

Micro CV Controller Manual

Introduction.....	1
Control Voltage to OSC conversion.	2
Configuration	3
Scalable Voltage Input.....	4
Scaling Analogue Input.....	5
Digital Switch Input.....	6
Control Voltage Outputs.....	6
Pulse Width Modulation	6
PWM Out.....	8
Connectors	8
Control Voltage Input	9
Control Voltage Output.....	9
Device Configuration.....	11
Configuring host computer Bluetooth OSC device driver.....	11
Configuring Mac OSX.....	11
Serial Port.....	11
Configuring for more than one Bluetooth device	12
Configuring Device Driver Settings	13
Network IP Address.....	13
UDP Port.....	13
OSC Address Space	13
UDP In	13
Configuration Ports.....	13

Introduction

The Micro CV Controller is a device used for converting control voltages to definable OSC messages and vice versa. The device has sixteen input/output ports: eight 10-bit 0 to +5 volt scalable control voltage input; and eight ports that can be configured as digital inputs or control voltage outputs.

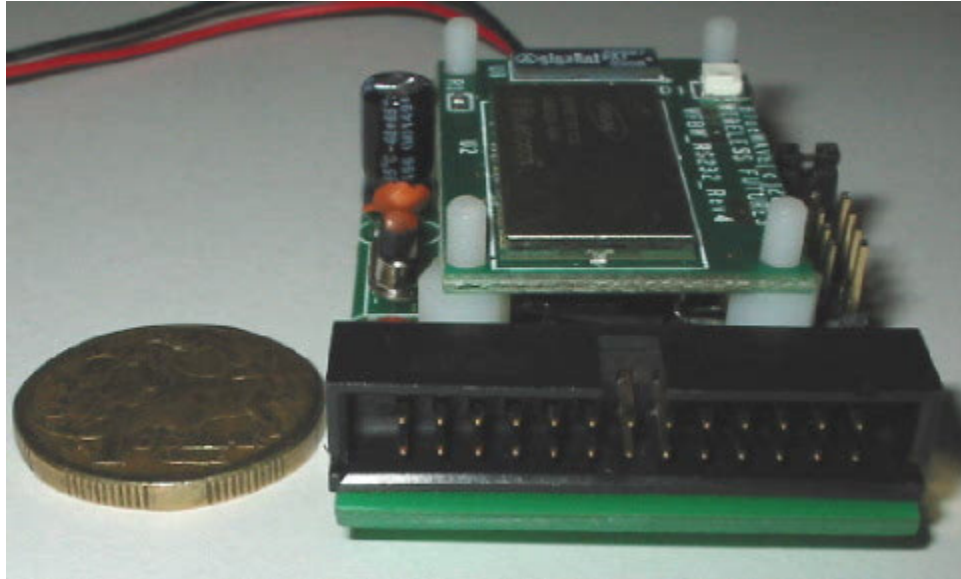


Figure 1 Micro CV Controller

When a port is configured as an input, a change in the input voltage will cause the device to generate an OSC messages to the host Bluetooth enabled computer. When a port is configured as an output, sending an OSC message through the host computer (if the controller is configured to respond to that message), the controller will alter the output voltage depending upon the message. This makes it possible to remote control up to eight analogue or digital devices—such as relays or robotic motors—using OSC. The Micro CV controller is powered by a 9V DC battery or 9V power supply.

Control Voltage to OSC conversion.

The control voltage to OSC conversion is separated into two sections: scalable voltage and digital switch, of which there is eight of each. Warning: Voltage inputs greater than 5V may cause damage to the unit.

Configuration

By default, the unit is configured for all I/O to be inputs. Any input that is not connected to a sensor will have a spurious voltage on it, known as a floating voltage. This may cause the device to generate more messages than can be handled by the computer receiving the Bluetooth data. This can be overcome by configuring the device to muting the I/O by selecting the “M” check box for each channel to be muted, and clicking the “Write Config Button.”

