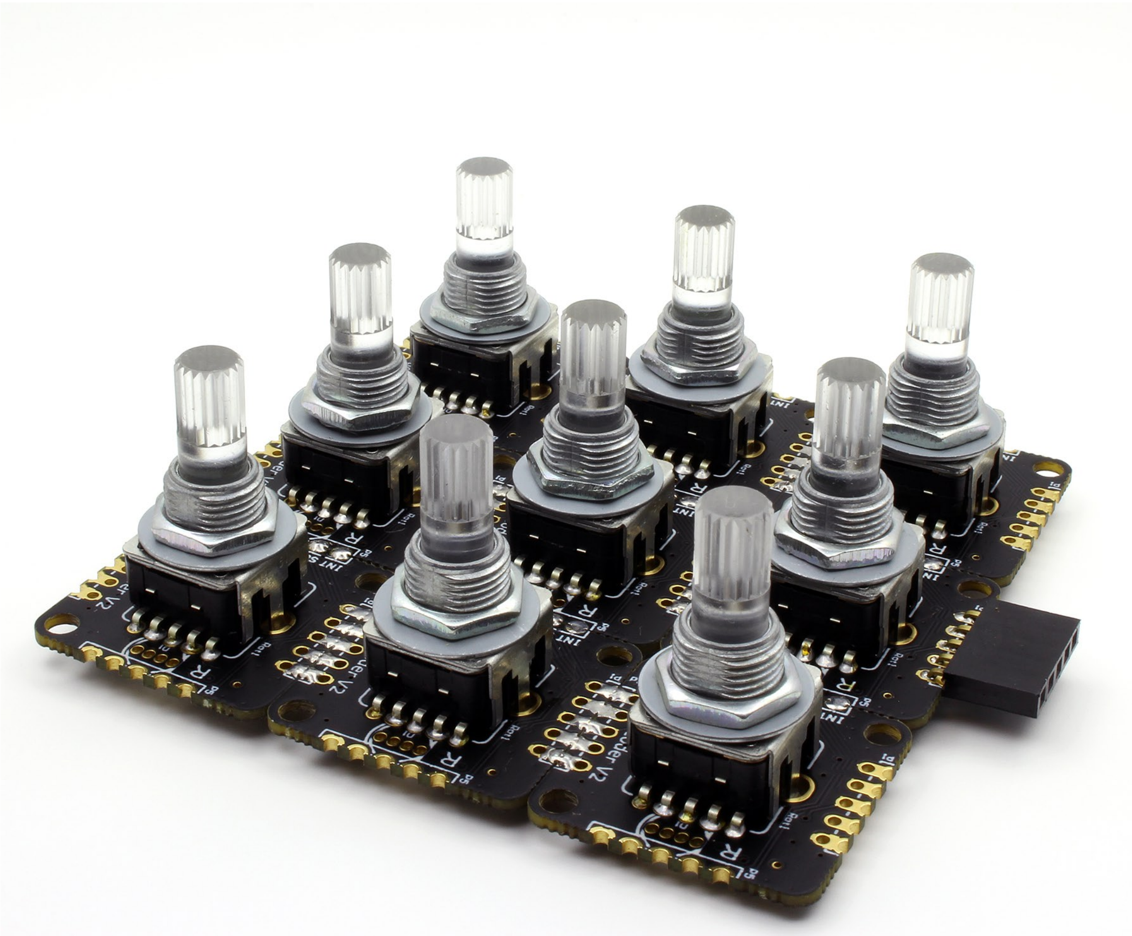
DuPPa

www.duppa.net



I2C Encoder V2.1

HW V2.1

February 16, 2020

Document written in LaTeX

Revision History

Revision	Date	Author(s)	Description
1.0	09.07.19	Simone	Update to V2.1

Contents

1. Device Overview

The I2C Encoder V2.1 is a small board where you can use a classical mechanical encoder, or an illuminated RGB encoder on I^2C bus. The device has also 3 configurable GPIOs with the same footprint of RGB LED. It's possible to connect up to 127 boards in cascade and read all of them with the same I^2C bus.
The I2C Encoder V2.1 has a series of 8 bit registers where it is possible to configure the parameters and four 32 bit of registers. These 32 bit registers store *counter value*, value of *increment steps*, *maximum* and *minimum thresholds*. Every time when encoder rotates at least one step, the *counter value* increases or decreases according to the rotation direction by the value of the *increment steps* register .
When the *counter value* is outside of the limit set by the *thresholds registers*, the counter value can be wrapped or can stuck on the threshold value reached.
The I2C Encoder V2.1 also has an open-drain interrupt pin. It is set to logic low every time an interrupt occurs, the source of interrupt can be customized.

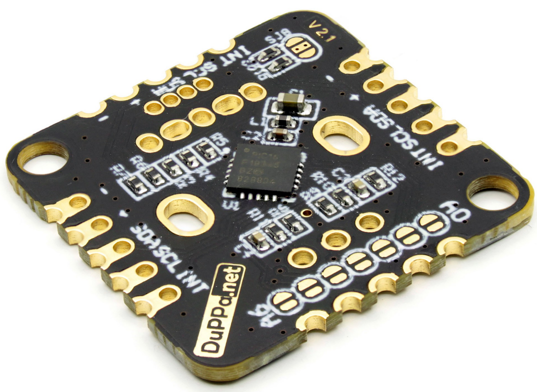


Figure 1.1: I2C Encoder V2.1 bottom view

1.1 Electrical characteristics

Parameter	Symbol	Min	Max
Supply voltage	V_{DD}	3.3V	5V
I^2C input-low level	V_{IL}	0	0.3 * V_{DD}
I^2C input-high level	V_{IH}	0.8 * V_{DD}	V_{DD}
I^2C clock input frequency	f_{SCL}		400kHz
Encoder frequency	f_{ENC}		100Hz
Current at 5V	I_{DD5V}	1.9mA (RGB LED off)	57mA (RGB LED on)
Current at 3.3V	I_{DD3V3}	1.7mA (RGB LED off)	25mA (RGB LED on)
Interrupt pull-up resistor	R_{INT}	15kΩ	120kΩ

