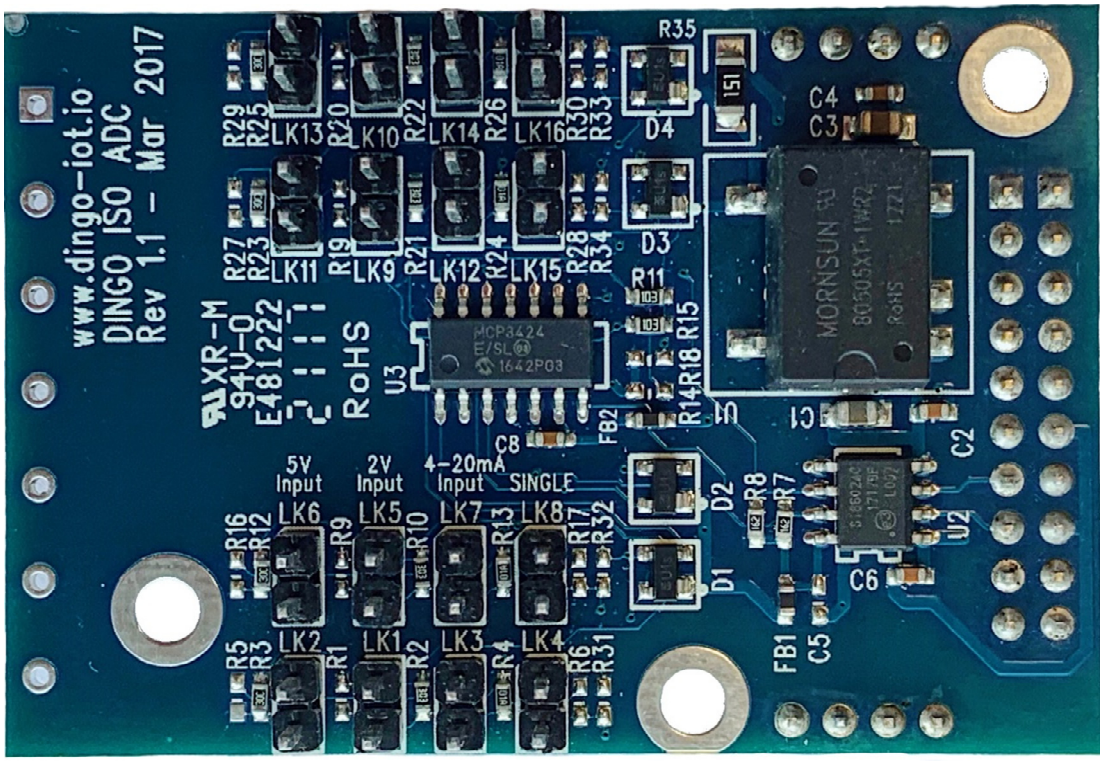
DINGO PLUG-IN

Opto Isolated 4 Channel 18bit ADC Plug-in

DATASHEET COMPLETE

Item specifications

Go-IoT Item Id:	Option	DINGO-PG-AI4-01
ADC Convertor		MCP3424-E/SL – 18bit
ADC Inputs	A B C D	0V to +2V , -2V to +2V 0V to +5V , -5V to +5V 0V to +10V , -10V to +10V 0mA to 20mA
Sample Rate		3,75 to 240 Samples per Second
I2C Channel 1 Interface		0xD0
Isolation Voltage		2.5kV RMS
Drivers		Linux, DINgo Stack
Expansion Connectors		1 x 20way header from Base Board 2 x 4way headers from Base Board
DC Input		+5V
Temperature		-20degree C to +85degree C
Size (L x W)		60 x 40 mm
Country/Region of Manufacture:		EU



20WAY ADC Interface

Pin	Port	Dir	Pull Up	Function	Description
1	+12V			POWER	
2	SPI_CLK	IN		SPI	SPI Clock
3	+3.3V			POWER	
4	SPI_MOSI	IN		SPI	SPI Master Out SLAVE In
5	TXD2	IN		Serial TX Data	Serial TTL Data from Host – Channel 2
6	SPI_MISO	IN		SPI	SPI Master In SLAVE Out
7	RXD2	OUT		Serial RX Data	Serial TTL Data to Host – Channel 2
8	SPI_SSx	OUT		SPI	Output from Power Line Module
9	NEVENTx	OUT		Power Line	SPI Slave Select
10	TXD3	IN		Serial TX Data	Serial TTL Data from Host – Channel 3
11	GND			POWER	
12	RXD3	OUT		Serial TX Data	Serial TTL Data to Host – Channel 3
13	ADDR1			IO	Module Specific
14	I2C_SCL	IN		I2C CLOCK	I2C – Channel 1 Clock
15	ADDR1			IO	Module Specific
16	I2C_SDA	BI		I2C DATA	I2C – Channel 1 Data
17	GPIOx	BI		IO	Module Specific
18	USB +	BI		USB Data	USB Positive Channel x
19	+5.0V	IN		POWER	+5.0V Output – 1000mA available
20	USB -	BI		USB Data	USB Negative Channel x