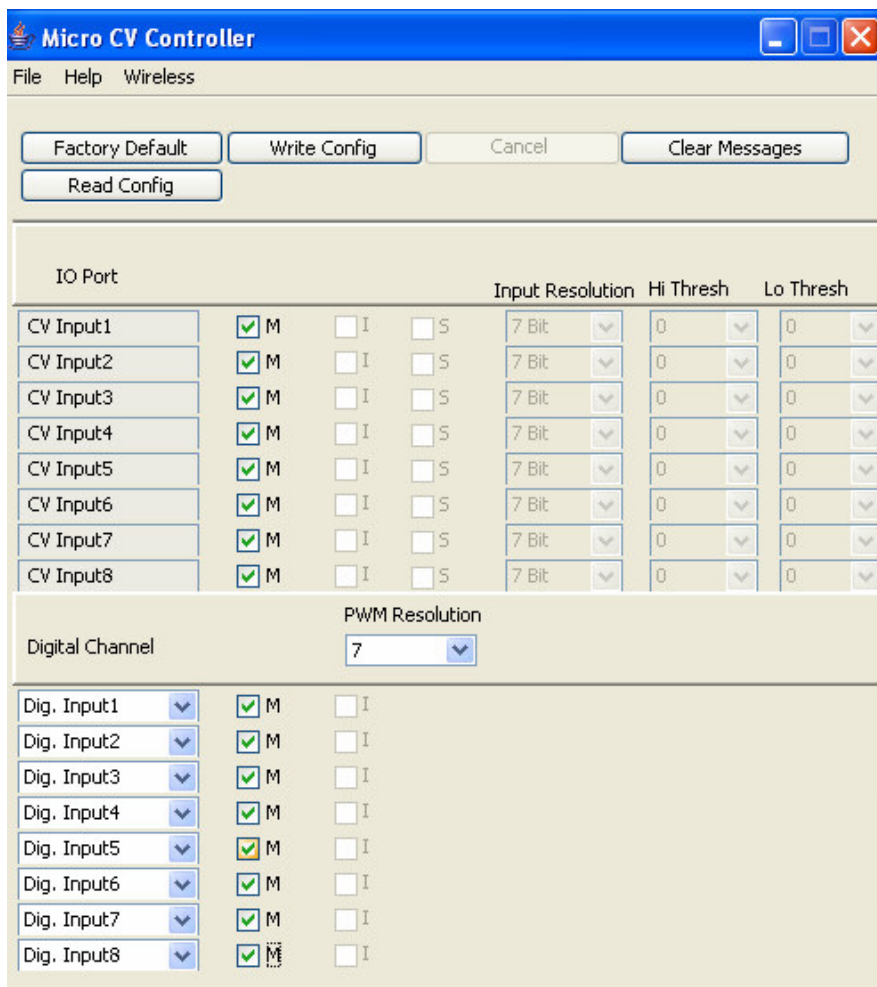


Figure 2 Configuring Device to Mute I/O

If too much OSC data is being transmitted, making it difficult to change the configuration, the device can be made to momentarily stop generating Bluetooth data by shorting the two pins closest to the IO Plug.

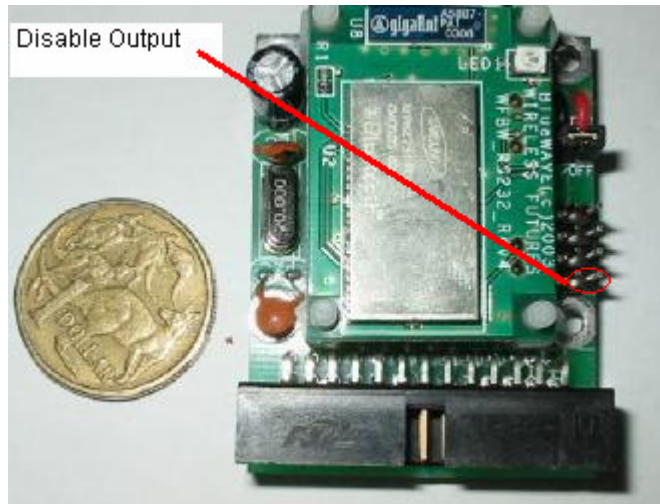


Figure 3 Momentarily Disabling Bluetooth Data

Scalable Voltage Input

Each scalable voltage input operates from a voltage ranging from 0 to +5V and generates OSC messages as configured. For example, consider the following configuration.