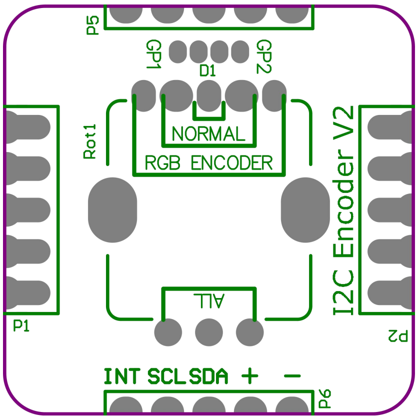


Figure 1.2: Top view of the board

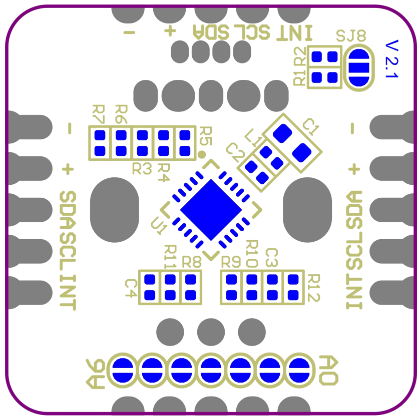


Figure 1.3: Bottom view of the board

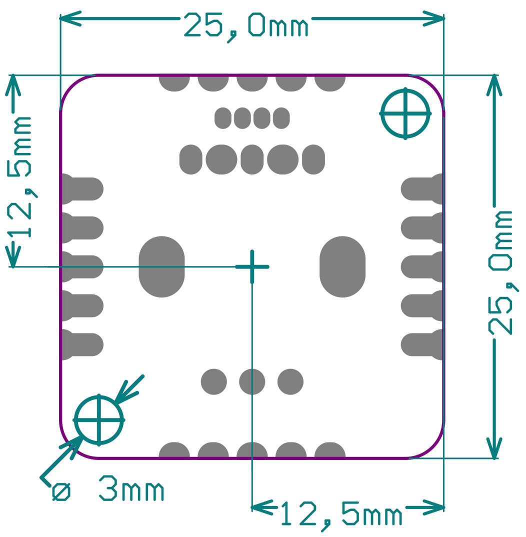


Figure 1.4: Dimensions of the board

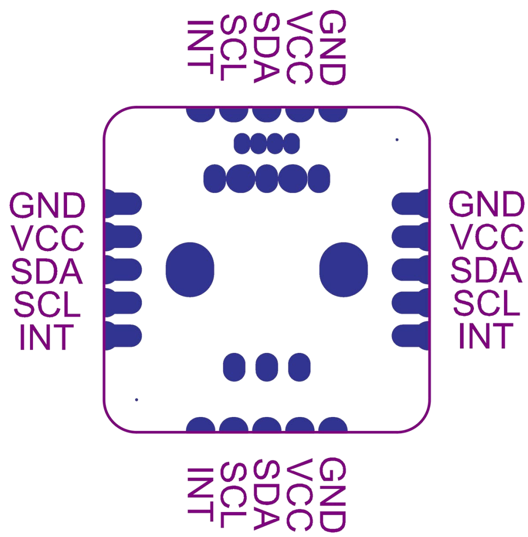


Figure 1.5: Pin-out of the board

1.2 Connection

Figure 1.5 shows the pin-out of the I2C Encoder V2.1.

Pin	I/O Type	Function
GND	Power	Ground reference for logic
Vcc	Power	Positive supply for logic
SDA	I/O	I^2C data
SCL	I	I^2C clock
INT	OD	Open-drain interrupt output

There are two 5 pin headers on the right and left sides of the I2C Encoder V2.1. There are also 5 castellated holes on each four side.
The I2C Encoder V2.1 can be connected together in a matrix by soldering the castellated holes. In order to avoid I^2C address conflict, the address of each device must be different. In the section 1.3, it is described how to set the address.

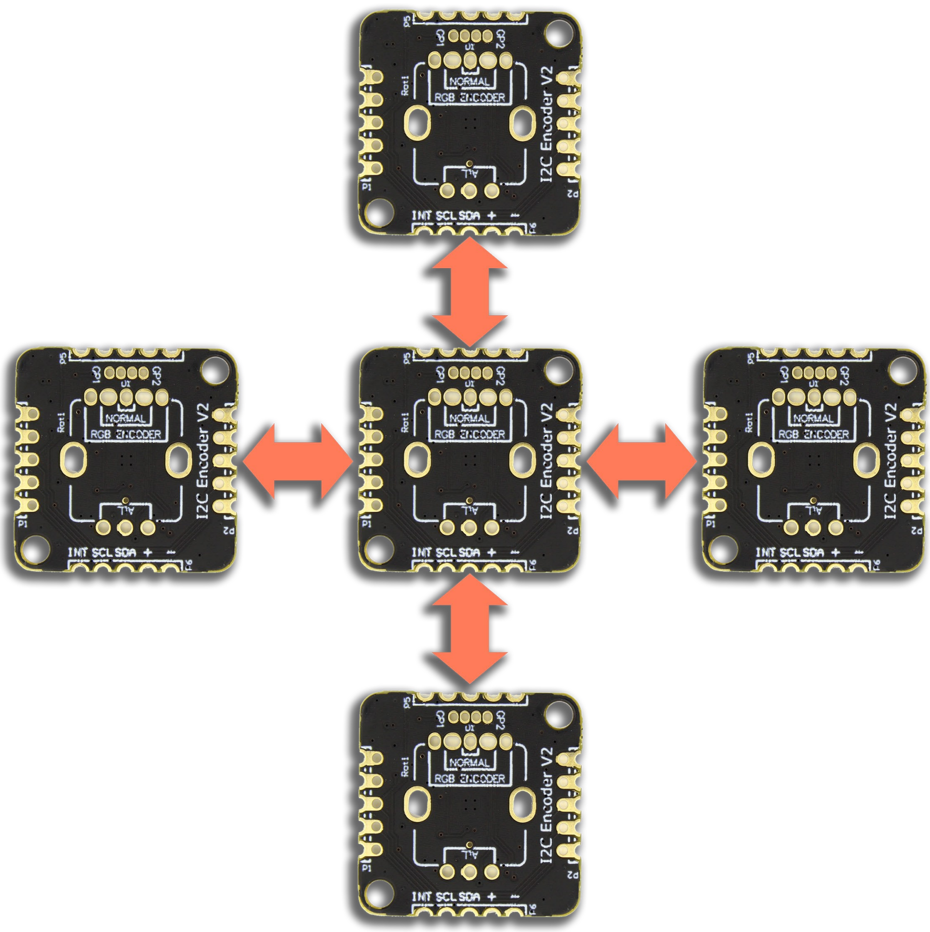


Figure 1.6: Easy way to create a custom encoder matrix

1.3 I^2C interface

The I2C Encoder V2.1 is a I^2C slave. In order to be as compatible as possible, the clock stretch feature is disabled by default, it is possible to enable it by setting the bit **CKSRC** of the second configuration register.

The 7-bit I^{2}C address is represented in a binary number with the jumpers A0 - A6 shown in the figure 1.7. It can be customized by soldering the jumpers. When the jumper is open, it means a logic 0. if jumper is shorted it means a logic 1. In figure 1.9 is shown some examples of I^{2}C address setting.
Avoid to set the address 0, all the jumpers open. The address 0 is a reserved address in the I^{2}C bus.

