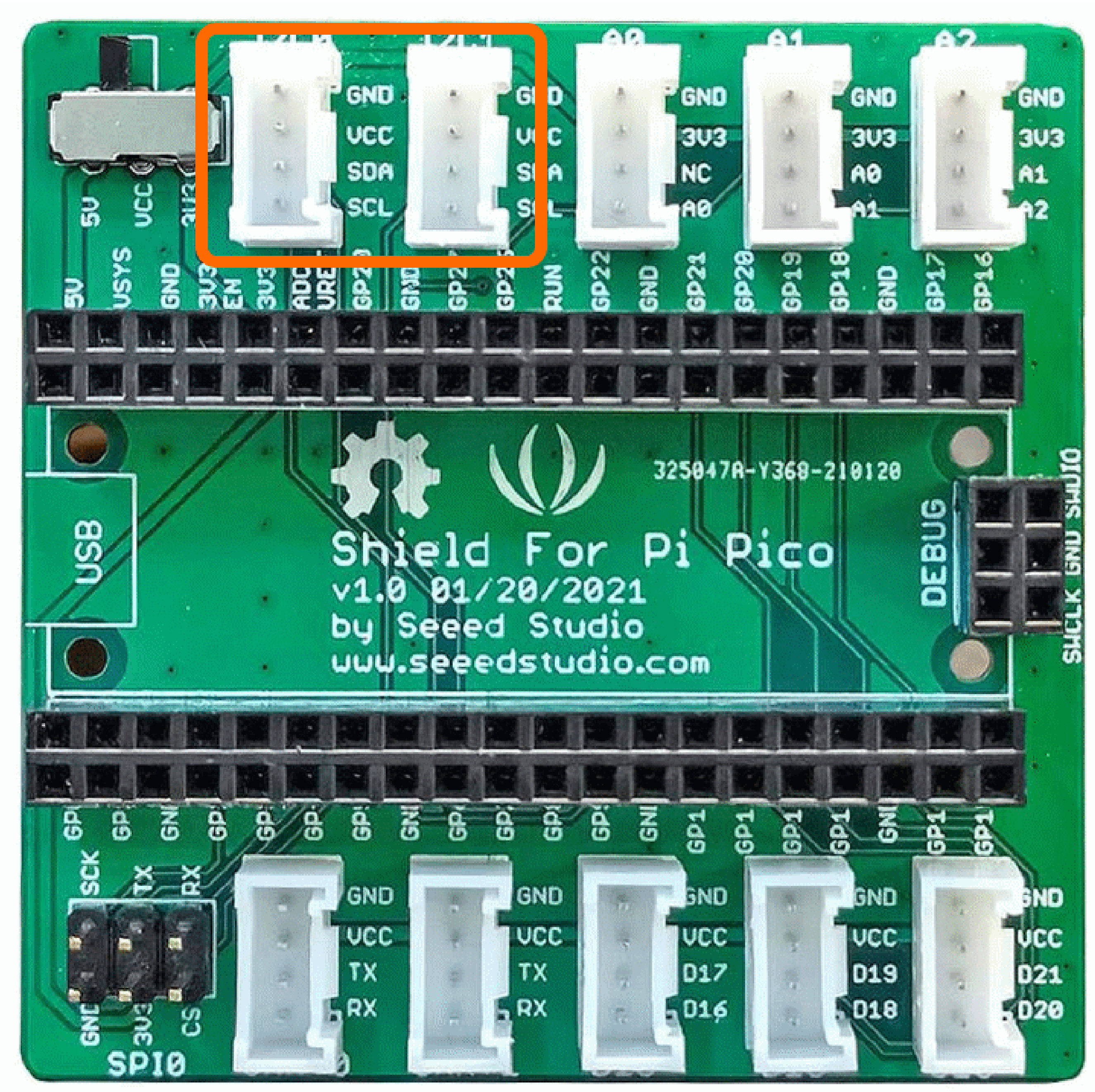


On the block diagram you can see that the selected Grove connector is input into **Adafruit NAU7802 24bit Strain Gauge.lvclass:Init.vi**. We then can set PU_CTRL Register to start measuring using **Adafruit NAU7802 24bit Strain Gauge.lvclass:Set PU_CTRL.vi** and set up the ADC using **Adafruit NAU7802 24bit Strain Gauge.lvclass:Set ADC.vi** Next we loop round the event structure and use the timeout to poll the Sensors using **Adafruit NAU7802 24bit Strain Gauge.lvclass:Read PU_CTRL.vi** if the Cycle Ready bit is high it means there is a reading available. The timeout of the event loop is adjusted to suit the Samples Per Second. If a reading is available we get the 24 bits using **Adafruit NAU7802 24bit Strain Gauge.lvclass:Read ADC.vi**. Pressing Stop will fire the Stop event and exit the loop.


CTI-Scratchpad.lvlib:I2C NAU7802 Strain 24 Bit Gauge.vi Block Diagram on cti-drivers-lv-visa.lvproj/My Computer
File Edit View Project Operate Tools Window Help
15pt Dialog Font
Grove Connector
"I2C0", Default
I2C #
SCL GPIO
SDA GPIO
Purpose
Grove ADC
Baud
400000
Minimal
Port
Identity
1. Main Loop
100
[2] "Data Rate", "LDOVoltage", "PGA": Value Change
Source
Type
Time
CtlRef
OldVal
NewVal
"320 SPS"
PGA
LDOVoltage
Data Rate
Register reset
Power Up Digital cct
Power Up Analog cct
Power Up Ready
Cycle Start
Cycle Ready
System Clock Select (Int 0)
AVVD Source Select (AVVD Pin Input 0)
Configure NAU7802 CTRL0
x32
3.9
10 SPS
Waveform Chart
History
x1
2.7
loop
error out
cti-drivers-lv-visa.lvproj/My Computer

Various Displays

Grove has a wide variety of displays that can be communicated to using I2C.



HD44870 LCD Display with RGB Backlight Controller

Overview

The LCD display is a useful, robust low power way show system information. The RGB backlight also can convey extra information like Warnings and Alerts.

Hardware Details

Grove - LCD RGB Backlight

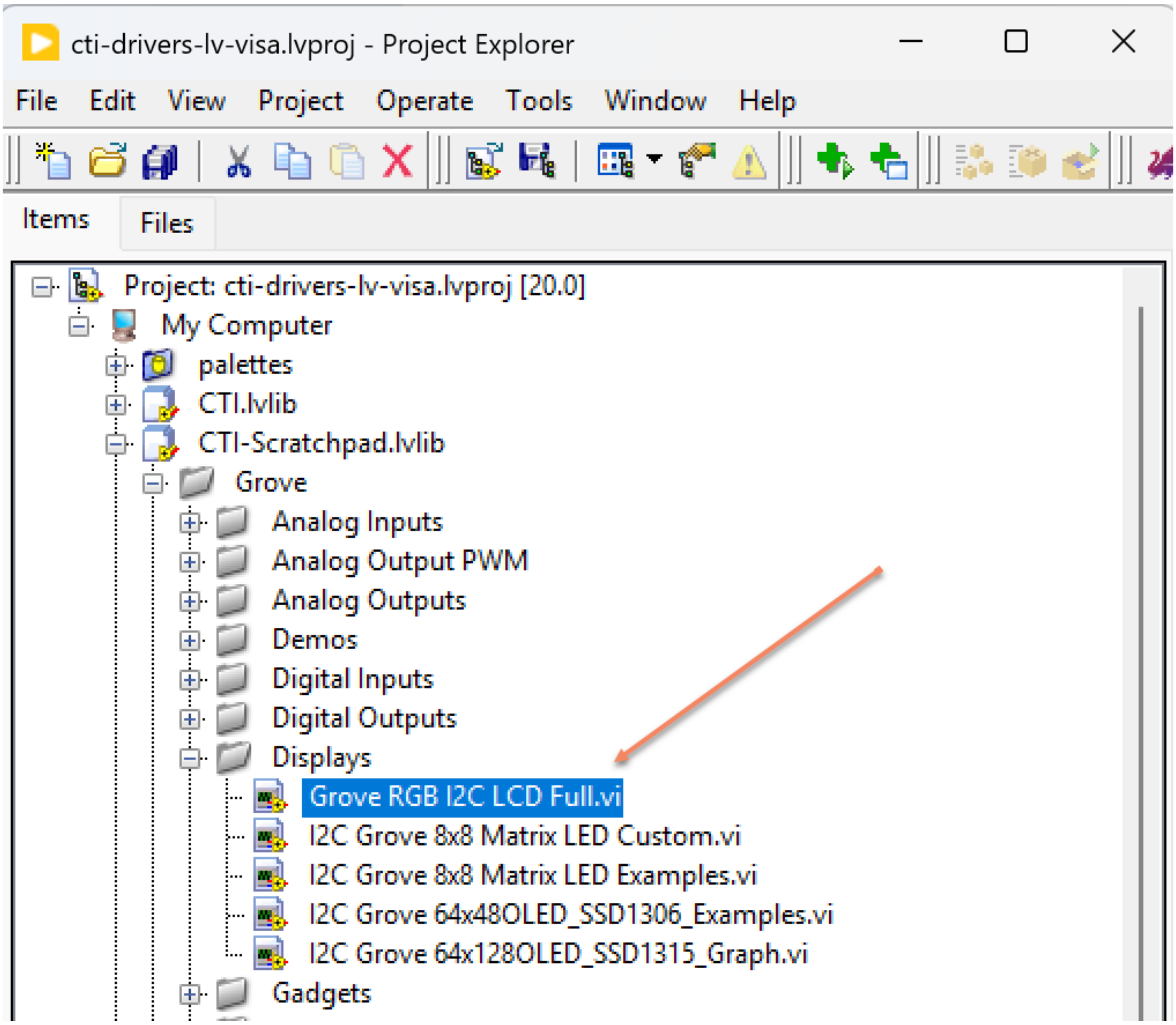
Demo Video

Here is a video that shows the set-up and running of the LCD Display with RGB Backlight.

▶ https://www.youtube.com/watch?v=_c_6OBtgLvg *(YouTube video)*

Example Code

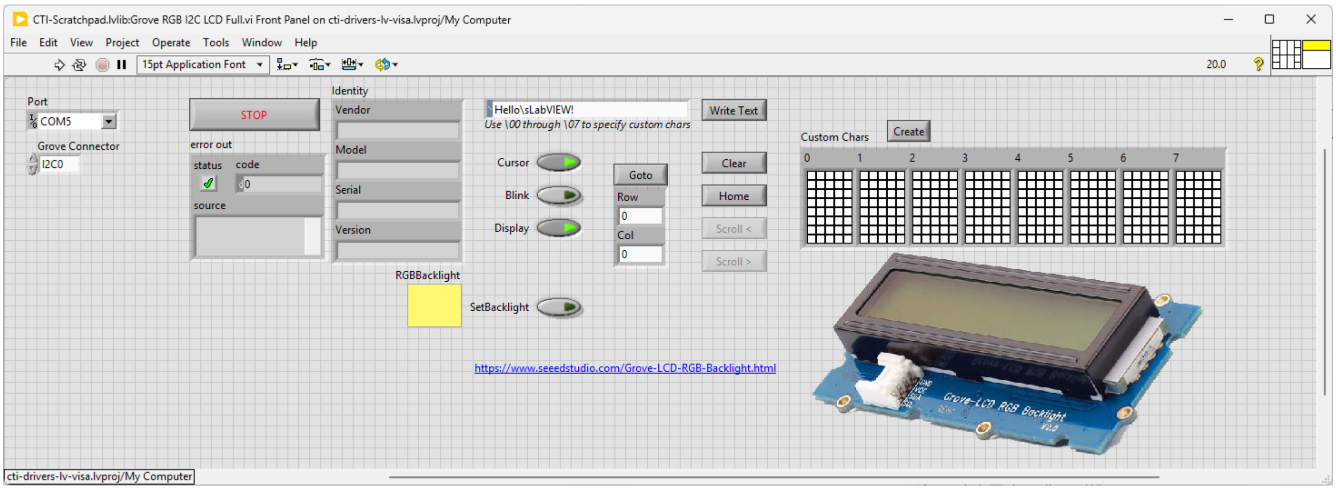
Navigate to >>Scratchpad>>Grove>>Displays>>Grove RGB I2C LCD Full.vi



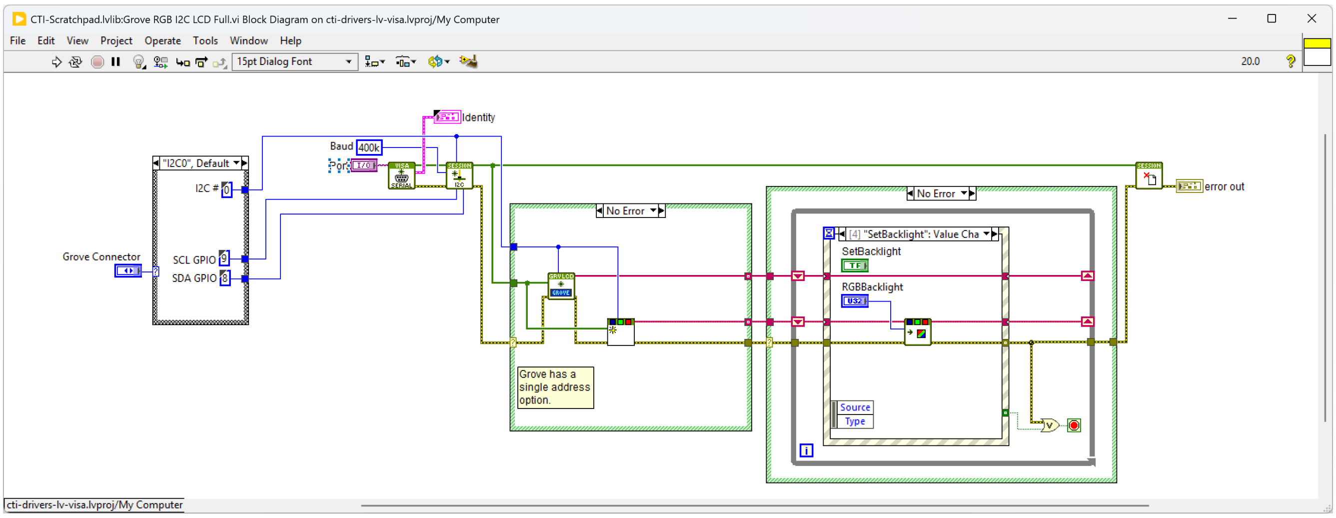
This VI uses sends text to the LCD display and also changes the backlight color. It communicates to 2 different I2C addresses to achieve this.