IO Port		Input Resolution	Hi Thresh	Lo Thresh
CV Input1	<input checked="" type="checkbox"/> M <input type="checkbox"/> I <input type="checkbox"/> S	7 Bit	0	0

Figure 4 Control Voltage to OSC Configuration

Control voltage input one is configured to generate seven bit OSC messages. Increasing the voltage from zero to five volts on voltage input 1 would cause the controller to generate OSC messages `'/<OSC ADDRESS>/Ana1'` with a value ranging from 0 to 127. The value of the data 2 byte in the message is proportional to the voltage at the input.

```
# Change voltage at input from 0 to +5V
/minicv/Ana1 0
.....
/minicv/Ana1 71
.....
/minicv/Ana1 127
```

Alternatively, consider the following configuration

IO Port				Input Resolution	Hi Thresh	Lo Thresh
CV Input1	<input checked="" type="checkbox"/> M	<input type="checkbox"/> I	<input type="checkbox"/> S	7 Bit	0	0
CV Input2	<input type="checkbox"/> M	<input checked="" type="checkbox"/> I	<input type="checkbox"/> S	10 Bit	0	0

Figure 5 Inverting and Configuring ten Bit

Control voltage input 2 is configured to generate 10 bit OSC messages. Additionally, the channel is configured to invert the value and to vary data byte 1. Increasing the voltage from 0 to +5V on control voltage input 2 would cause the controller to generate OSC messages ‘/OSC ADDRESS>/Ana2’ with the value ranging from 0 to 1023, with the value inversely proportional to the voltage at the input.

```
# Change voltage from 0 to + 5 VDC
/minicv/Ana2 1023
.....
/minicv/Ana1 71
.....
/minicv/Ana1 0
```

Scaling Analogue Input

Another method of using 10-bit resolution is through the use of the scaling mechanism. This can be particularly useful when the user is only interested in values that range between certain voltages. By using the scaling function, the device can be set to output values only between certain values, outputting an OSC value ranging from 0 to 127. In this instance, the device can be set to monitor between 30 and 90. Given that the scaled value only has seven bit resolution, the new resolution for this sensor would be approximately 18mV, which is twice what it was without scaling on seven bit resolution. The sensor would be configured as shown in Figure 6.

IO Port				Input Resolution	Hi Thresh	Lo Thresh
CV Input1	<input type="checkbox"/> M	<input checked="" type="checkbox"/> S		7 Bit	90	30

Figure 6 Scaling CV Input

Moving the sensor from minimum to maximum will produce an OSC messages ‘/OSC ADDRESS>/Ana1’ with a value ranging from 0 to 127.

Digital Switch Input

Warning: Voltages greater than 5V on the Digital inputs may cause damage to the unit.

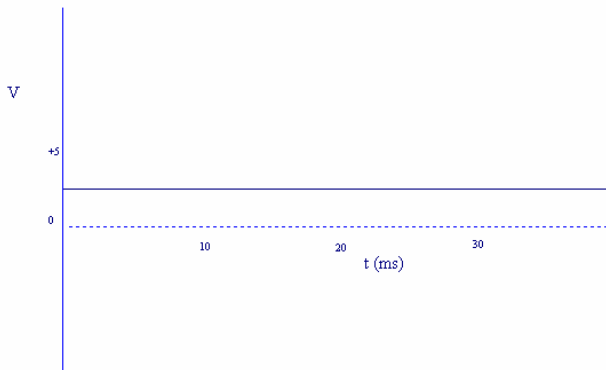
The digital switch input generates an OSC message when the input is shorted to ground or the +5V rail. Shorting Digital Input 1 to ground will cause the device to generate OSC messages ‘/OSC ADDRESS/Dig1 1’, while removing the short will produce ‘/OSC ADDRESS/Dig1 0’. The digital channels can also be configured to invert the value.