I^{2}C address							
7	6	5	4	3	2	1	0
A6	A5	A4	A3	A2	A1	A0	R/W

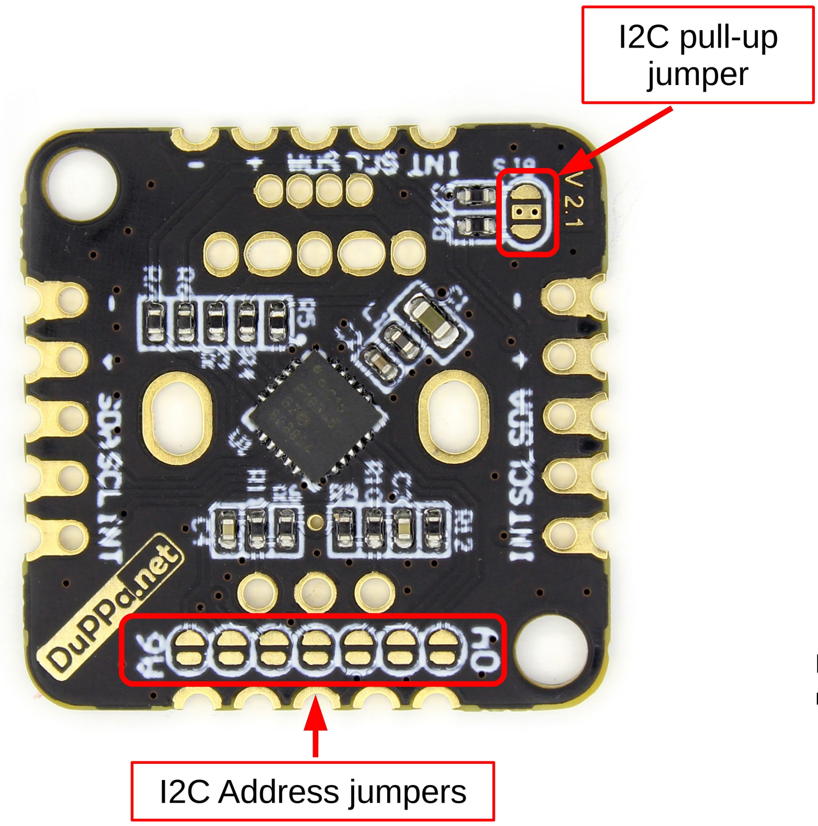


Figure 1.7: Jumpers location

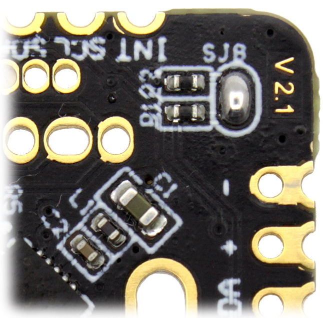


Figure 1.8: I^{2}C jumper pull-up correctly shorted

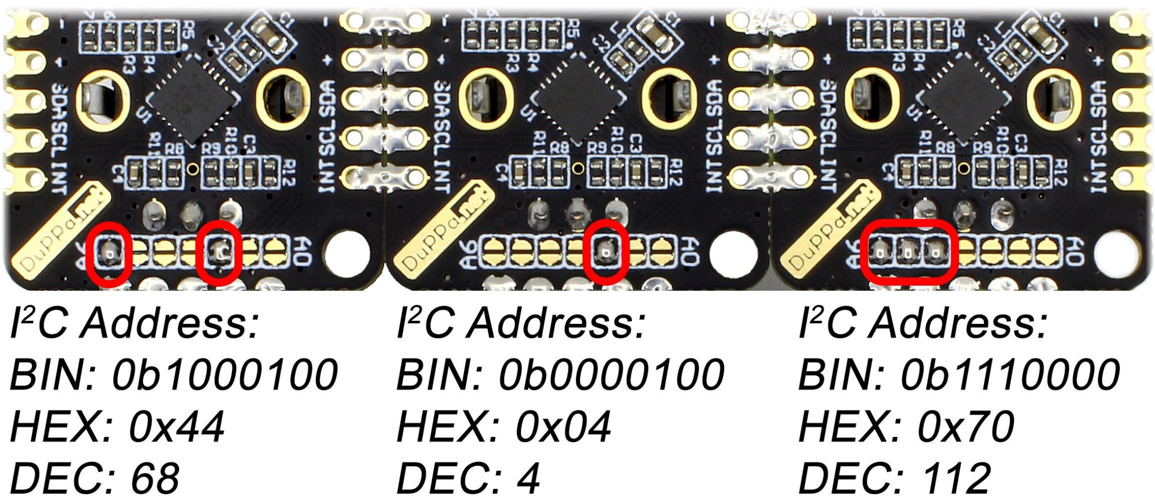


Figure 1.9: Example of the address setting

The I2C Encoder V2.1 has built in 4.7kΩ pull-up resistors for the I^{2}C bus. By default the resistors are disabled. It's possible to enable them by soldering the jumper SJ8, in the figure 1.7 is showed the correct soldering. This must be done in case that the master doesn't have these resistors and must be enabled only one I2C Encoder V2.1 in a chain.

1.4 Rotary encoder

On the I2C Encoder V2.1, it is possible to solder a standard rotary encoder and an illuminated RGB encoder. In the figure 1.12 is showed in witch pins with the standard encoder have to be soldered, while in the figure 1.13 is showed the pins used for the illuminated RGB encoder.

In both cases the encoder have to be soldered on the TOP face, where there are no components.

Figure 1.10: Example of assembled board with EC11 encoder, RGB LED and connectors

Figure 1.11: Example of assembled board with Illuminated RGB Encoder and connectors

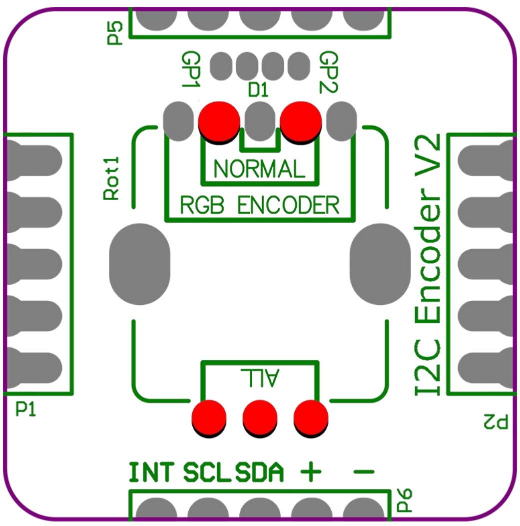


Figure 1.12: Used pins for mounting the standard rotary encoder

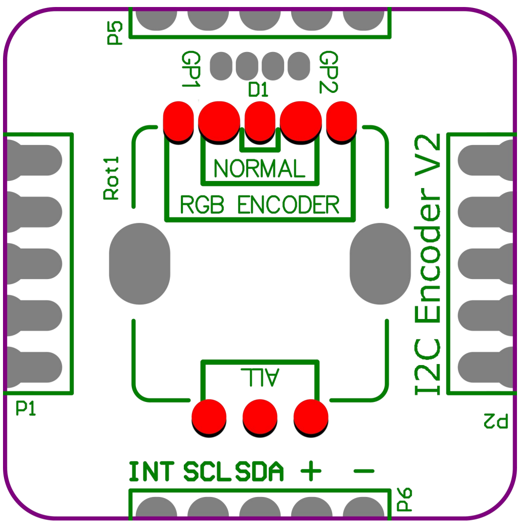


Figure 1.13: Used pins for mounting the RGB encoder

It's possible to configure several option by setting the registers with the I^{2}C bus. With the **GCONF** and **GCONF2** registers, it is possible to configure several parameters.