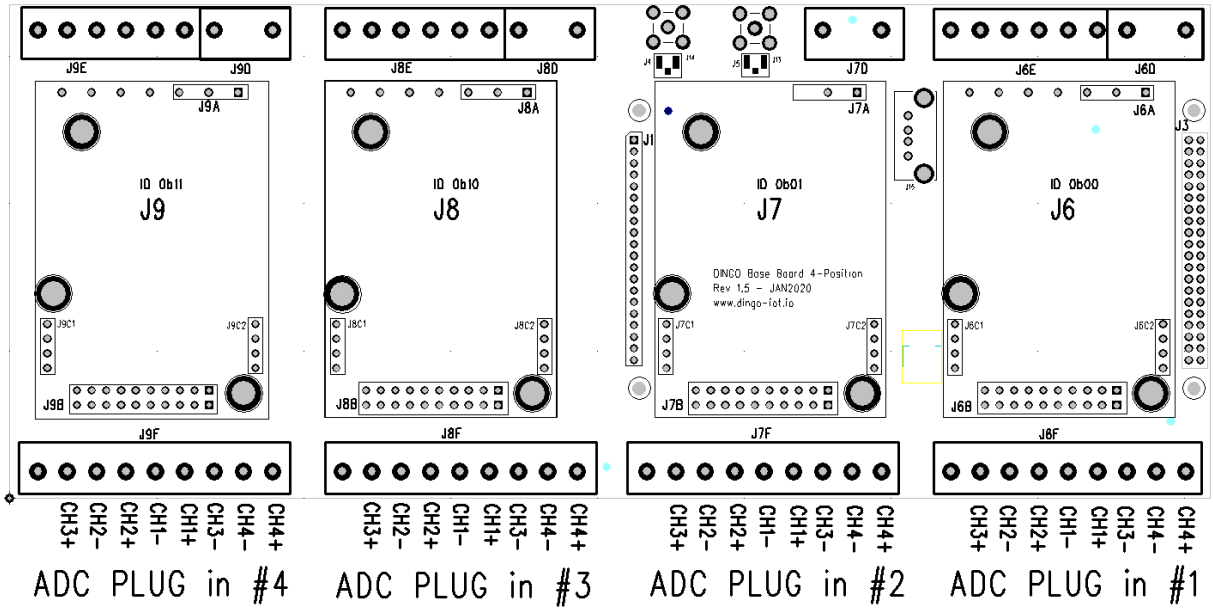
x = Channel / Number depend on location on Base Board

Blue Text is signals used on Module

8WAY ADC Interface to External Connectors

21	Channel 1+	IN		Analog Channel 1 + Input	
22	Channel 1-	IN		Analog Channel 1 - Input	
23	Channel 2+	IN		Analog Channel 2 + Input	
24	Channel 2-	IN		Analog Channel 2 - Input	
25	Channel 3+	IN		Analog Channel 3+ Input	
26	Channel 3-	IN		Analog Channel 3- Input	
27	Channel 4+	IN		Analog Channel 4 + Input	
28	Channel 4-	IN		Analog Channel 4 - Input	