Select the port for the connected Pico and the Grove connector that the board is plugged into. Press the run arrow.

I've only implemented and tested the basic functionality for this board, so Write Text works, Clear and Home works and SetBacklight.



dictates the GPIO Pin for the IC2 Port. We set up the IC2 port for the device in **Grove I2C LCD.lvclass:Init.vi** and **Grove I2C RGB SGM31323.lvclass:Init.vi**. Next we loop round the event structure and wait for a button on the Front Panel to be pressed. When a button is detected (event:button>>Value Change will fire) the associated VI will run. Explore the different events to see how they operate. Pressing Stop will fire the Stop event and exit the loop.



SSD1315 64x128 OLED Display

Overview

This example shows how to use a 2D picture control to display on a 64x128 monochrome OLED display

Hardware Details

Grove - OLED Display 0.96" (SSD1315)

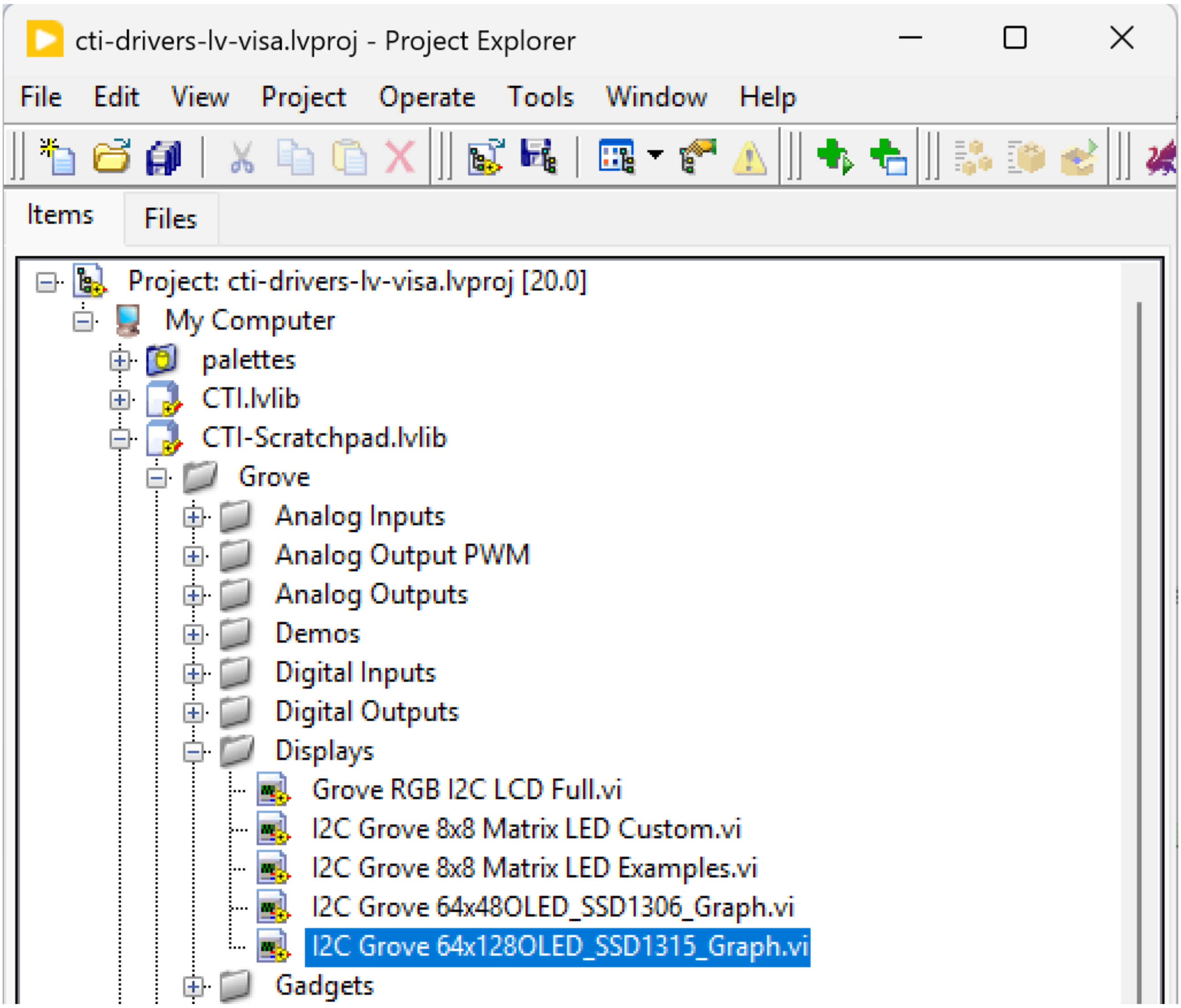
Demo Video

Here is a video that demonstrates how to use the 64x128 OLED display

https://www.youtube.com/watch?v=U05stmxbPzs *(YouTube video)*

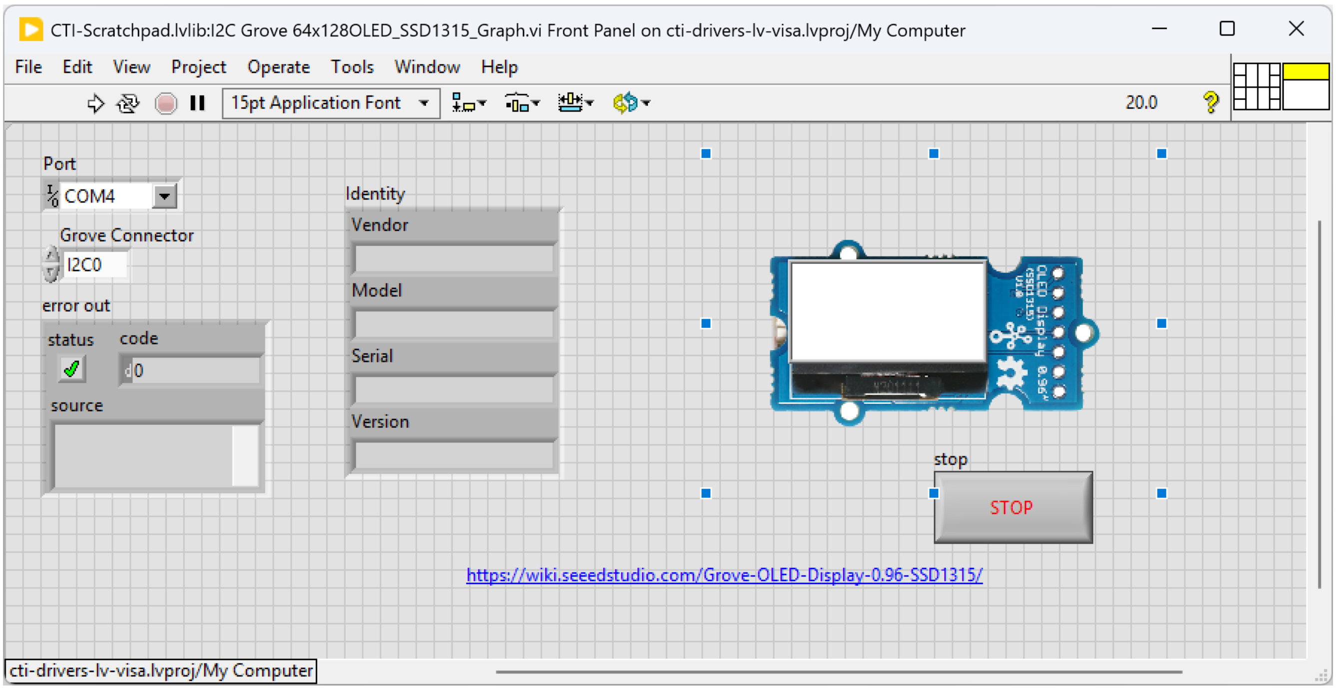
Example Code

Navigate to >>Scratchpad>>Grove>>Displays>>I2C Grove 64x128OLED_SSD1315_Graph.vi



This particular VI builds a small graph image in a 2D picture control. This picture control is converted to a monochrome image and this image is converted into messages in I2C. Select the port for the connected Pico and the Grove connector that the board is plugged into. Press the run arrow.

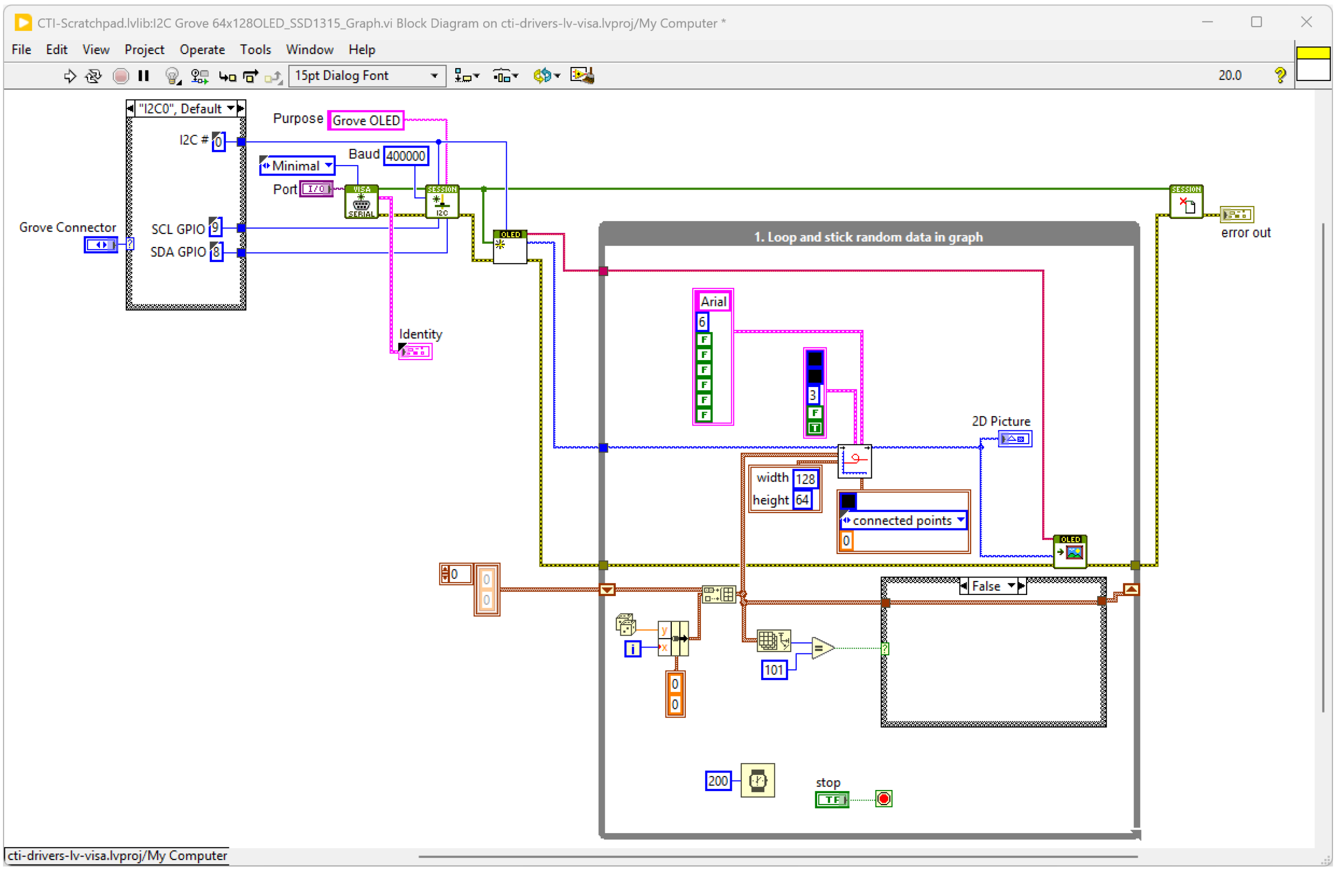
You should see the graph fillng up with random numbers.