In the configuration, it's possible to set the rotation direction and also to set if the output of the encoder is X1 or X2.
In X1 mode the counting happens only on the falling edge of the channel A, The B channel is used for the direction. Most of the encoder have this type of encoding.
In X2 mode the it's used both the falling and rising edge of the channel A, in this way the resolution is the double respect the X1. Few encoder have this type of encoding, or can be used for the encoder without dent.

For reading the rotary encoder movement, there are 4 32bit registers: **CVAL**, **CMAX**, **CMIN** and **ISTEP**. All of these 4 registers can be configured to work as 32bit int or as IEEE 754 floating number, this format can be set in the **GCONF** register.
The I2C Encoder V2.1 can work in two different way: absolute mode and relative mode. This option can be configured with the bit **RELMODE**.
By default it works in absolute mode. Every time the encoder moves one step, the value of the **CVAL** register is increased or decreased of the value of **ISTEP**. The direction of the rotation decides if **ISTEP** is added or subtracted from **CVAL**.
CMAX and **CMIN** are used for setting a minimum and maximum thresholds of **CVAL**. In the **GCONF** register, there is **WRAPE** bit. This bit is used to enable or disable a wrap functionality of **CVAL** when it exceeds from the thresholds.
If the bit **RELMODE** the I2C Encoder V2.1 works in relative mode, this mode is similar to the absolute mode, but every time the register **CVAL** is read it is automatically set to 0. In this way you know the number of step between two **CVAL** reading.

The counter limits are the following:

- **32bit INT:** from $-2.147.483.648$ to $+2.147.483.647$
- **IEEE 754 float:** from $-126,0$ to $+127,0$

In case of 32bit INT, it is not necessary to read all the 4 bytes, you can read only the first 8 bit or the first 16 bit. For example, if you want to count between 0 and 10, you can read only the first byte of the **CVAL** register. In this way you can save I^2C transactions.

Here a couple of example of how the I2C Encoder V2.1 works:

- **CVAL**$=0$
- **CMAX** $=5$
- **CMIN** = -5
- **ISTEP**$=1$

I will have **CVAL** is incremented of 1 at each rotation step of the encoder. The maximum value that **CVAL** can reach will be 5 while the minimum is -5. In the figure 1.14 shows the value of **CVAL**.
As showed in the figure 1.15, when **WRAPE** is set to 1 when **CVAL** reaches the value of 5, at the next increment **CVAL** it will be wrapped to -5.
Every time when the encoder is rotated one step and when **CVAL** touch the thresholds, an interrupt is generated and is possible to read in the register **ESTATUS**.
The I2C Encoder V2.1 support also the rotary encoder with the push button. When the push button is pressed an interrupt is generated at the rising and falling edge. In this way, it is possible to check when the push button is pressed or released.
There is also possibility to read a fast double push by setting a window time in the register **DPPERIOD**. When a double push is made inside of the **DPPERIOD** window, an interrupt is generated.
If the **DPPERIOD** is 0, the double push function is disabled.
All the above interrupt are possible to read in the register **ESTATUS**, and can be also disabled with the register **INTCONFIG**.

The I2C Encoder V2.1 has a RC filer as anti-bouncing filter, but sometimes is not enough especially when the encoder is rotated slowly. The bouncing can happen and you can see it as fast rotation in the opposite direction. To avoid this problem, there is the register **ANTBOUNC**. With this register is possible to set a period where the inversion of rotation can be ignored.

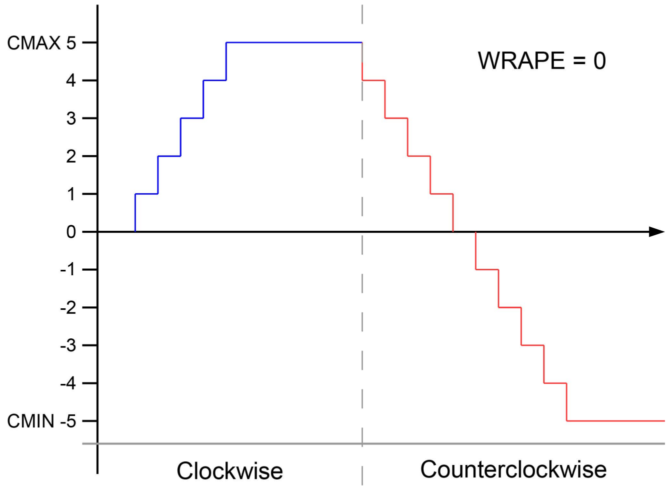


Figure 1.14: Blue and red line are the **CVAL** values when the encoder is rotate and the **WRAPE** is disabled

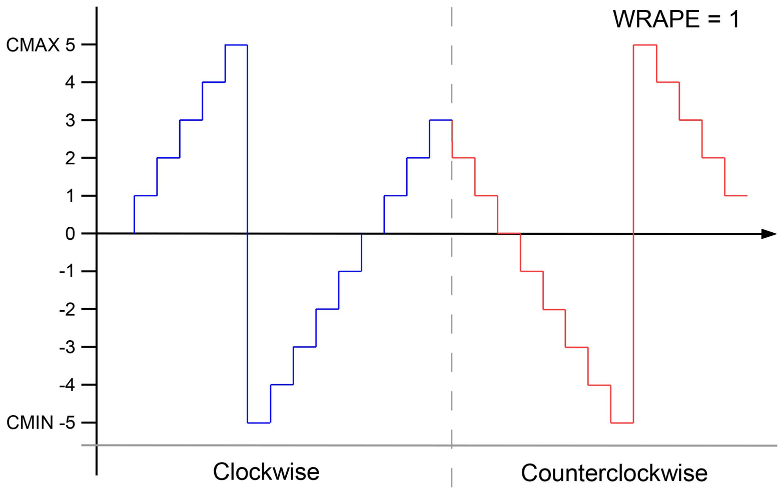


Figure 1.15: Blue and red line are the **CVAL** values when the encoder is rotate and the **WRAPE** is enabled

1.5 GP pins

There are 3 configurable GP pins, they are called GP1, GP2 and GP3. They have the same footprint of an 5mm RGB led. In figure 1.16, it is shown the position. Each pin can be configured to be:

- **PWM:** The pins is configured to be a PWM output.
- **Analog input:** The pins is configured to be an input of the internal ADC, the resolution is 8 bit.
- **GPIO output:** The pins is configured to be a digital output.
- **GPIO Input:** The pin is configured to be a digital input. It's possible to set an interrupt at every signal edge.