Control Voltage Outputs

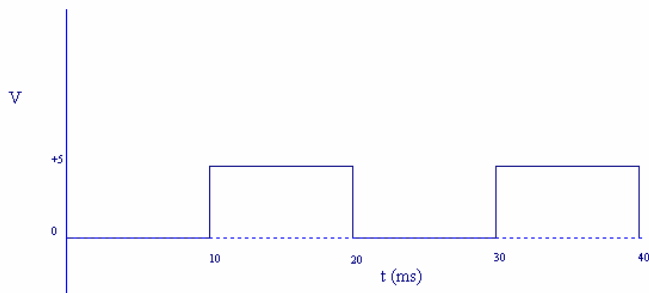
The device can generate control voltage through the use of pulse width modulation.

Pulse Width Modulation

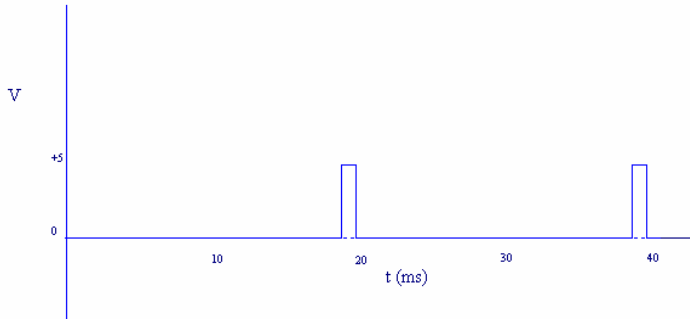
Pulse width modulation, known as PWM, is different to DC voltage in that DC voltage is a steady electrical force at any instance of time. The graph below shows 2.5V DC as a steady voltage over time.



PWM, on the other hand, is the average DC potential over a fixed period of time. For example, if during a 20ms space of time, a +5V pulse existed for 10ms, the average voltage would be +2.5V. If we repeat this every 20ms, the output will transmit on average 2.5V.



If we reduced the pulse width to 1ms, this would mean that a 1 ms pulse would appear every 20ms, which means that the average voltage would be $1/20^{\text{th}}$ of 5V, which is 0.25V.



The amount of time that the signal is positive with respect to the repeat time is called the duty cycle. This is the whole concept of PWM.

The apparent smoothness of the voltage is dependant upon how often the pulse is refreshed. For example, a pulse width of 1ms every 2ms, will appear significantly smoother than a 10s pulse every 20s. The number of times the pulses are repeated is known as the frequency—higher frequency means smoother output.

Another factor is resolution, which means how many different pulse widths can be set.

For example, a 7 bit resolution means that you can have 127 different PWM values.



The resolution is inversely proportional to the frequency; i.e. a high resolution normally means a lower frequency. The frequency available for the following resolutions is listed below.

Bits	Duration ms	Frequency
8	15	67
7	7	143
6	10	100
5	2.5	400
4	2	500
3	1	1000
2	0.8	1250
1	DC	0

Figure 7 PWM resolution versus frequency

The Micro CV controller uses PWM to generate CV output as opposed to the Dumb Controller and Smart Controllers, which use a steady DC voltage. Although PWM is better for controlling motors, steady DC voltage is better for controlling analogue synthesisers in that PWM introduces audible noise

PWM Out

The control voltage output generates a PWM voltage of 0 and +5 V, with the duty cycle depending upon the incoming OSC message and the configuration. Consider the following configuration: scalable output 1 is configured to generate a voltage when the controller receives ‘/<OSC ADDRESS>/AnaOut1’ message

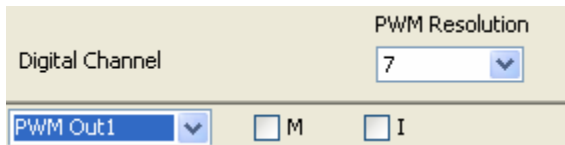


Figure 8 PWM Outputs

If a series of ‘/<OSC ADDRESS>/AnaOut1’ messages with values ranging from 0 to 127 are received, the device will generate a PWM output with the duty cycle ranging from 0 to 100%. The duty cycle is directly proportional to the value.

By selecting the invert checkbox, the output can be configured to also generate a voltage that is indirectly proportional to the OSC message value.