ADC Plug In Connector Positions



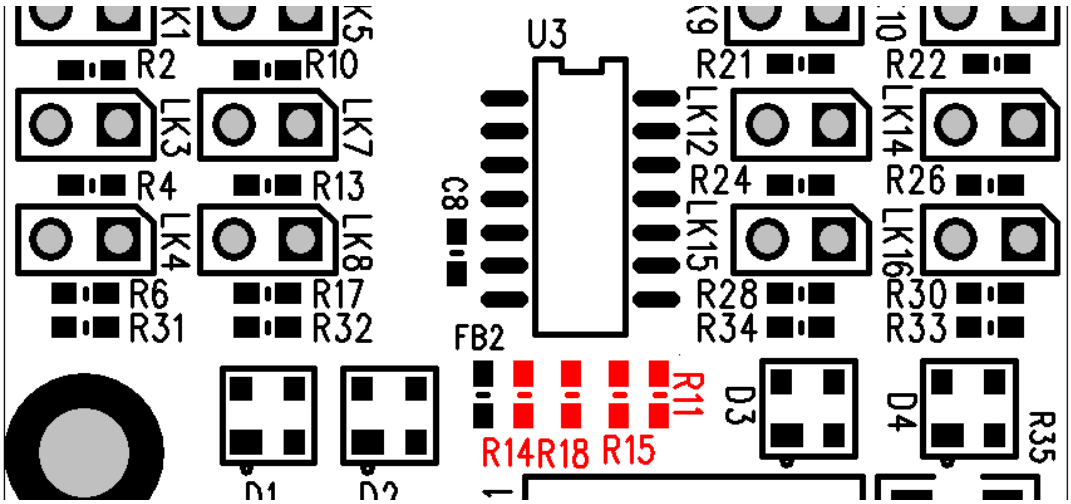
MCP3424 I2C Address (Factory Default)

```
pi@raspberrypi:~$ i2cdetect -y 0
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- 6e --
70: -- -- -- -- -- -- -- --
```

MCP3424 I2C Address Links

R11, R14, R15 and R18 set the address of the I2C Interface

The Factory Default is
R11 , R15 – 10K fitted
R14 , R18 – Not Fitted

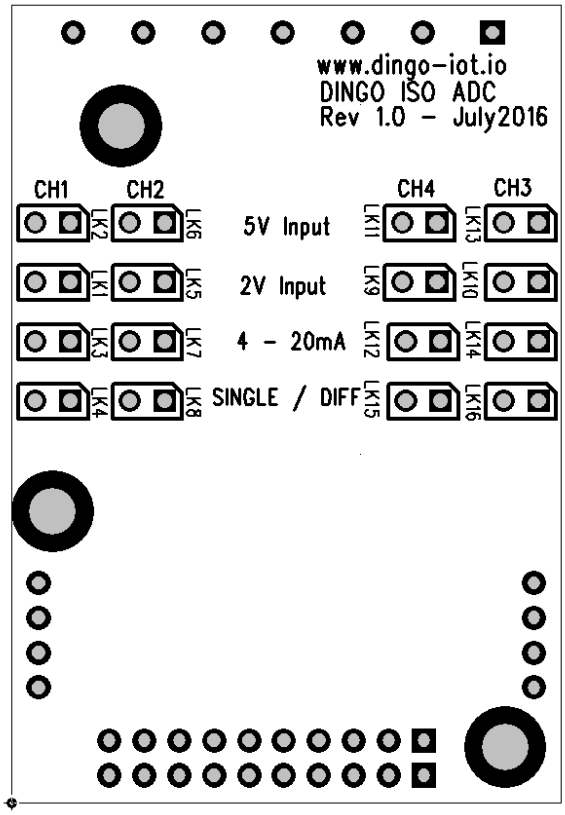


R15	R18		R11	R14	I2C Address
ADD0	ADD0		ADD1	ADD1	ADC Address Pins
10K	NF		10K	NF	0x6E - Factory Default
10K	0R		10K	0R	0x68 *** NOT ALLOWED ***
10K	0R		NF	NF	0x69
10K	0R		10K	NF	0x6A
10K	NF		10K	0R	0x6C
10K	NF		NF	NF	0x6D
NF	NF		10K	0R	0x6C
NF	NF		10K	NF	0x6F
NF	NF		NF	NF	0x68 *** NOT ALLOWED ***

Note: 0x68 is the I2C address of the RTC on the CPU PCB

MCP3424 Jumper Settings

Analog Input Connections – Jumper Settings



	CH1	CH2	CH3	CH4
0-2V	LK1	LK5	LK10	LK9
0-5V	LK2	LK6	LK13	LK11
0-20mA	LK3	LK7	LK14	LK12
Single / Differential	LK4	LK8	LK16	LK15

Therefore for +5V Analog Input on Channel 1 (Single Ended) fit LK2 and LK4

Therefore for +2V Analog Input on Channel 2 (Differential) fit LK5 and LK8

MCP3424 Datasheet

http://ww1.microchip.com/downloads/en/DeviceDoc/22088c.pdf

MCP3424 C Drivers

https://github.com/abelectronicsuk/ABElectronics_C_Libraries/blob/master/ADCDifferetialPi/ABE_ADCDifferentialPi.c