In case of a RGB encoder is mounted, the GP3 will not be available. This because GP3 is internally connect to the red color of the RGB led and it is automatically configured to be a PWM.

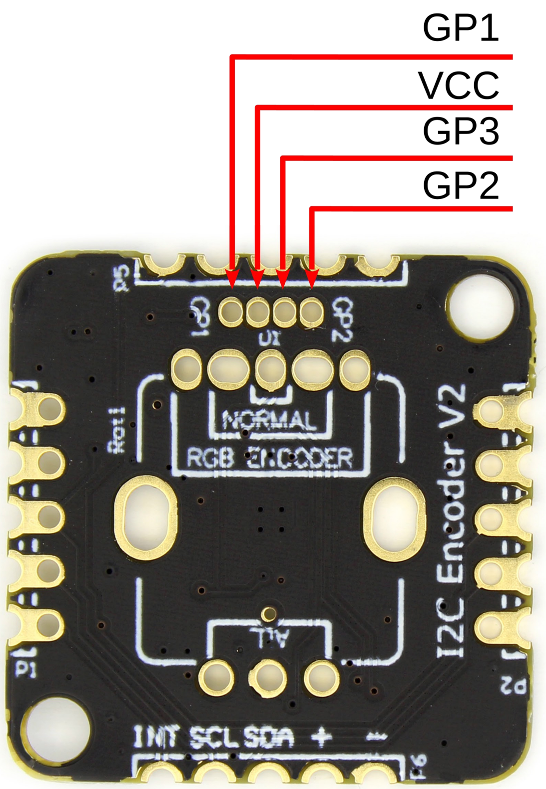


Figure 1.16: GP pins pinout

1.6 PWM

The I2C Encoder V2.1 have 6 configurable PWM outputs, 3 belong to the RGB LED of the RGB encoder, while the other 3 can be the GP pins. If the RGB Encoder is configured, only 2 PWM are available on the GP pins. The corresponding register are described in the section 2.5 for the RGB LEDs, and in the section 2.6 for the GP pins. There are a couple of features around the PWM, there is an automatic fading feature and the gamma correction in order to optimize the LED light.
The PWM resolution 10bit with a frequency of 31.25kHz, but the user can not access to the full resolution. By default the gamma correction in off and the user can access of a resolution of 8bit, from minimum is 0 and maximum of 255. [1] If a gamma correction is chose the PWM range change to maximum 100. In this way it's possible to set a PWM value between 0% to 100% by writing 0 to 100 to the corresponding register.
In both case when the PWM is set to 0 the output is VCC, while when is set to it's maximum the output is GND The PWM value can be changed in any moment during the board operation.

1.6.1 Fade function

The I2C Encoder V2.1 has an auto fading feature. This function can be enabled for the RGB LED of the rotary encoder, or in the PWM output of the GP pins.
There are two registers:

- **FADERGB**: for setting the RGB LED of the rotary encoder
- **FADEGP**: for setting the GP pins

The value of these registers is the step speed of the fading ramp, and it's in *ms*. If the value is 0, the fade feature is disabled. It means that when the new value of PWM is written, it is immediately updated to the output.

[1] In this way it's compatible with the previous version I2C Encoder V2

The fading process starts when a PWM value is written. Fading process is complete according to the PWM value you have set. When the internal fade PWM value is the same of the PWM value (it means that when fading is complete), an interrupt will be generated. Let's make an example:

At the startup, i write as:

- **RLED** = 0
- **GLED** = 0
- **BLED** = 0
- **FADERGB** = 1

In above case, the LEDs of the encoder are off and the fade has 1ms step size.
Now i write as: *RLED=0xFF*.
This moment, the ramp of the **RLED** starts and reaches the value 0xFF in 255ms. (the other LEDs remain OFF)
When the ramp reaches 0xFF, i get an interrupt from the **INT** pin.
After this, i write as: *RLED=0x00*. At this point, the fade ramp starts again and it will turn off the LED in 255ms, and an interrupt will be generated at the end.
The figure 1.17 is showing an example of the fade output when a PWM value is written in different moments, as well as when the interrupt of the fade is generated.

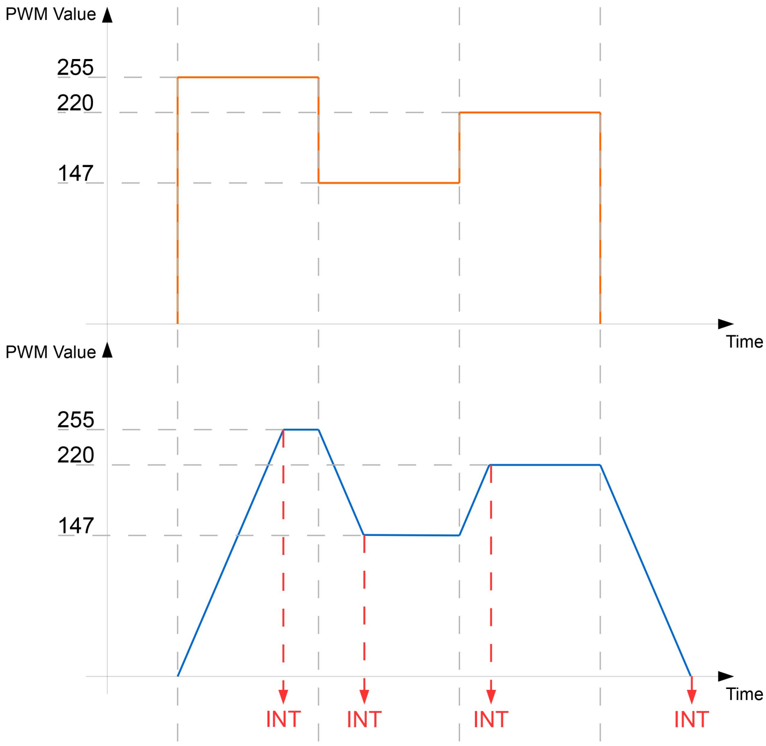


Figure 1.17: Time diagram of the fade output when writing on the PWM register

1.6.2 Gamma correction

The I2C Encoder V2.1 have in build a gamma correction function. This correction can be applied to all of the PWM output, such as the LEDs of the RGB encoder and the GP pins. The gamma correction is very useful in case you are using a LED, this because it make the luminosity of the LED more linear to our eys according to the PWM value. [2]
There are six register where to configure the gamma correction and they are described in the section 2.8. It's possible to configure 7 type of curve:

- Gamma OFF
- Gamma 1 (linear)
- Gamma 1.8
- Gamma 2.0
- Gamma 2.2
- Gamma 2.4
- Gamma 2.6
- Gamma 2.8

By default, the gamma correction is OFF, this means that it is possible to access to a 8bit PWM resolution. In the figure 1.18 is showed the relationship between the PWM at the full resolution and the PWM settable from the user.