On the block diagram you can see that the selected Grove connector dictates the GPIO Pin used by the I2C port.

We set up the IC2 port for the device in **64x128 I2C OLED SSD1315.lvclass:Init.vi**, this VI also outputs a picture of the correct dimensions as a template. The While Loop will now iterate every 200msecs adding a random number to the 100 element rolling buffer. This buffer is input to Plot XY.vi that creates the picture to fit our template. This picture is input into **64x128 I2C OLED SSD1315.lvclass:Write Image.vi** that converts and transmits all the I2C messages.

Pressing Stop will exit the loop.



SSD1306 64x48 OLED Display

Overview

This example shows how to use a 2D picture control to display on a tiny monochrome OLED display

Hardware Details

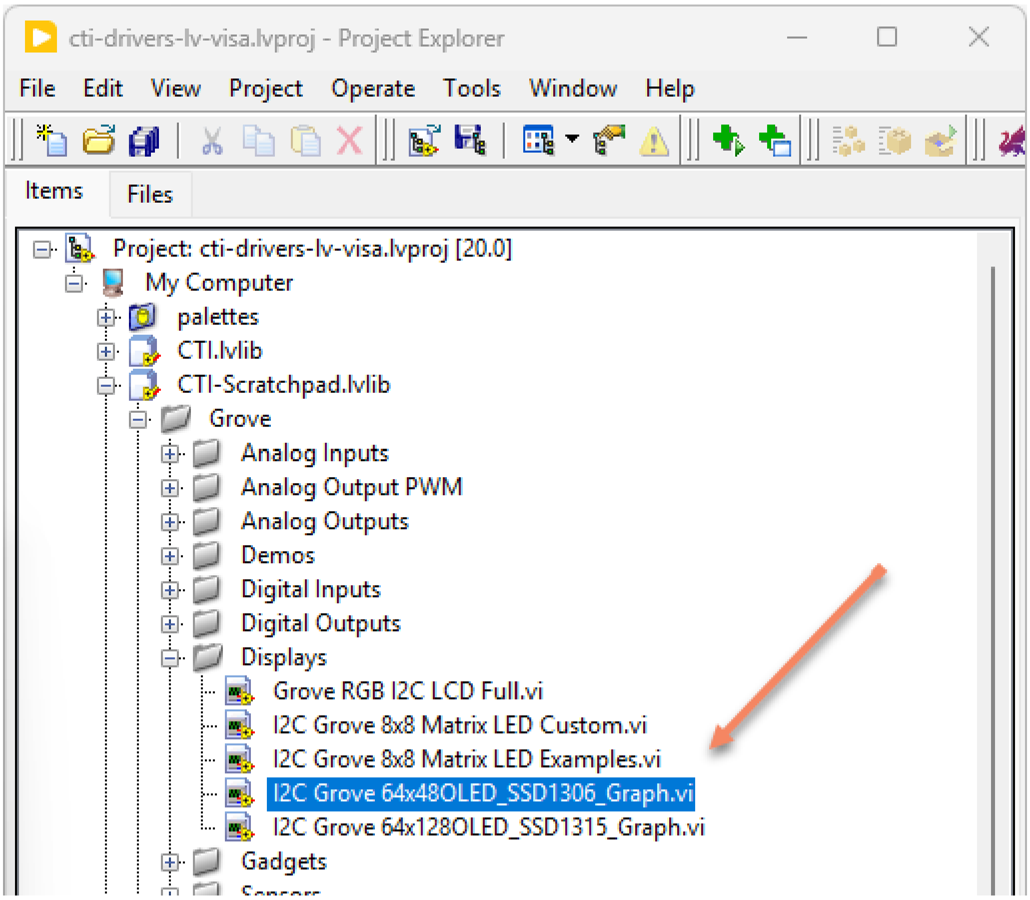
Grove - OLED Display 0.66" (SSD1306)

Demo Video

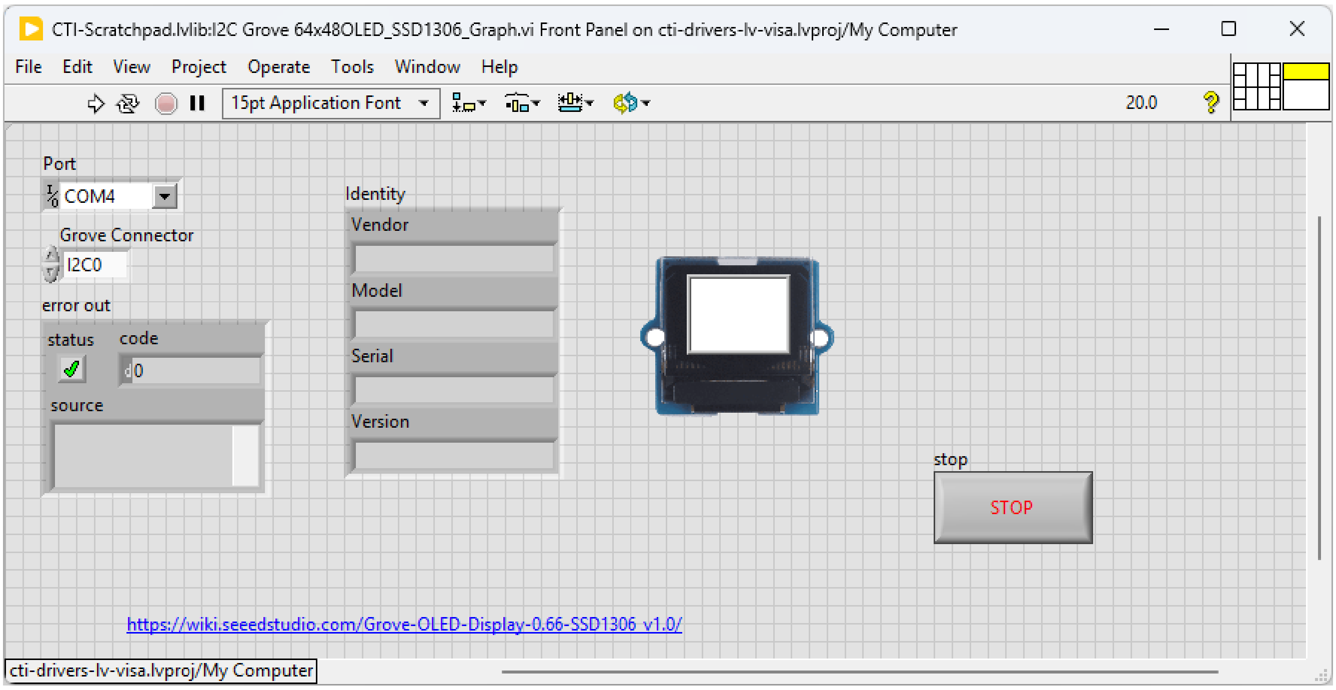
▶ https://www.youtube.com/watch?v=w0F2TISiCmE *(YouTube video)*

Example Code

Navigate to >>Scratchpad>>Grove>>Displays>>I2C Grove 64x48OLED_SSD1306_Graph.vi



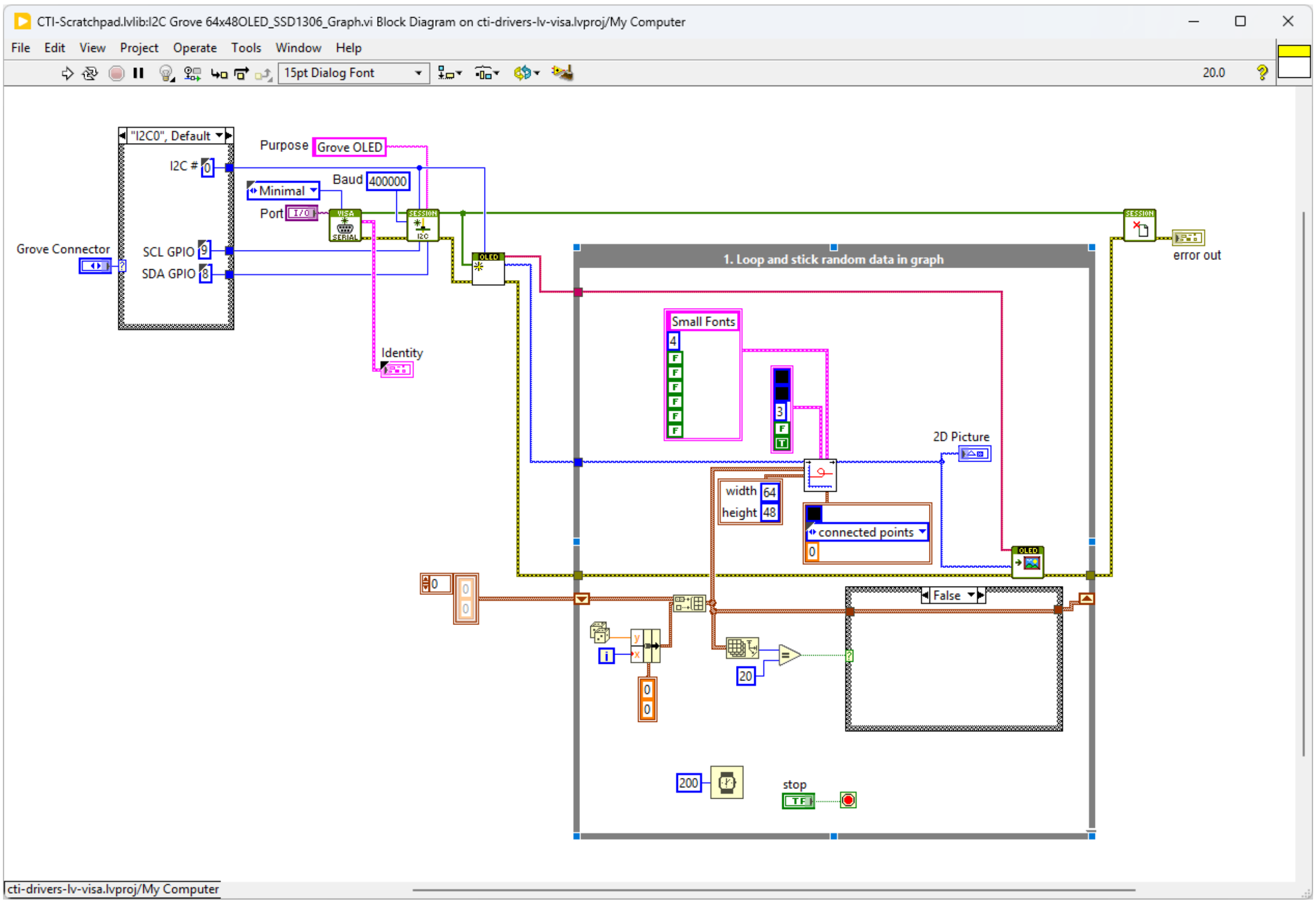
This particular VI builds a small graph image in a 2D picture control. This picture control is converted to a monochrome image and this image is converted into messages n I2C. Select the port for the connected Pico and the Grove connector that the board is plugged into. Press the run arrow.



On the block diagram you can see that the selected Grove connector dictates the GPIO Pin used by the I2C port.

We set up the IC2 port for the device in **64x48 I2C OLED SSD1306.lvclass:Init.vi**, this VI also outputs a picture of the correct dimensions as a template. The While Loop will now iterate every 200msecs adding a random number to the 100 element rolling buffer. This buffer is input to Plot XY.vi that creates the picture to fit our template. This picture is input into **64x48 I2C OLED SSD1306.lvclass:Write Image.vi** that converts and transmits all the I2C messages.

Pressing Stop will exit the loop.



8x8 RGB Matrix LED

Overview

8x8 RGB LED Matrix is great for simple image display, 64 pixel leds and 255 colors for each pixel means almost infinite possibilities. However, the complicated wiring of the matrix is daunting. Now we present the Grove - RGB LED Matrix w/Driver for you, leave all the complex and variable wireing and soldering behind, just one single grove connector to control the RGB 8x8 LED matrix easily.