MCP3424 Python Script

sudo git clone https://pypi.python.org/pypi/MCP342x

alternative

sudo git clone https://github.com/lachtanek/python-MCP342x/

cd python-MCP342X
sudo python setup.py install

MCP3424 Kernel Driver

sudo su
cd \

root@raspberrypi:~# modprobe mcp3422
root@raspberrypi:~# lsmod
Module Size Used by
mcp3422 4965 0
industrialio 45124 1 mcp3422
rtc_ds1307 10459 0
i2c_dev 6386 0
sg 20575 0
ftdi_sio 32602 0
sierra 8968 0
usbserial 29593 2 sierra,ftdi_sio
i2c_bcm2708 5988 0
spi_bcm2835 7868 0
bcm2835_gpiomem 3703 0
bcm2835_rng 2207 0
uio_pdrv_genirq 3526 0
uio 10078 1 uio_pdrv_genirq

```
root@raspberrypi:~# echo "mcp3424 0x6e" >/sys/bus/i2c/devices/i2c-0/new_device

ln -s /sys/bus/i2c/devices/1-006e/iio:device0/ /dev/i2cadc
```

root@raspberrypi:~# ln -s /sys/bus/i2c/devices/1-006e/iio:device0/ /dev/i2cadc
root@raspberrypi:~# ls -lls /dev/i2cadc/

total 0
-r--r--r-- 1 root root 4096 Mar 9 19:27 dev
-rw-r--r-- 1 root root 4096 Mar 9 19:27 in_voltage0_raw
-rw-r--r-- 1 root root 4096 Mar 9 19:27 in_voltage0_scale
-rw-r--r-- 1 root root 4096 Mar 9 19:27 in_voltage1_raw
-rw-r--r-- 1 root root 4096 Mar 9 19:27 in_voltage1_scale
-rw-r--r-- 1 root root 4096 Mar 9 19:27 in_voltage2_raw
-rw-r--r-- 1 root root 4096 Mar 9 19:27 in_voltage2_scale
-rw-r--r-- 1 root root 4096 Mar 9 19:27 in_voltage3_raw
-rw-r--r-- 1 root root 4096 Mar 9 19:27 in_voltage3_scale
-rw-r--r-- 1 root root 4096 Mar 9 19:27 in_voltage_sampling_frequency
-r--r--r-- 1 root root 4096 Mar 9 19:27 in_voltage_scale_available
-r--r--r-- 1 root root 4096 Mar 9 19:27 name
-r--r--r-- 1 root root 4096 Mar 9 19:27 sampling_frequency_available
lrwxrwxrwx 1 root root 0 Mar 9 19:27 su

bsystem -> ../../../../../../bus/iio
-rw-r--r-- 1 root root 4096 Mar 9 19:27 uevent

MCP3424 Examples

+1.8V supply has been attached to ADC Channel 0 to then read the ADC channel run the following

```
$ cat /dev/i2cadc/in_voltage0_scale
0.001000000

$ cat /dev/i2cadc/in_voltage0_raw

1790
```

So 1790 x 0.001000000 = 1.790V - which in comparison to a DVM reading of 1.793 is pretty good.

Changing the sample frequency affects the accuracy of the ADC reading taken, as taking fewer samples leads to a higher resolution conversion, which is especially important when working with low voltage or slowly varying signals. Changing the sample frequency will also alter the available scaling factors.

The first step is checking the available sample frequencies and then echo the chosen one into the input sample frequency controller

```
# cat /dev/i2cadc/sampling_frequency_available
240 60 15 3
# cat /dev/i2cadc/in_voltage_sampling_frequency
240
# echo 15 >/dev/i2cadc/in_voltage_sampling_frequency
# cat /dev/i2cadc/in_voltage_sampling_frequency
15
# cat /dev/i2cadc/in_voltage0_raw
28635
# cat /dev/i2cadc/in_voltage0_scale
0.000062500
# cat /dev/i2cadc/in_voltage_scale_available
0.000062500 0.000031250 0.000015625 0.000007812
```

So here the lower sample rate has given way to a more accurate reading, and using the same input voltage as before :

28635 x 0.000062500 = 1.7896875V

Likewise the input scaling multiplier can be altered in a similar fashion

```
# cat /dev/i2cadc/in_voltage_scale_available
0.001000000 0.000500000 0.000250000 0.000125000
# echo 0.000125000 >/dev/i2cadc/in_voltage0_scale
# cat /dev/i2cadc/in_voltage0_raw
2047
```

If the scaling is set too high, the amplifier stage is before the converter stage within the device, then the voltage presented to the ADC stage for conversion will be greater than the 2.048V internal reference and as a result you'll just see the max value (FSD) of the ADC i.e. 2047 (2.047V)

Note that you do not need to be root user to read these values, but only the root user can alter the sample or scaling factors