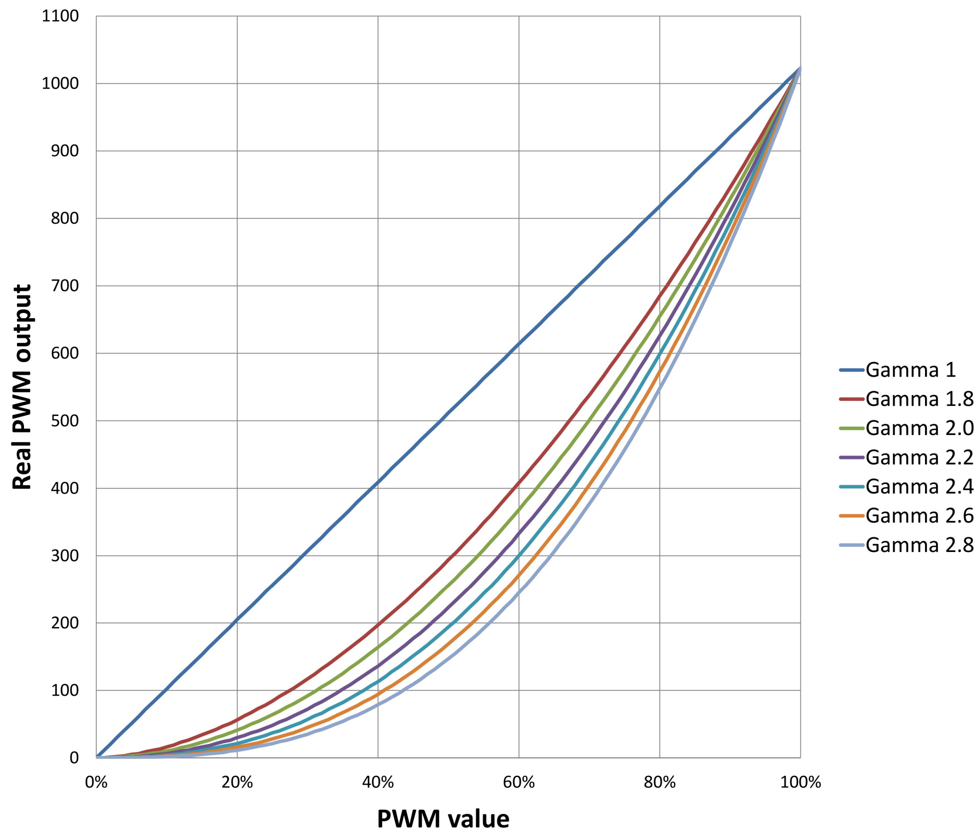


Figure 1.18: Possible gamma option

[2] For more information: https://en.wikipedia.org/wiki/Gamma_correction

1.7 EEPROM

The I2C Encoder V2.1 has 256 bytes of EEPROM.
This memory is divided in two banks of 128 bytes. With the bit **MBANK** in the **GCONF** register, it is possible to choose the bank 1 or the bank 2.
The memory area is between 0x80 and 0xFF address. To use the EEPROM, user only needs to perform reading or a writing in these address areas.
The writing time takes 4 - 5ms to be executed. Wait this time before sending other commands.

1.8 Interrupt

The I2C Encoder V2.1 has multiple interrupt source previously described. When an interrupt is generated, the **INT** pin is tied low. By reading the register **ESTATUS**, the interrupts are cleared and the **INT** pin returns high.
The INT pin is open-drain output. Hence it requires an external pull-up resistor, or internal pull-up resistor can be enabled by setting the bit **IPUD** to 1.
In a chain of I2C Encoder V2.1 all the **INT** pins can be connected together, like the pin of the I^2C. When an interrupt occurs, user has to scan the boards in the chain to find who generates the interrupt.
With the register **INTCONF**, it is possible to enable or disable interrupt. When an interrupt is disabled, the corresponding bit is set, but the **INT** pin is not affected.

2. Registers

In this section, the internal registers of I2C Encoder V2.1 is described.
The I2C Encoder V2.1 has the *auto increment* feature. This means that after writing or reading a register, the internal address pointer is automatically incremented by one.

Address range	Name	Description	Dimension	Default value
0x00	GCONF	General Configuration	1 Byte	0
0x01	GP1CONF	GP 1 Configuration	1 Byte	0
0x02	GP2CONF	GP 2 Configuration	1 Byte	0
0x03	GP3CONF	GP 3 Configuration	1 Byte	0
0x04	INTCONF	INT pin Configuration	1 Byte	0
0x05	ESTATUS	Encoder Status	1 Byte	0
0x06	I2STATUS	Secondary interrupt status	1 Byte	0
0x07	FSTATUS	Fade process status	1 Byte	0
0x08 - 0x0B	CVAL	Counter Value	4 Byte	0
0x0C - 0x0F	CMAX	Counter Max value	4 Byte	0
0x10 - 0x13	CMIN	Counter Min value	4 Byte	0
0x14 - 0x17	ISTEP	Increment step value	4 Byte	0
0x18	RLED	LED red color intensity	1 Byte	0
0x19	GLED	LED green color intensity	1 Byte	0
0x1A	BLED	LED blue color intensity	1 Byte	0
0x1B	GP1REG	I/O GP1 register	1 Byte	0
0x1C	GP2REG	I/O GP2 register	1 Byte	0
0x1D	GP3REG	I/O GP3 register	1 Byte	0
0x1E	ANTBOUNC	Anti-bouncing period	1 Byte	25
0x1F	DPPERIOD	Double push period	1 Byte	0
0x20	FADERGB	Fade timer RGB Encoder	1 Byte	0
0x21	FADEGP	Fade timer GP ports	1 Byte	0
0x27	GAMRLED	Gamma correction on red LED	1 Byte	0
0x28	GAMGLED	Gamma correction on green LED	1 Byte	0
0x29	GAMBLED	Gamma correction on blue LED	1 Byte	0
0x2A	GAMMAGP1	Gamma correction on PWM of GP1	1 Byte	0
0x2B	GAMMAGP2	Gamma correction on PWM of GP2	1 Byte	0
0x2C	GAMMAGP3	Gamma correction on PWM of GP3	1 Byte	0
0x30	GCONF2	Second configuration	1 Byte	0
0x70	IDCODE	I2C Encoder V2.1 unique code	1 Byte	0x53
0x71	VERSION	I2C Encoder V2.1 version	1 Byte	0x21
0x80 - 0xFF	EEPROM	EEPROM memory	128 Byte	0

2.1 Configuration

2.1.1 General Configuration

GCONF Address: **0x00**							
7	6	5	4	3	2	1	0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RESET	MBANK	ETYPE	RMOD	IPUD	DIRE	WRAPE	DTYPE

❖ **DTYPE** Data type of the register: **CVAL**, **CMAX**, **CMIN** and **ISTEP**.