Connectors

The connectors on the Micro CV Controller are a 26 Pin IDE connector, which enables the performer to use a single plug connection during a performance setup to connect all the sensor inputs; and a 9V Battery clip. The pin-out of the I/O plug is

Pins	Description	Comment
1 to 4	+5 V	+ 5V supply for sensors
5 to 12	Analogue inputs 1 to 8	Sensor inputs for CV to OSC conversion
13 to 20	Digital Input / PWM outputs 1 to 8	Configurable as Digital input of PWM output
21 to 26	0V	Ground supply for sensor inputs and PWM outputs

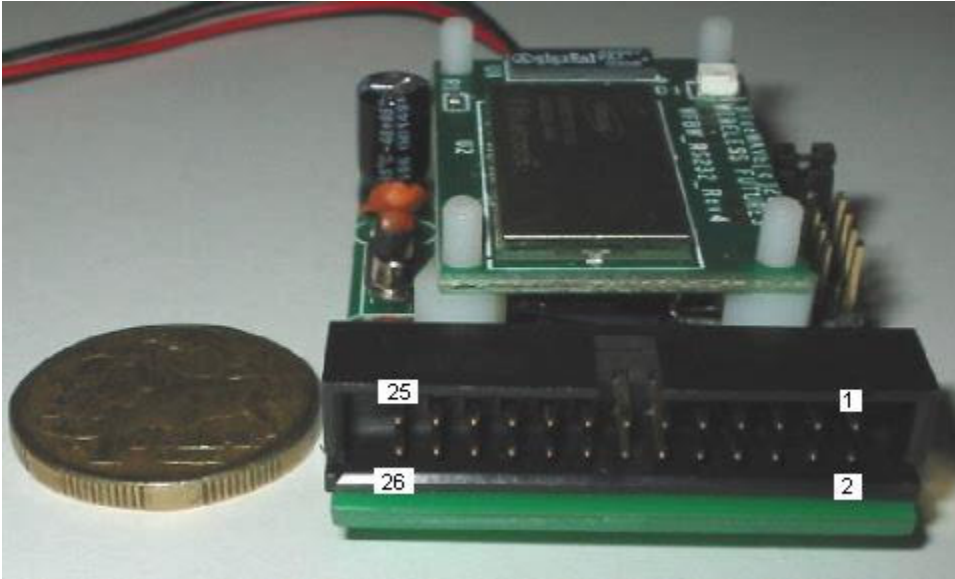


Figure 9 Connectors

Control Voltage Input

A simple sensor can be created by connecting the outer pins of a potentiometer to pins 1 and 26 of the IO Socket and the wiper pin to the sensor input.

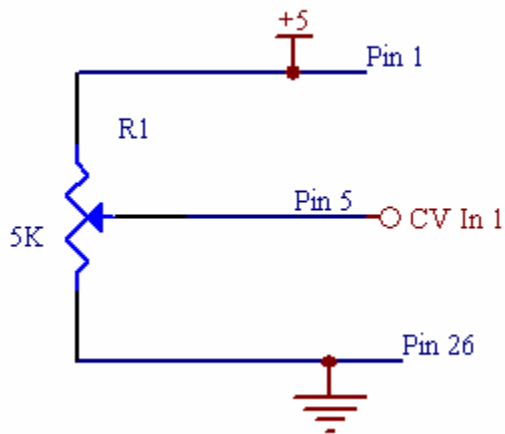


Figure 10 Simple Potentiometer Sensor

Control Voltage Output

The control voltage outputs are used to provide WM control voltages

The outputs can supply approximately 2mA per channel, and as such, must be used with a driving transistor. When driving inductive loads, such as motors, solenoids, and relays, a commutating diode must be used to prevent inductive spikes from damaging the unit. Controllers damaged from inductive spikes are considered to be abused and are not eligible for warranty repair.

The uses of a commutating diode with a motor is shown in Figure 11. Note that the diode is configured in reverse bias.

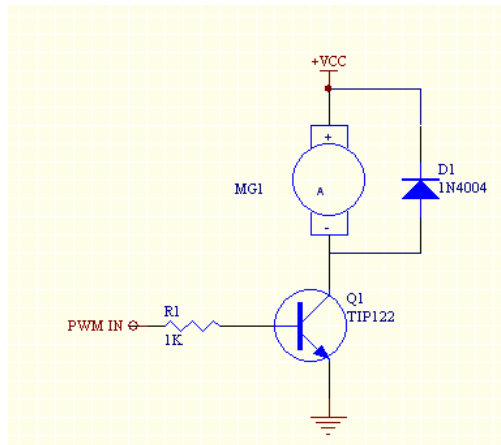


Figure 11 PWM driving a motor

Device Configuration

The Micro CV controller comes with the following factory default configuration:

- Scalable CV inputs 1 to 8 – 10 Bit resolution, value range 0 to 127 proportional to input control voltage.
- Digital switch inputs 1 to 8, values 0 and 1 for switch open and closed respectively.