Hardware Details

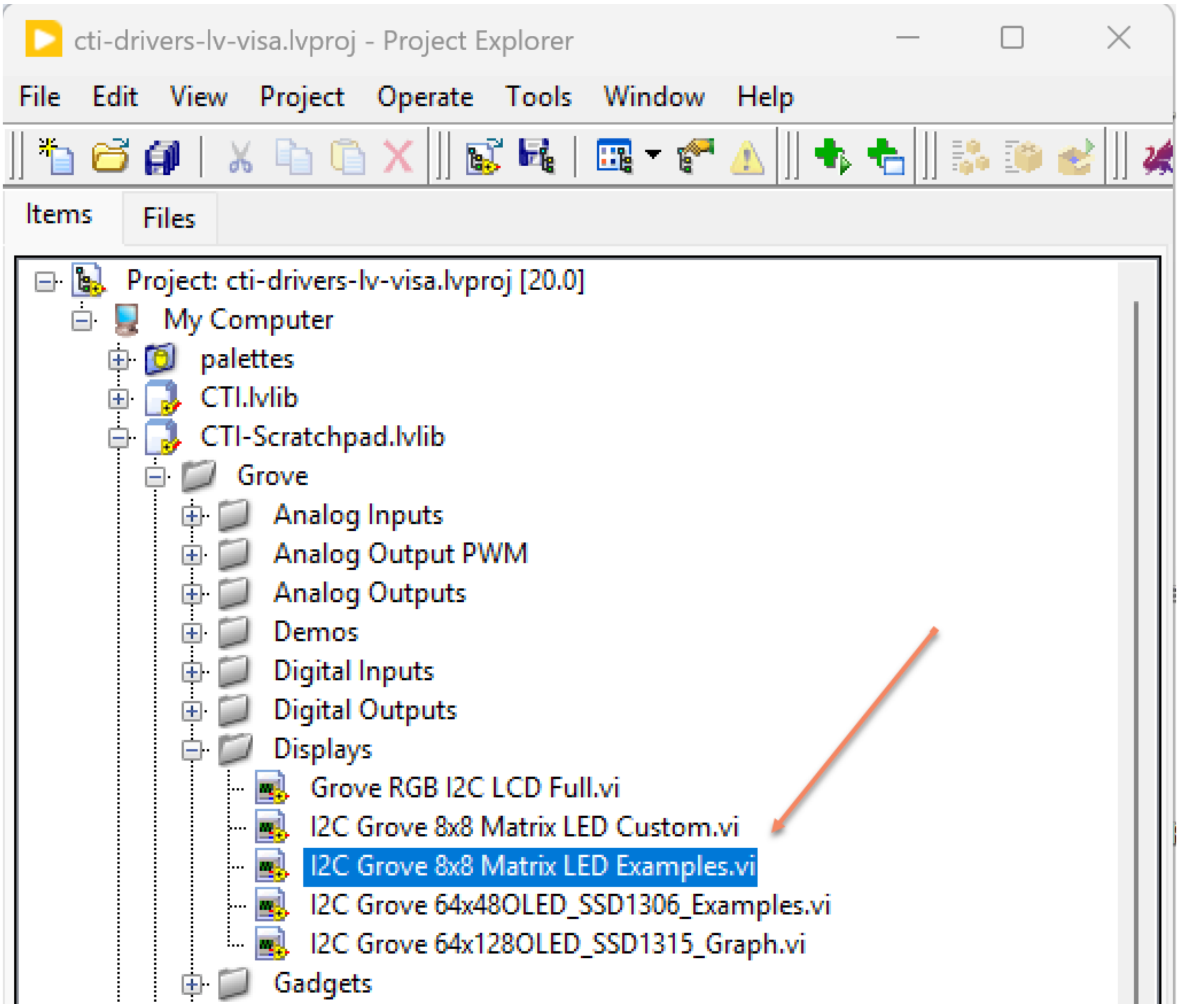
Grove - RGB LED Matrix w/Driver

Demo Video

Here is a video that demonstrates the capabilities of the 8x8 RGB Matrix LED Board https://www.youtube.com/watch?v=OjOcmV7EbRs *(YouTube video)*

Example Code

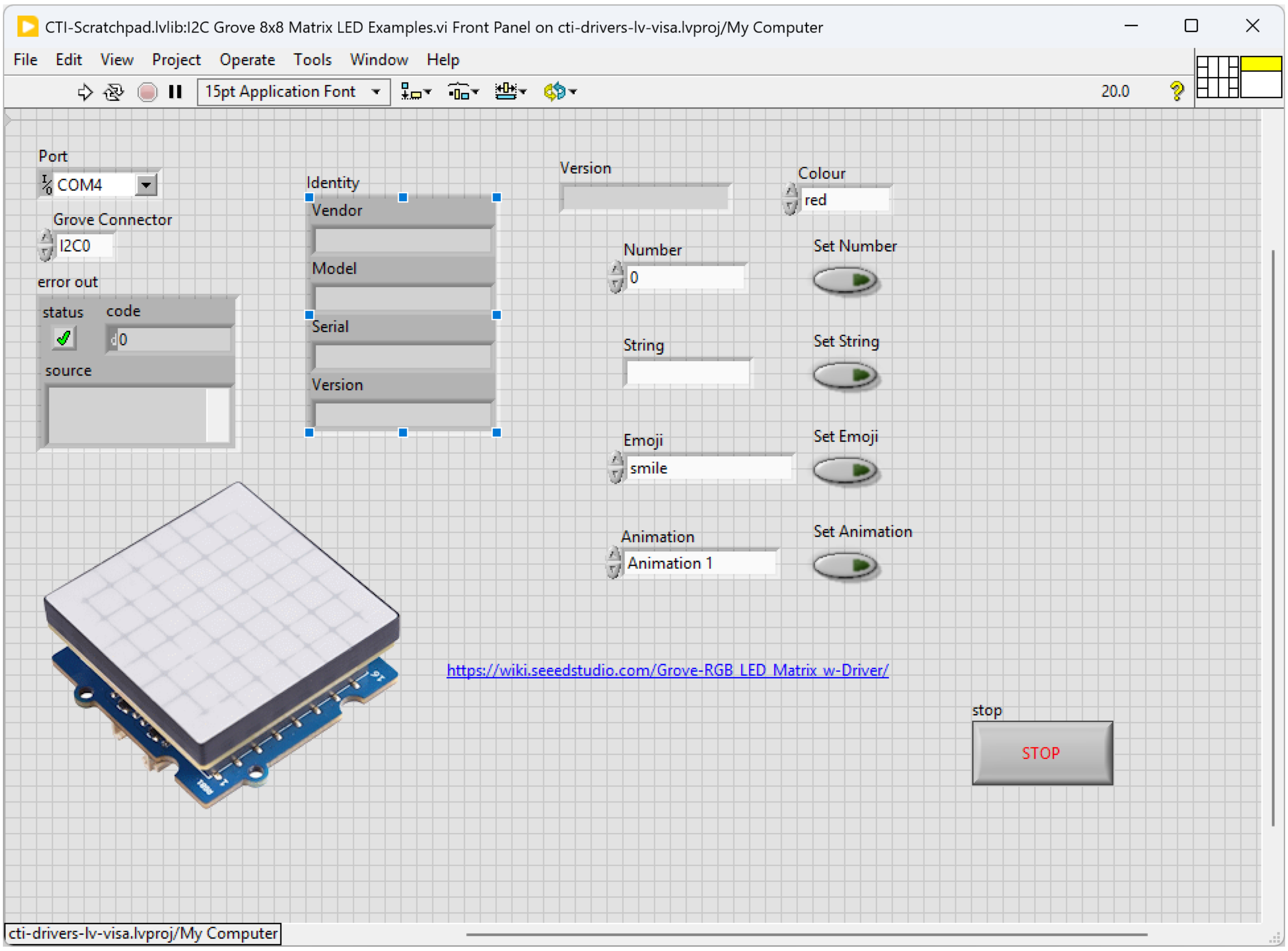
Navigate to >>Scratchpad>>Grove>>Displays>>I2C Grove 8x8 Matrix LED Examples.vi



Select the port for the connected Pico and the Grove connector that the board is plugged into. Press the run arrow.

Various pre-programed examples are available by selecting the colour from the Colour enum and then pressing Set Number or Set String to display your chosen number or string.

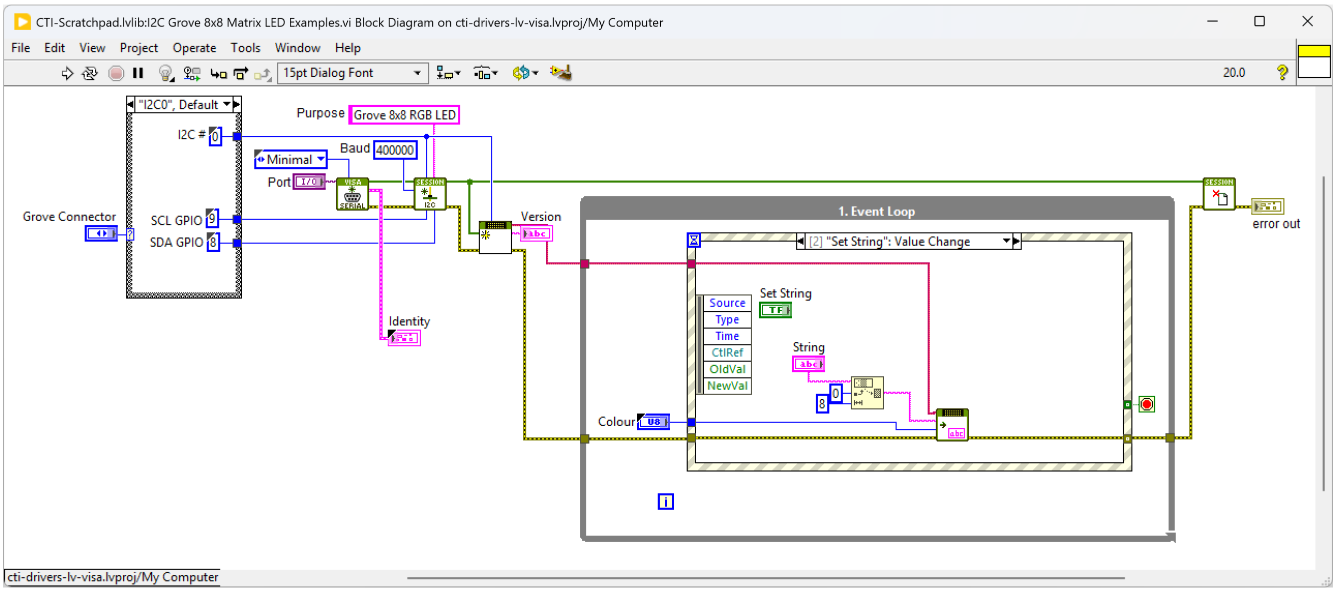
Set Emoji or Set Animation will display your selected Emoji or Animation.



We set up the IC2 port for the device in **Grove 2 RGB LED Matrix.lvclass:Init**

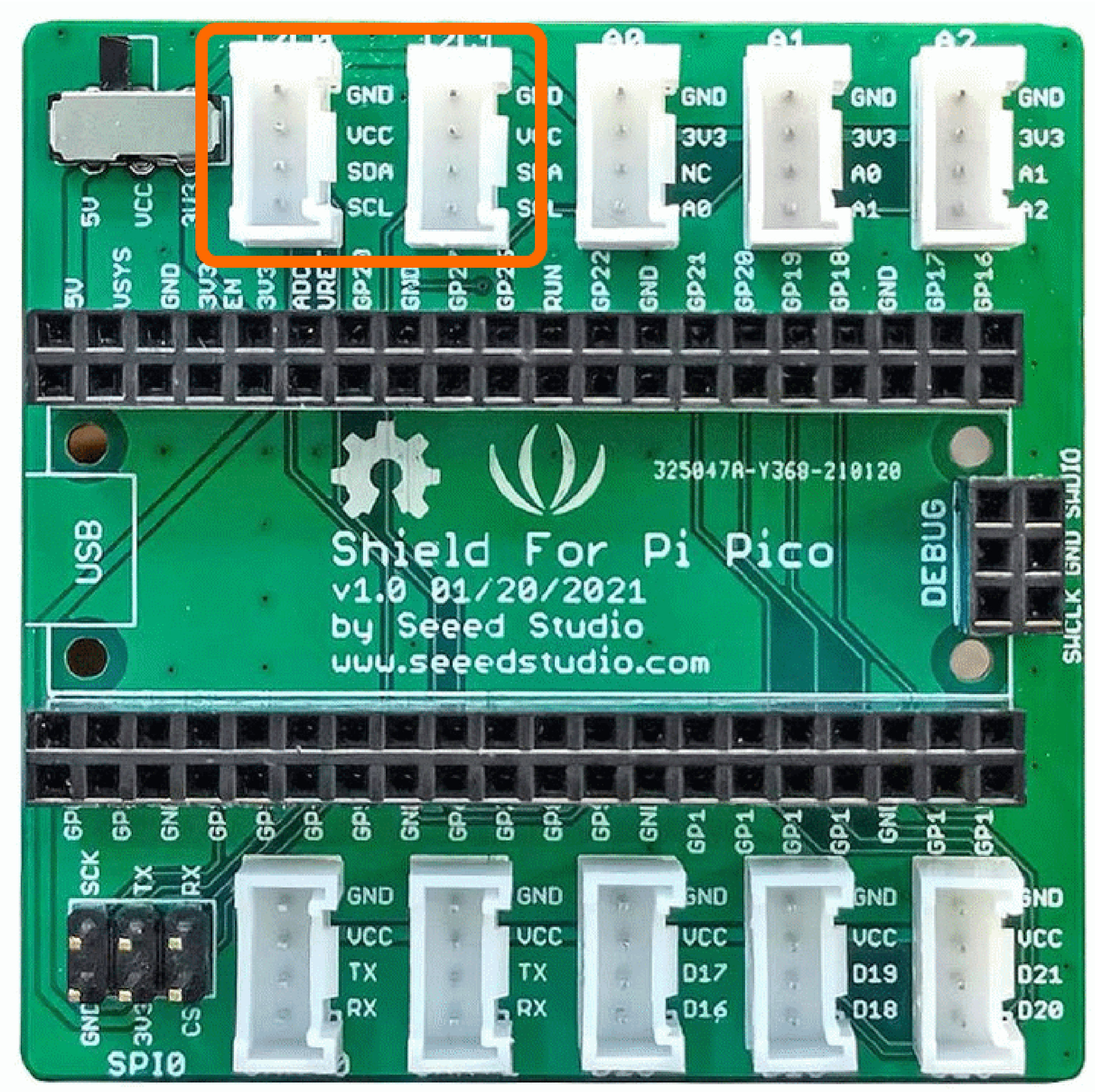
On the block diagram you can see that the selected Grove connector dictates the GPIO Pin for the I2C Connector. Next we loop round an event structure and use the detect button value change events for Set Number, Set String, Set Emojo or Set Animation. The fired event will use the associated data as an input to the selected function.