1: The registers are considered float numbers IEEE 754

0: The registers are considered int 32bit

❖ **WRAPE** Enable counter wrap.

1: Wrap enable. When the counter value reaches the **CMAX**+1, restart to the **CMIN** and vice versa

0: Wrap disable. When the counter value reaches the **CMAX** or **CMIN**, the counter stops to increasing or decreasing

❖ **DIRE** Direction of the encoder when increment.

1: Rotate left side to increase the value counter

0: Rotate right side to increase the value counter

❖ **IPUD** Interrupt Pull-UP disable.

1: Disable

0: Enable

❖ **RMOD** Reading Mode.

1: X2 mode

0: X1 mode

❖ **ETYPE** Set the encoder type

1: RGB illuminated encoder

0: Standard encoder

❖ **MBANK** Select the EEPROM memory bank. Each bank are 128 byte wide

1: Second memory bank

0: First memory bank

❖ **RST** Reset of the I2C Encoder V2.1

1: Reset of the I2C Encoder V2.1. The RESET command takes 400us to be executed.

0: No reset

2.1.2 Second Configuration

This register is available only in the I2C Encoder V2.1.

GCONF2 Address: **0x30**							
7	6	5	4	3	2	1	0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
-	-	-	-	-	-	RELMOD	CKSRC

❖ **CKSRC** I^2C clock stretch option.

1: Enable the I^2C clock stretch.

0: Enable the I^2C clock stretch.

❖ **RELMOD** CVAL working mode

1: Enable the relative mode of the CVAL register.

0: Disable the relative mode of the CVAL register.

2.1.3 GP1 Configuration

GP1CONF Address: **0x01**							
7	6	5	4	3	2	1	0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
-	-	-	GP1INT		GP1PUL	GP1MODE	

❖ **GP1MODE** Configuration of the pin GP1

00: PWM output
01: GPIO output Push-Pull
10: Analog input
11: GPIO Input

❖ **GP1PUL** Enable or disable the internal pull-up.

0: Pull-UP disable
1: Pull-UP enabled

❖ **GP1INT** Configuration of the interrupt, available only when the pin is configured as input

00: Interrupt disabled
01: Interrupt on positive edge
10: Interrupt on negative edge
11: Interrupt on both edges

2.1.4 GP2 Configuration

GP2CONF Address: **0x02**							
7	6	5	4	3	2	1	0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
-	-	-	GP2INT		GP2PUL	GP2MODE	

❖ **GP2MODE** Configuration of the pin GP2

00: PWM output
01: GPIO output Push-Pull
10: Analog input
11: GPIO Input

❖ **GP2PUL** Enable or disable the internal pull-up.

0: Pull-UP disable
1: Pull-UP enabled

❖ **GP2INT** Configuration of the interrupt, available only when the pin is configured as input

00: Interrupt disabled
01: Interrupt on positive edge
10: Interrupt on negative edge
11: Interrupt on both edges

2.1.5 GP3 Configuration

GP3CONF Address: **0x03**							
7	6	5	4	3	2	1	0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
-	-	-	GP3INT		GP3PUL	GP3MODE	

Note: When the **ETYPE** bit is set, this pin is in PWM mode automatically and can't be changed.

❖ **GP3MODE** Configuration of the pin GP3

00: PWM output

01: GPIO output Push-Pull

10: Analog input

11: GPIO Input

❖ **GP3PUL** Enable or disable the internal pull-up.

0: Pull-UP disable

1: Pull-UP enabled

❖ **GP3INT** Configuration of the interrupt, available only when the pin is configured as input

00: Interrupt disabled

01: Interrupt on positive edge

10: Interrupt on negative edge

11: Interrupt on both edges

2.2 Interrupt output Configuration

This register is used for enable or disable the interrupt source selectively. When an interrupt event occurs, the **INT** pin goes low and the event is stored in the status register.

INTCONF Address: **0x04**							
7	6	5	4	3	2	1	0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INT2	IRMIN	IRMAX	IRDEC	IRINC	IPUSHD	IPUSHP	IPUSHR

❖ **IPUSHR** Push button release bit

1: Interrupt enabled when the push button is released.

0: Interrupt disabled

❖ **IPUSHP** Push button press bit

1: Interrupt enabled when the push button is pressed.

0: Interrupt disabled

❖ **IPUSHD** Push button double press

1: Interrupt enabled when the push button is double pressed.

0: Interrupt disabled

❖ **IRINC** Rotary encoder direction of increase

1: Interrupt enabled when the encoder is rotated in the direction of increase

0: Interrupt disabled

❖ **IRDEC** Rotary encoder direction of decrease

1: Interrupt enabled when the encoder is rotated in the direction of decrease

0: Interrupt disabled

❖ **IRMAX** **CVAL** reaches **CMAX** bit

1: Interrupt enabled when **CVAL** reaches **CMAX**

0: Interrupt disabled

❖ **IRMIN** **CVAL** reaches **CMIN** bit

1: Interrupt enabled when **CVAL** reaches **CMIN**

0: Interrupt disabled

❖ **INT2** Enable the secondary interrupts

1: Secondary interrupt enabled

0: Secondary interrupt disabled

2.3 Status

2.3.1 Encoder Status

This register if only readable, when is read is automatically cleared.

ESTATUS Address: **0x05**							
7	6	5	4	3	2	1	0
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
INT2	RMIN	RMAX	RDEC	RINC	PUSHD	PUSHP	PUSHR

❑ **PUSHR** Status of the push button of the encoder

1: Push button is released

0: Push button is not released

❑ **PUSHP** Status of the push button of the encoder

1: Push button is pressed

0: Push button is not pressed

❑ **PUSHD** Status of the push button of the encoder

1: Push button is double pressed

0: Push button is not double pressed

❑ **RINC** Rotary encoder is rotated in the increase direction

1: Encoder is rotated

0: Encoder is not rotated

❑ **RDEC** Rotary encoder is rotated in the decrease direction

1: Encoder is rotated

0: Encoder is not rotated

❑ **RMAX** Status of the counter value

1: **CVAL** reaches the **CMAX** value

0: **CVAL** is below the **CMAX** value

❑ **RMIN** Status of the counter value

1: **CVAL** reaches the **CMIN** value

0: **CVAL** is above the **CMIN** value

❑ **SINT2** Secondary interrupt status

1: Secondary interrupt event occurs

0: No secondary event occurs

2.3.2 Secondary interrupt status

In this register is possible to check the event of the fade process as well as the GP pins when is configured as input. This register if only readable, when is read is automatically cleared.