CV output duty cycle ranges from 0 100% with controller value 0 to 127 respectively.

The following configuration options (default off) are available for each individual channel:

- Mute – cause the input or output to have no effect.
- Invert – reverses the effect of the controlling data byte.
- Scale Input—Scales the CV input to produce OSC messages only between certain ranges. Available only for inputs.

Configuring host computer Bluetooth OSC device driver

The Micro CV controller uses a host computer running the serial Bluetooth protocol to convert serial Bluetooth data from the device to OSC messages. The host computer must be configured to connect to the device.

Configuring Mac OSX

Serial Port

For each device to be set up on your computer, each device must be configured as a Bluetooth Serial Port

To set up the Bluetooth serial port, please take the following steps:

1. Make sure your Micro CV is turned on.

2. Open the "Bluetooth Serial Utility", located in the "Applications/Utilities" folder on your hard-disk.
3. The "Bluetooth Serial Ports" page will be displayed. Click "New".
4. Type the port name "minicv", making sure that it is all in lower case.
5. Select the "Outgoing" port direction.
6. Click "Select Device". The "Select Bluetooth Service" page is displayed. If you do not see the device, click the "refresh" button.
7. Select "MiniCv" and then the "Bluewave" service.
8. Click "Select" to close the "Select Bluetooth Service" page.
9. Select "RS-232" from the "Port Type" combo box.
10. Click "OK".
11. The serial port is set up and ready to use.
12. Close the "Bluetooth Serial Utility".

Configuring for more than one Bluetooth device

If you are configuring more than one device, you will need to create a different name for step 4 of the previous step; for example, 'minicv2' instead of 'minicv'.

If you have more than one Micro or Mini CV controller using Bluetooth, you will need to make a copy of the configuration file, and a copy of the command file. I will assume that you have named the second Bluetooth Serial Port 'minicv2' in step 4.

1. Copy 'minicv.ini' and name it 'minicv2.ini'
2. Edit 'minicv2.ini' and change the 'serial' directive to reflect the new serial port name; for example:

```
serial /dev/tty.minicv2
```
3. Copy the file 'MiniCV.command' and rename the copy to 'MiniCV2.command'.
4. Edit 'MiniCV2.command' and change the configuration file name to 'minicv2.ini'; for example:

```
./MiniCV minicv2.ini
```
5. Modify the OSC address space to reflect the new name as detailed in the next section.

Configuring Device Driver Settings

The device, OSC address space and UDP settings for the device are configured in the initialisation file 'minicv.ini'. The device driver is started using the 'MiniCV' program followed by the name of the configuration file. This has been simplified through a shell script, 'MiniCV.command', which contains the appropriate parameters to the application.

Network IP Address

The Network IP address is set by the 'srv' directive in the initialisation file. The following settings sends OSC data to the local host.

```
srv localhost
```

The following sends the OSC data to ip address 192.168.0.1

```
srv 192.168.0.1
```

UDP Port

The UDP port to send the OSC data is set by the 'udp' directive. The following directive sets the UDP port to 7000.

```
udp 7000
```

OSC Address Space

The OSC address space is set by the 'osc' directive. Consider the following directive:

```
osc /minicv
```

CV input on analogue input 1 will be sent as OSC message '/minicv/Ana1 <CVvalue>'.

UDP In

The UDP port for receiving OSC messages to convert to PWM is set with the directive 'udpin'. For example

```
udpin 7700
```

Will listen for OSC messages on UDP port 7700, causing the device to generate a PWM value if the OSC address and device configuration match.

Configuration Ports

The UDP ports used for configuration are defined by directives 'configin and 'configout'.

```
configin 1113  
configout 8000
```

These ports must be free on the computer running the patch editor.