Pressing Stop will fire the Stop event and exit the loop.



Misc Gadgets Examples

Grove has a wide variety of fun toys that can be communicated to using I2C.



Grove Gesture Sensor

Overview

The sensor on Grove - Gesture is PAJ7620U2 that integrates gesture recognition function with general I2C interface into a single chip. It can recognize 9 basic gestures, and these gestures information can be simply accessed via the I2C bus.

Hardware Details

Grove - Gesture Sensor V1.0

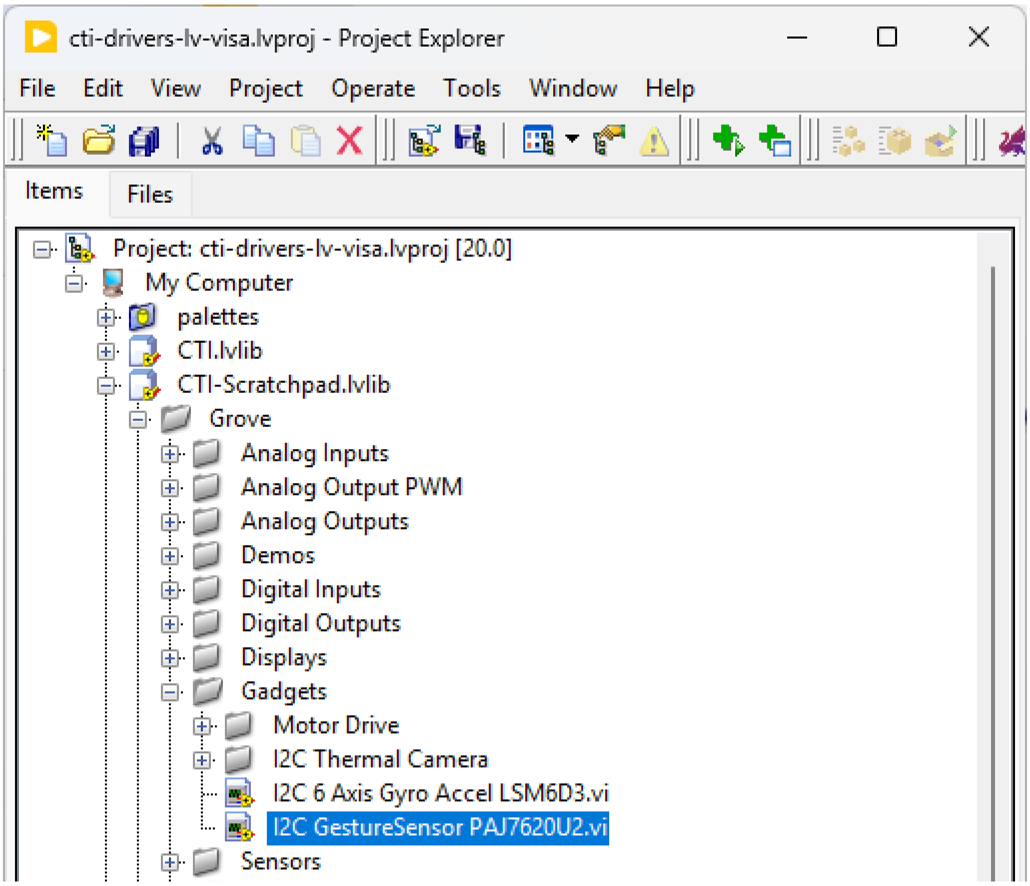
Demo Video

This video demonstrates the gesture sensor recognising a gesture and displaying it on the front panel.

▶ https://www.youtube.com/watch?v=5MPWDghnn5Q *(YouTube video)*

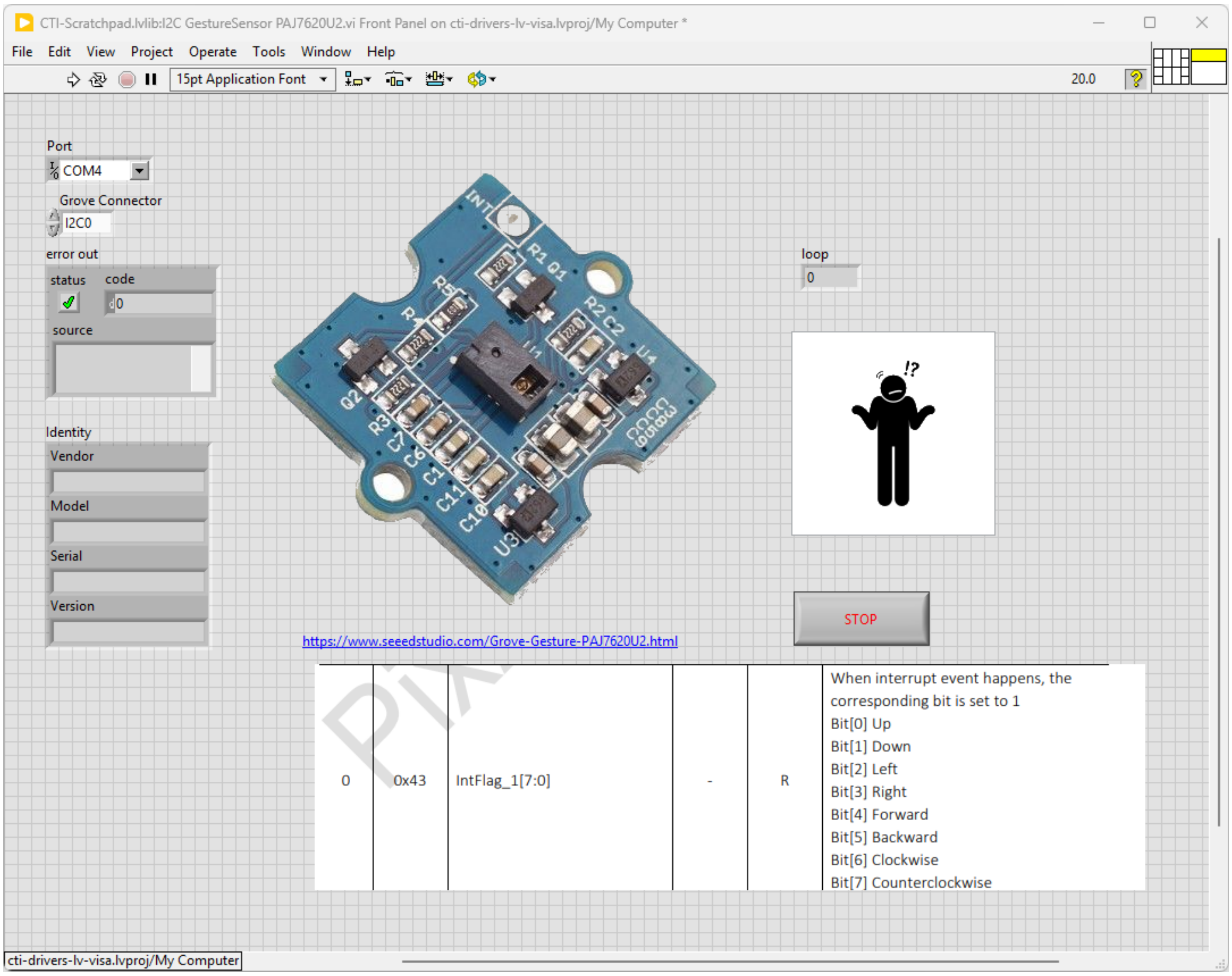
Example Code

Navigate to >>Scratchpad>>Grove>>Gadgets>>I2C GestureSensor PAJ7620U2.vi



Select the port for the connected Pico and the Grove connector that the board is plugged into. Press the run arrow.

The device will pick up various gestures generated by flapping your hands around in front of it. Works best when the background is not to busy and lit up with light-sources.



On the block diagram you can see that the selected Grove connector dictates the GPIO Pins used for communication. This session connection is input into **Grove Gesture Sensor PAJ7620U2.lvclass:Init**.

Next we loop round and the while loop and read the detected gesture using **Grove Gesture Sensor PAJ7620U2.lvclass:ReadGesture.vi** which returns a numbe corresponding with the detected gesture. A picture ring is used to give a nice graphical flourish on the front panel.

Pressing Stop will exit the loop.

[GroveI2CGestureSensor PAJ7620U2 BD] | *GroveI2CGestureSensor-PAJ7620U2-BD.png*

Grove 6 Axis Gyro Accellerometer

Grove - 6-Axis Accelerometer&Gyroscope is a cost-effective Grove interfaced and integrated sensor combination of 3-axis digital accelerometer and 3-axis digital gyroscope.

Hardware Details

Grove - 6-Axis Accelerometer & Gyroscope

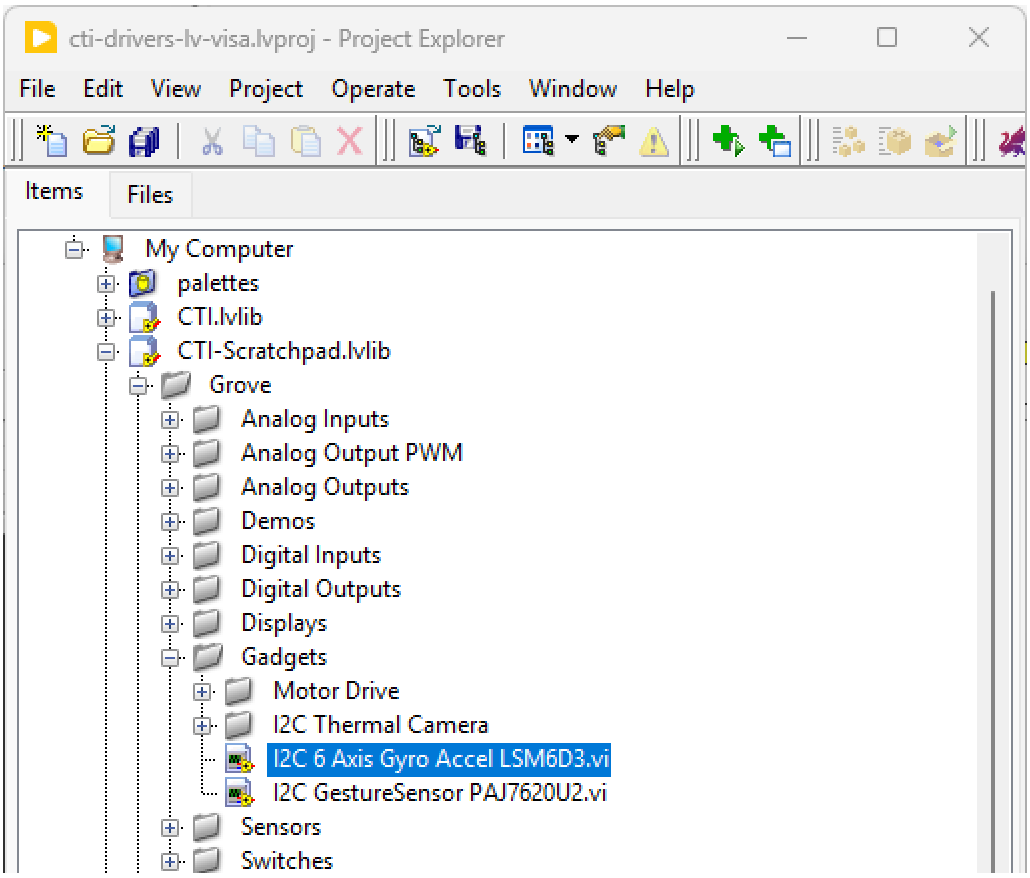
Demo Video

Shows the Grove - 6-Axis Accelerometer & Gyroscope showing it's axes being displayed on a graph.

▶ https://www.youtube.com/watch?v=Ou9bYsspwoc *(YouTube video)*

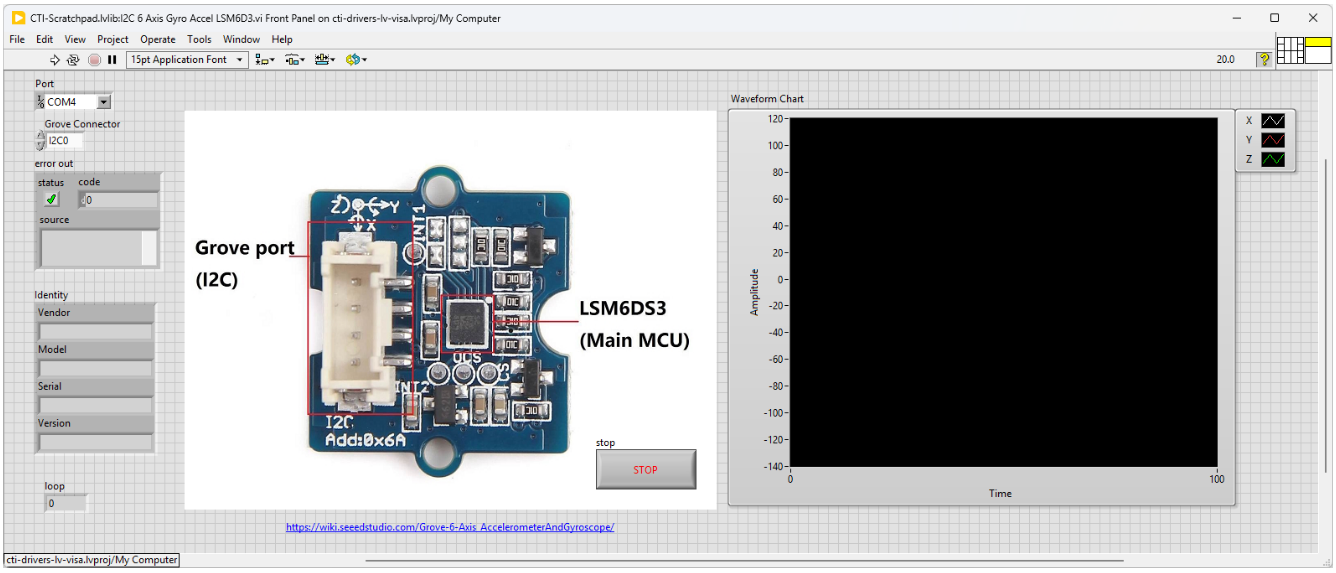
Example Code

Navigate to >>Scratchpad>>Grove>>Gadgets>>I2C 6 Axis Gyro Accel LSM6D3.vi



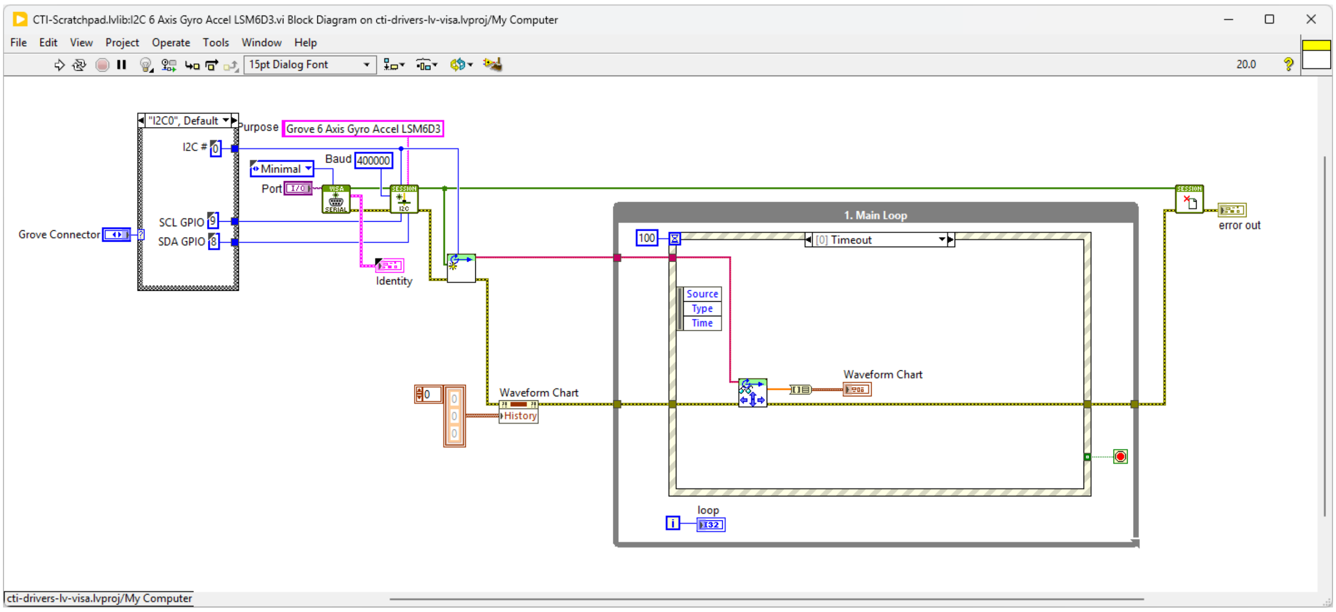
This VI polls the 6 Axis controller and displays the X,Y,Z values on a graph. Select the port for the connected Pico and the Grove connector that the board is plugged into. Press the run arrow.

When you move the board you should see a corresponding movement in the graph



On the block diagram you can see that the selected Grove connector dictates the GPIO Pins used for communication. This session connection is input into **Grove 6 Axis Gyro Accel LSM6DS3.lvclass:Init**. Next we clear the chart and loop round an event structure and use the timeout event to run the **Grove 6 Axis Gyro Accel LSM6DS3.lvclass:ReadPosition.vi** - this VI returns the latest X,Y and Z values and updates the graph.

Pressing Stop will fire the Stop event and exit the loop.



8x8 Thermal Camera

The Grove - Infrared Temperature Sensor Array (AMG8833) is a high precision infrared array sensor which based on advanced MEMS technology. It can support temperature detection of two-dimensional area: 8 × 8 (64 pixels) and maximum 7 meters detection distance.

Hardware Details

Grove - Infrared Temperature Sensor Array(AMG8833)

Demo Video

Here is a video that shows the set-up and running of the Thermal Camera

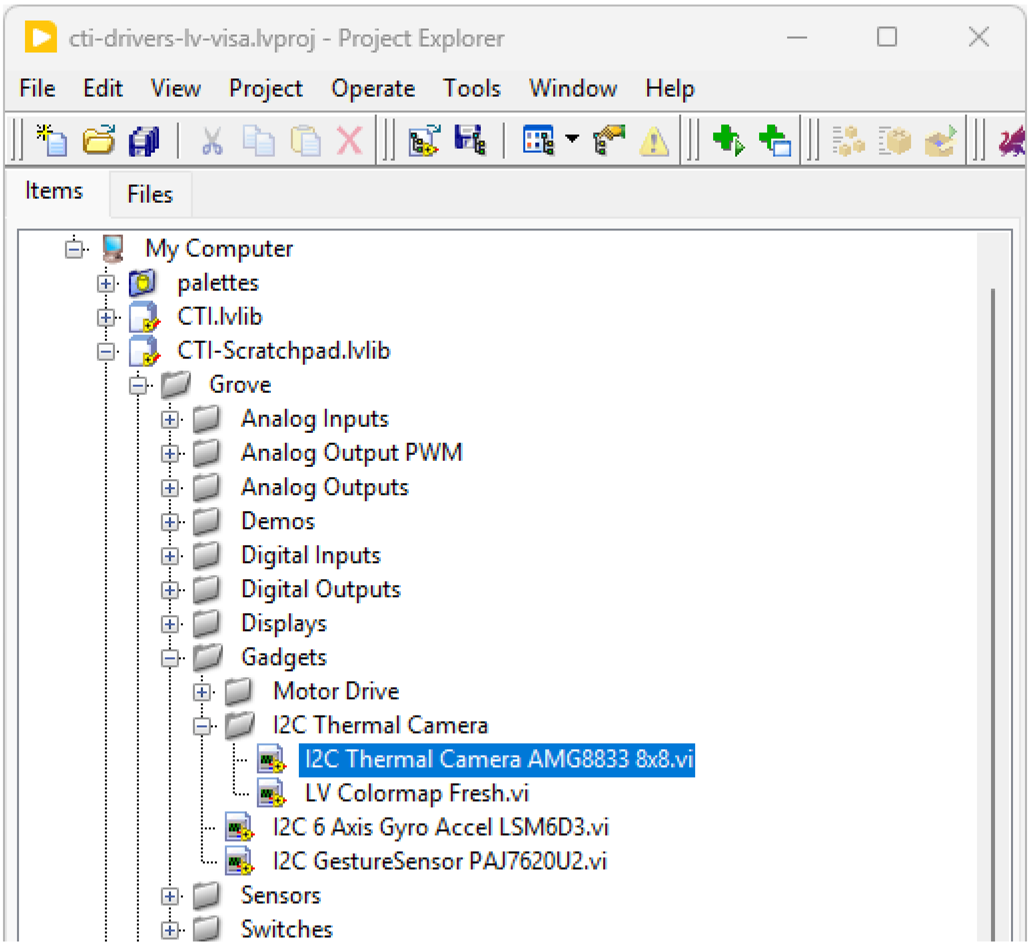
▶ https://www.youtube.com/watch?v=7fWAwL9hBnc *(YouTube video)*

This video shows some of the development process

▶ https://www.youtube.com/watch?v=XtYEoU-dgPk *(YouTube video)*

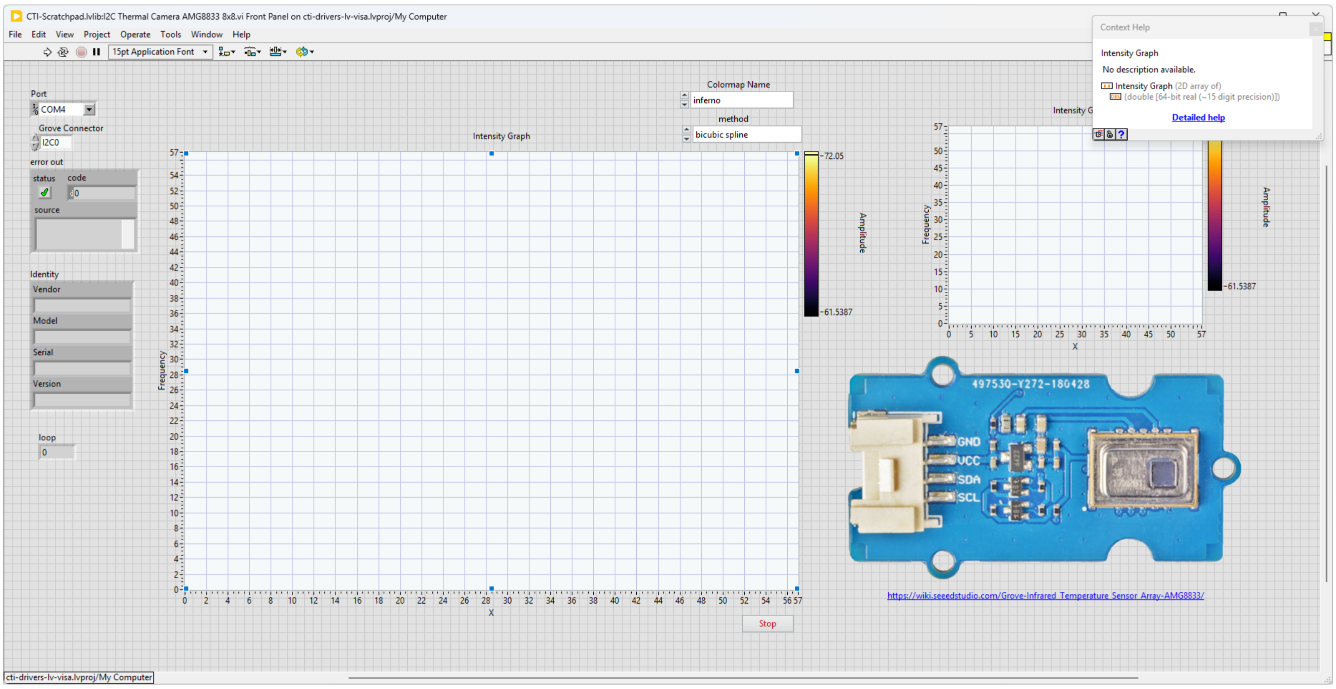
Example Code

Navigate to >>Scratchpad>>Grove>>Gadgets>>I2C Thermal Camera>>I2C Thermal Camera AMG8833 8x8.vi



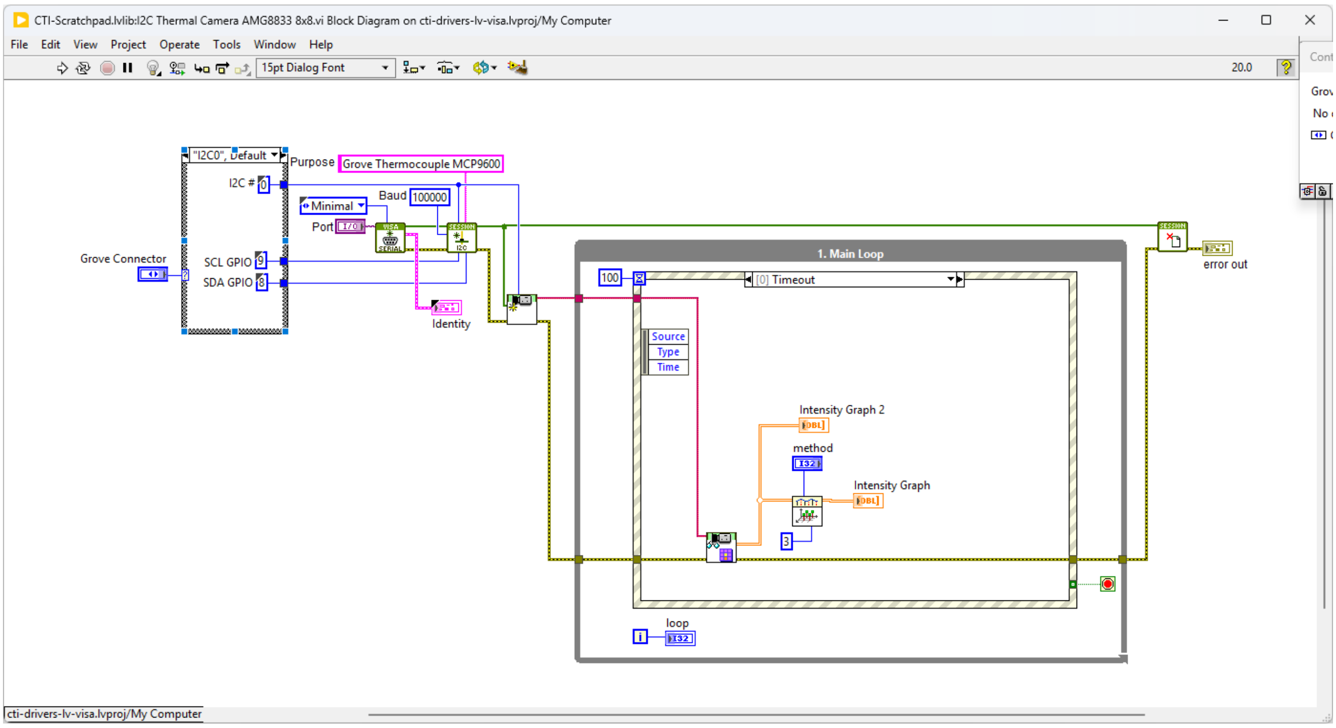
This VI uses the AMG8833 to demonstrate interpolation in LabVIEW. Select the port for the connected Pico and the Grove connector that the board is plugged into. Press the run arrow.

The VI will poll the camera and return the 8x8 thermal signal, this will then be interpolated to a higher resolution signal using magic.



On the block diagram you can see that the selected Grove connector dictates the GPIO Pins used for communication. This session connection is input into **AMG8833 8x8 Camera.lvclass:Init**. Next we clear the chart and loop round an event structure and use the timeout event to run the **AMG8833 8x8 Camera.lvclass:Read Camera.vi** - this VI returns and array of values updates the intensity graph. This 8x8 array is put through a 2D Interpolate function that applies the selected interpolation algorithm (bicubic spline seems to work best) and produces a higher reolution signal.

Pressing Stop will fire the Stop event and exit the loop.



L298 Motor Driver

The Grove - I2C Motor Driver V1.3 (latest version) can directly control Stepper Motor or DC Motor. Its heart is a dual channel H-bridge driver chip L298N that can handle current up to 2A per channel, controlled by an Atmel ATmega8L which handles the I2C communication with platforms such as Arduino. Both motors can be driven simultaneously while set to a different speed and direction. It can power two brushed DC motors or one 4-wire two-phase stepper motor. It requires a 6V to 15V power supply to power the motor and has an onboard 5V voltage regulator which can power the I2C bus and the Arduino(selectable by jumper). All driver lines are protected by diodes from back-EMF.

Hardware Details

Grove - I2C Motor Driver V1.3

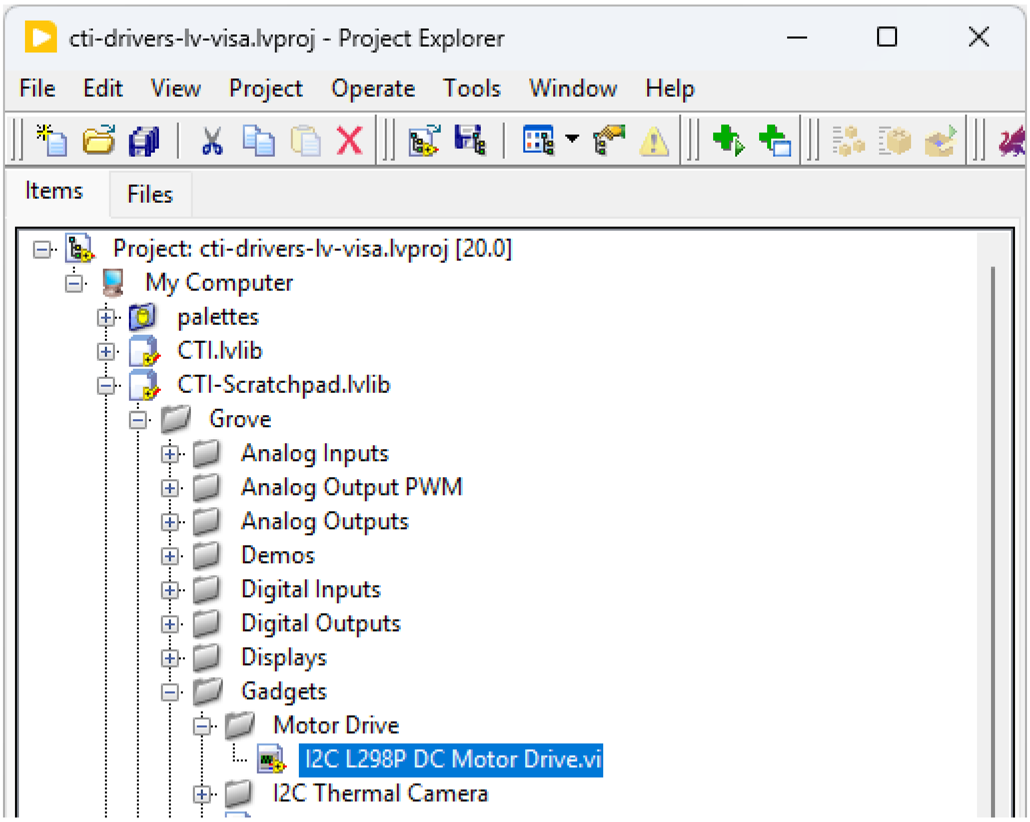
Demo Video DC Motor Drive

This video demonstrates the Grove L298 Motor Drive board used with 2 DC motors.

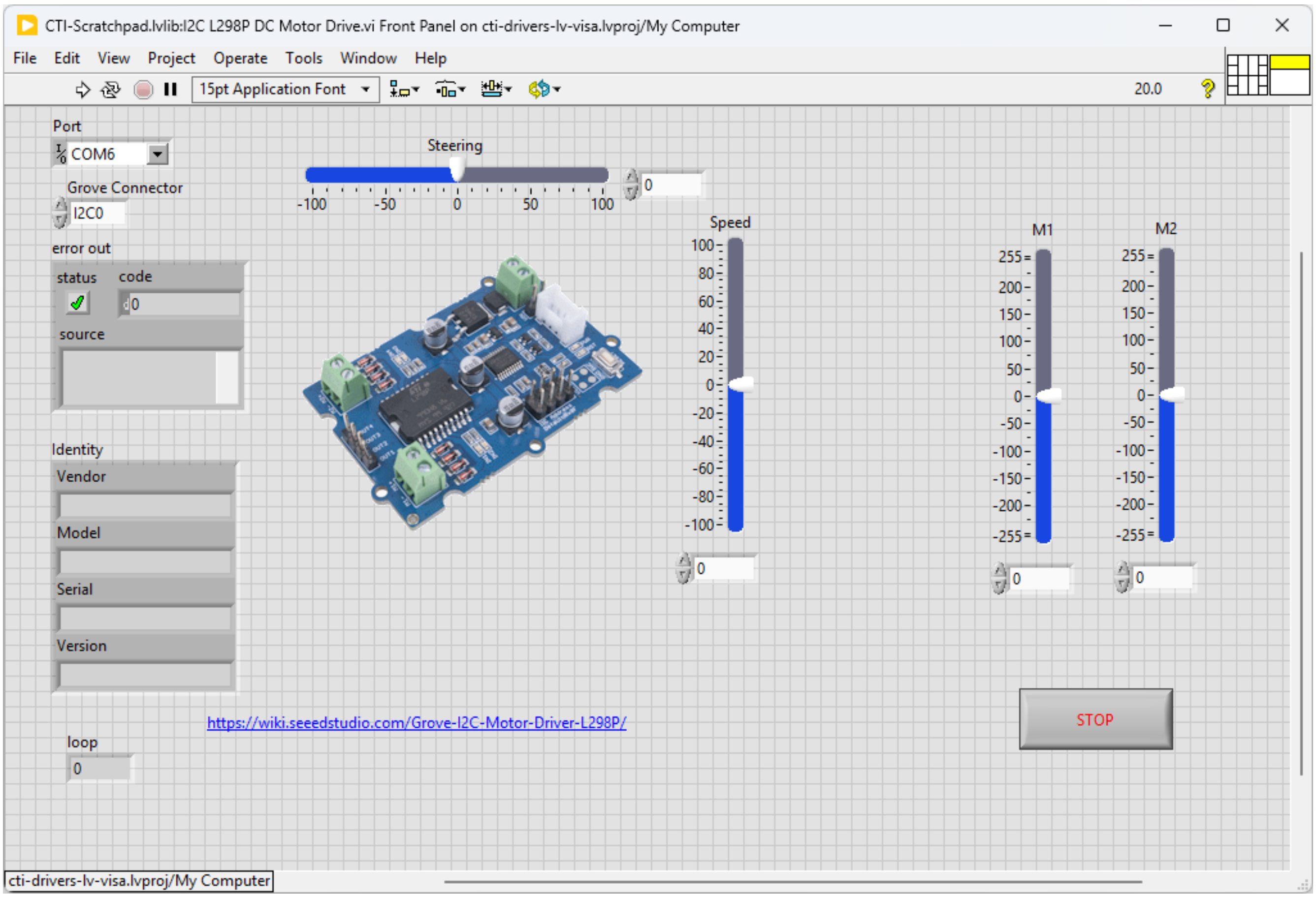
▶ https://www.youtube.com/watch?v=j-9ya1YdSZY *(YouTube video)*

Example Code

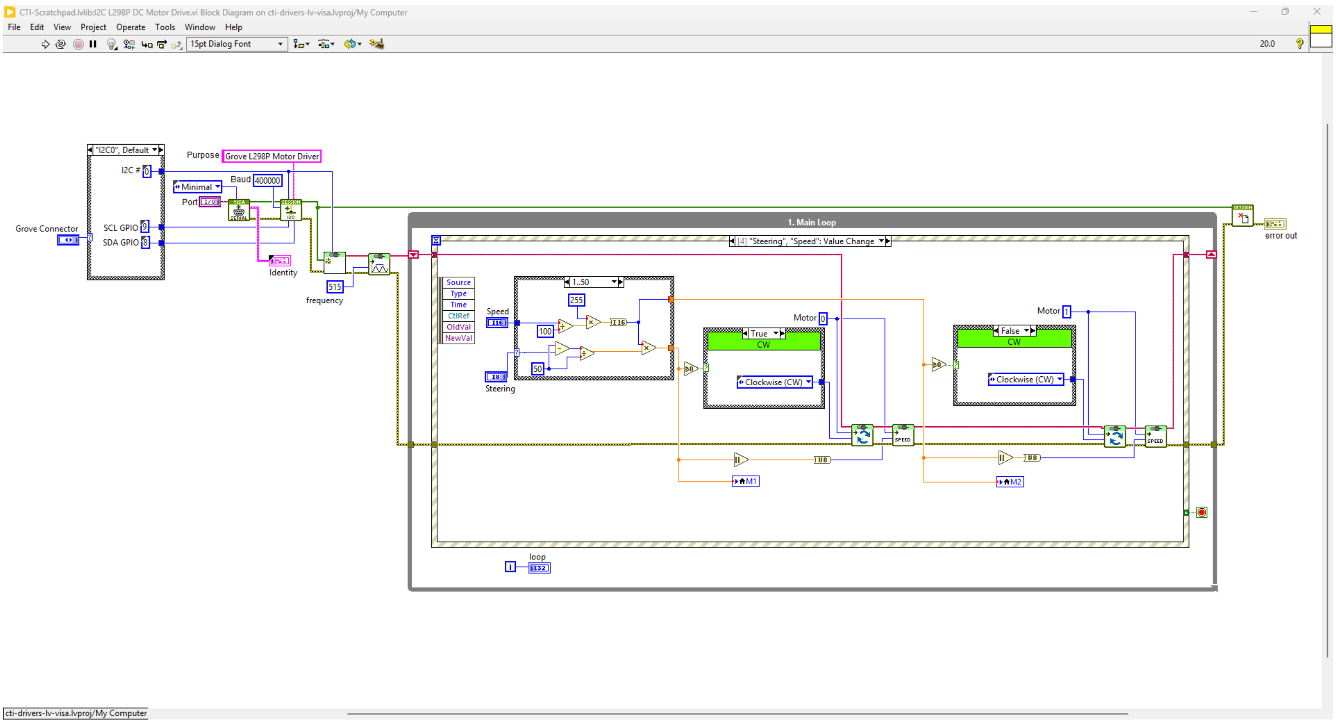
Navigate to >>Scratchpad>>Grove>>Gadgets>>Motor Drive>>**I2C L298P DC Motor Drive.vi**



This VI uses the L298P board to rotate 2 wheels forward and backward. Select the port for the connected Pico and the Grove connector that the board is plugged into. Press the run arrow.



On the block diagram you can see that the selected Grove connector dictates the GPIO Pins used for communication. This session connection is input into **Grove I2C L298P Motor Drive.lvclass:Init**. Next we set the PWM frequency to 515Hz using **Grove I2C L298P Motor Drive.lvclass:Set PWM Frequency.vi**. We the loop round the event structure waiting for Value Change events for the Steering, Speed, M1 and M2 sliders. Pressing Stop will fire the Stop event and exit the loop.



Incremental Rotary Encoder

An incremental encoder is a type of rotary encoder that generates a series of electrical pulses as the encoder shaft is rotated. These pulses are used to measure the rotational position, speed, and direction of the shaft.

This particular board is the Adafruit I2C Stemma QT Rotary Encoder with a SeeSaw protocol microcontroller and NeoPixel LED.

Adafruit seesaw is a near-universal converter framework which allows you to add and extend hardware support to any I2C-capable microcontroller or microcomputer. Instead of getting separate I2C GPIO expanders, ADCs, PWM drivers, etc, seesaw can be configured to give a wide range of capabilities.

Adafruit seesaw an I2C to ... whatever! interface

Hardware Details

Adafruit I2C Stemma QT Rotary Encoder

This board uses the Qwiic connector, so we will need to use a Qwiic to Grove lead.. These can be bought directly or made to the circuit diagram below.

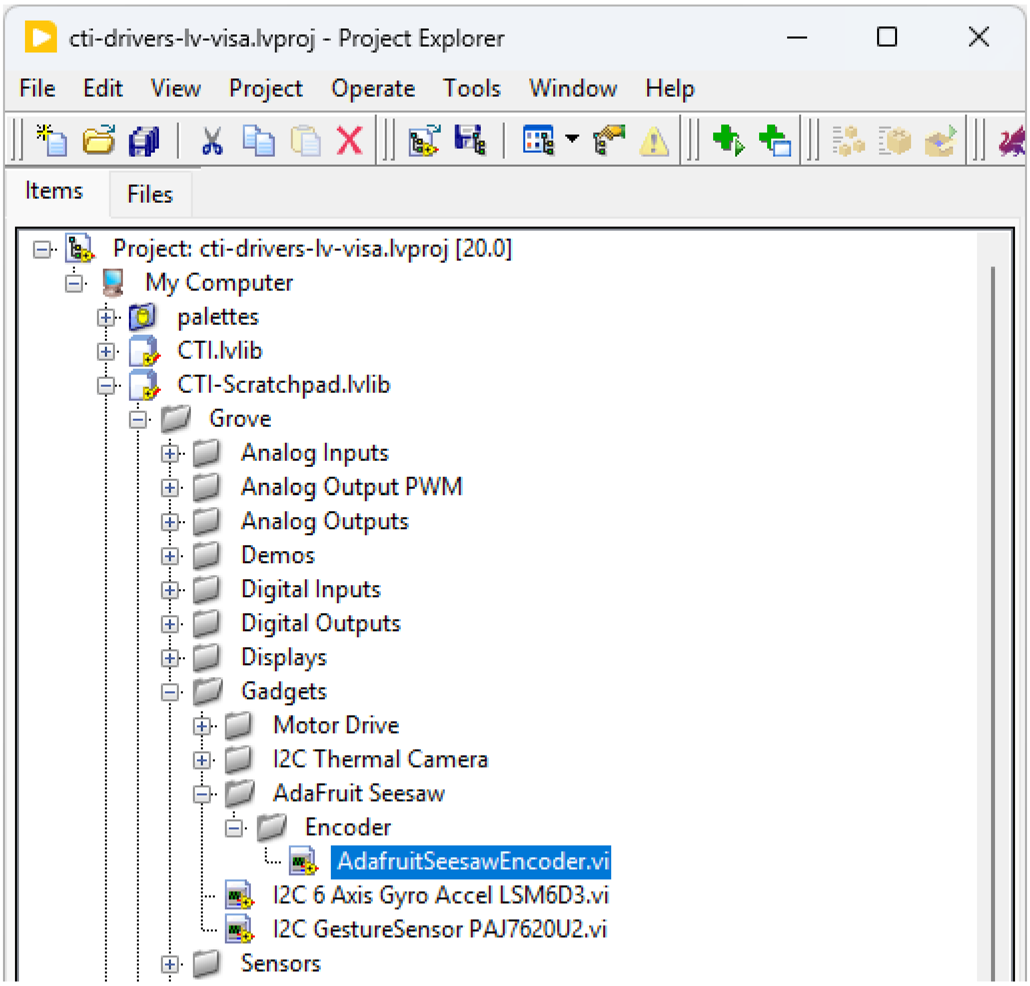
Demo Video

This video demonstrates the Adafruit Rotary Encoder

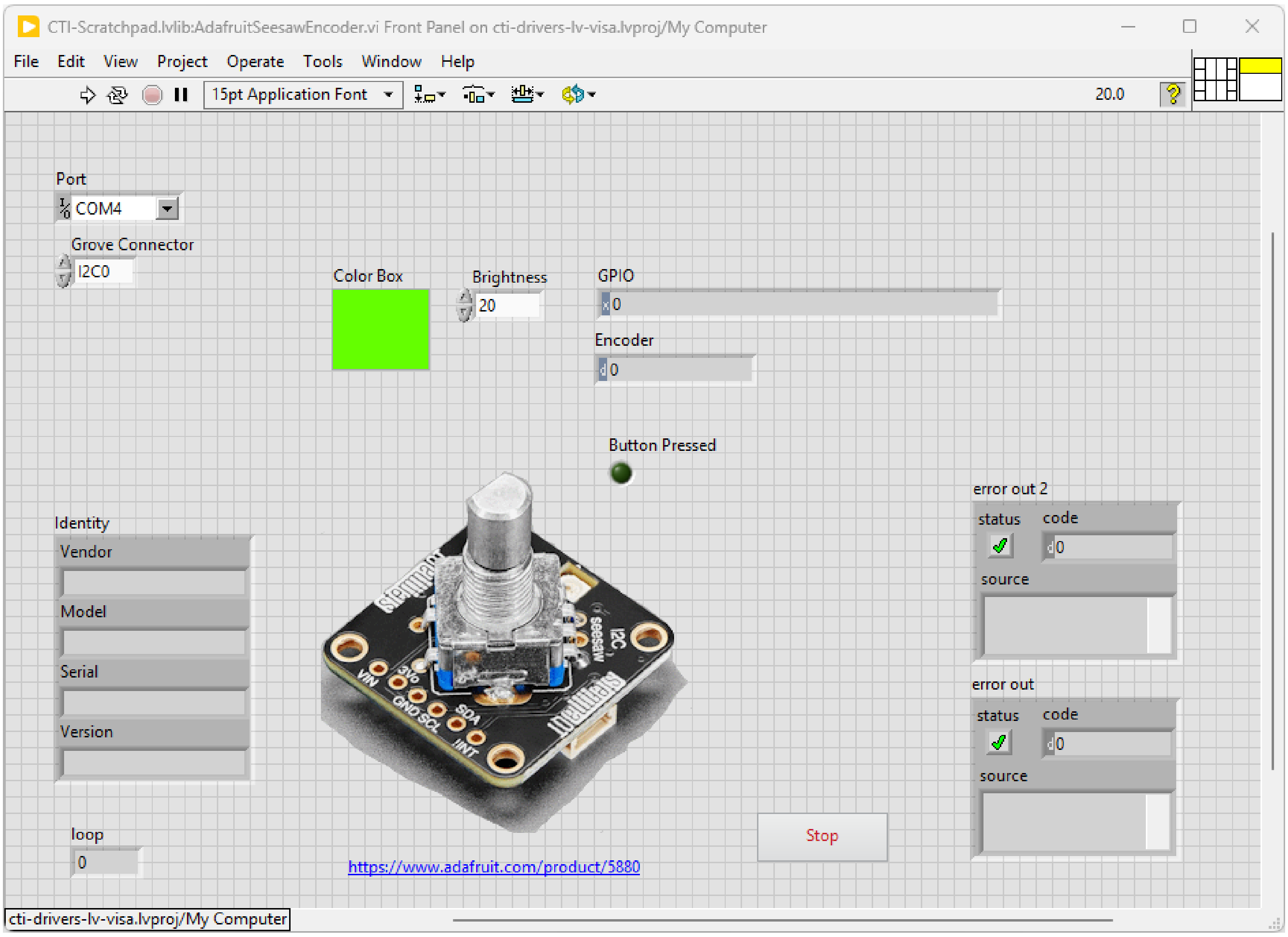
▶ https://www.youtube.com/watch?v=bmwwQtxbOBA *(YouTube video)*

Example Code

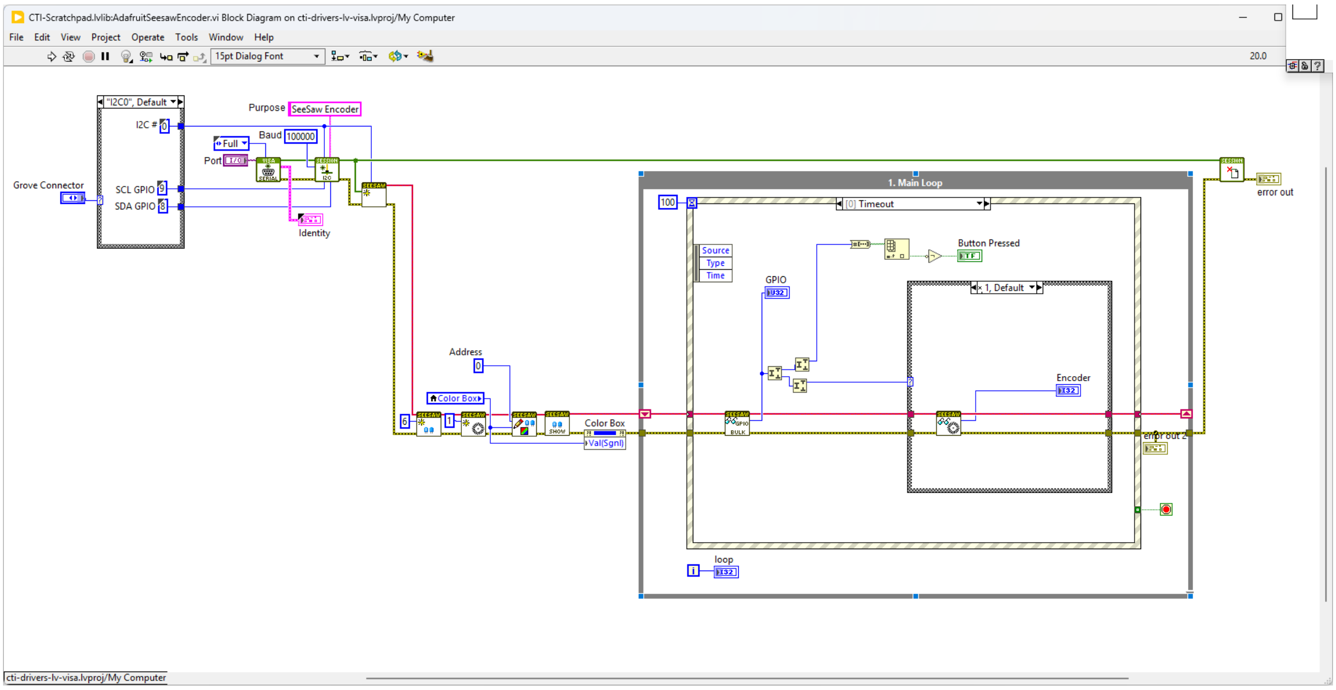
Navigate to >>Scratchpad>>Grove>>Gadgets>>AdaFruit SeeSaw>>Encoder>>**AdafruitSeesawEncoder.vi**



This VI uses the Adafruit Enocder Breakout Board, this board has a small microcontroller running SeeSaw firmware. This gives a lot of functionality through I2C register setting and reading. Select the port for the connected Pico and the Grove connector that the board is plugged into. Press the run arrow.



On the block diagram you can see that the selected Grove connector dictates the GPIO Pins used for communication. This session connection is input into **AdaFruit-Seesaw.lvclass:Init**. Next we set up GPIOs, Encoder and NeoPixel LED using **AdaFruit-Seesaw.lvclass:Init:NeoPixelInit.vi**,**AdaFruit-Seesaw.lvclass:Init:EncoderInit.vi**. We then initialise the NeoPixel Buffer by setting **AdaFruit-Seesaw.lvclass:Init:SetNeoPixelBuf.vi**,**AdaFruit-Seesaw.lvclass:Init:NeoPixelShow.vi**. We the loop round the event structure in the timeout checking the GPIO pins for the encoder interupt. If the encoder has changed it will read it. Changing the brightness or Color Box will change the Neopixel LED. If you press the rotary encoder a GPIO pin will change, this is caught in the timeout loop. Pressing Stop will fire the Stop event and exit the loop.


CTI-Scratchpad.lvlib:AdafruitSeesawEncoder.vi Block Diagram on cti-drivers-lv-visa.lvproj/My Computer
File Edit View Project Operate Tools Window Help
15pt Dialog Font
Purpose
SeeSaw Encoder
"I2C0", Default
I2C #
Baud
100000
Port
Grove Connector
SCL GPIO
SDA GPIO
Identity
Address
Color Box
1. Main Loop
100
[0] Timeout
Source
Type
Time
GPIO
Button Pressed
1, Default
Encoder
error out
error out 2
loop
cti-drivers-lv-visa.lvproj/My Computer