I2STATUS Address: **0x06**							
7	6	5	4	3	2	1	0
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
-	FADE	EGP3		EGP2		EGP1	

❑ **GP1** Configuration of the GP1

00: No event occurs

01: Positive edge event

10: Negative edge event

❑ **GP2** Configuration of the GP2.

00: No event occurs

01: Positive edge event

10: Negative edge event

❑ **GP3** Configuration of the GP3. Available only with the standard encoder

00: No event occurs

01: Positive edge event

10: Negative edge event

❑ **FADE** Fade event occurs

1: Fade process event

0: No event in the fade process

2.3.3 Fade process status

With this register is possible to check the status of the PWM channel during the fade process. This register is only readable and the bits are set or cleared automatically during the fade process.

FSTATUS Address: **0x07**							
7	6	5	4	3	2	1	0
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
-	-	FGP3	FGP2	FGP1	FEB	FEG	FER

- **FER** Fade status process of the RGB Encoder color red

  1: Fade process is running

  0: Fade process terminated

- **FEG** Fade status process of the RGB Encoder color green

  1: Fade process is running

  0: Fade process terminated

- **FEB** Fade status process of the RGB Encoder color blue

  1: Fade process is running

  0: Fade process terminated

- **GP1** Fade status process of the GP pin 1

  1: Fade process is running

  0: Fade process terminated

- **GP2** Fade status process of the GP pin 2

  1: Fade process is running

  0: Fade process terminated

- **GP3** Fade status process of the GP pin 3

  1: Fade process is running

  0: Fade process terminated

2.4 Encoder registers

2.4.1 Counter Value

CVAL Address: **0x08**							
31	30	29	28	27	26	25	24
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CVAL BYTE 4 <31 - 24>							
Address: **0x09**							
23	22	21	20	19	18	17	16
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CVAL BYTE 3 <23 - 16>							
Address: **0x0A**							
15	14	13	12	11	10	9	8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CVAL BYTE 2 <15 - 8>							
Address: **0x0B**							
7	6	5	4	3	2	1	0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CVAL BYTE 1 <7 - 0>							

2.4.2 Counter Max

CMAX Address: **0x0C**							
31	30	29	28	27	26	25	24
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CMAX BYTE 4 <31 - 24>							
Address: **0x0D**							
23	22	21	20	19	18	17	16
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CMAX BYTE 3 <23 - 16>							
Address: **0x0E**							
15	14	13	12	11	10	9	8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CMAX BYTE 2 <15 - 8>							
Address: **0x0F**							
7	6	5	4	3	2	1	0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CMAX BYTE 1 <7 - 0>							

2.4.3 Counter Min

CMIN Address: **0x10**							
31	30	29	28	27	26	25	24
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CMIN BYTE 4 <15 - 8>							
Address: **0x11**							
23	22	21	20	19	18	17	16
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CMIN BYTE 3 <7 - 0>							
Address: **0x12**							
15	14	13	12	11	10	9	8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CMIN BYTE 2 <15 - 8>							
Address: **0x13**							
7	6	5	4	3	2	1	0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CMIN BYTE 1 <7 - 0>							

2.4.4 Increment step

ISTEP Address: **0x14**							
31	30	29	28	27	26	25	24
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ISTEP BYTE 4 <15 - 8>							
Address: **0x15**							
23	22	21	20	19	18	17	16
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ISTEP BYTE 3 <7 - 0>							
Address: **0x16**							
15	14	13	12	11	10	9	8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ISTEP BYTE 2 <15 - 8>							
Address: **0x17**							
7	6	5	4	3	2	1	0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ISTEP BYTE 1 <7 - 0>							

2.5 LEDs registers

2.5.1 LED Red intensity

RLED Address: **0x18**							
7	6	5	4	3	2	1	0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RED LED PWM Value <7 - 0>							

This register is used for setting the PWM of the red LED of the RGB encoder. A value of **0x00** means PWM at 0%, LED OFF. A value of **0xFF** means PWM at 100%, LED completely ON.

2.5.2 LED Green intensity

GLED Address: **0x19**							
7	6	5	4	3	2	1	0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GREEN LED PWM Value <7 - 0>							

This register is used for setting the PWM of the green LED of the RGB encoder. A value of **0x00** means PWM at 0%, LED OFF. A value of **0xFF** means PWM at 100%, LED completely ON.

2.5.3 LED Blue intensity

BLED Address: **0x1A**							
7	6	5	4	3	2	1	0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BLUE LED PWM Value <7 - 0>							

This register is used for setting the PWM of the blue LED of the RGB encoder. A value of **0x00** means PWM at 0%, LED OFF. A value of **0xFF** means PWM at 100%, LED completely ON.

2.6 GPs registers

The usage of these register depends of the configuration of the GP pins:

➤ **PWM** Is possible to write the PWM value.

➤ **GPIO output:** Is possible to write 1 or 0 for setting the output logic level.

➤ **Analog input:** Is possible to read the ADC value

➤ **GPIO Input:** Is possible to read the logic level of the output

2.6.1 GP1 register

GP1REG Address: **0x1B**							
7	6	5	4	3	2	1	0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GP1 Value <7 - 0>							

2.6.2 GP2 register

GP2REG Address: **0x1C**							
7	6	5	4	3	2	1	0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GP2 Value <7 - 0>							

2.6.3 GP3 register

GP3REG Address: **0x1D**							
7	6	5	4	3	2	1	0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GP3 Value <7 - 0>							

2.7 Timing registers

This register are used for changing some timing parameter

2.7.1 Anti-bouncing period

This register is used for setting the anti-bounce period. The value is in ms × 10. For example, when you rotate the encoder especially at low speed, some times bouncing can happen and you can see it when rotation is in the opposite direction. With this register, it is possible to set the period where the opposite rotation is ignored. The default value is 25, that means that the de-bounce time is 250ms.

ANTBOUNC Address: **0x1E**							
7	6	5	4	3	2	1	0
R/W-0	R/W-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-0	R/W-1
PBOUN <7 - 0>							

2.7.2 Double push time

This register is used for setting the double push of the rotary encoder switch. The value is in ms × 10. When this register is 0 this function is disabled.

DPPERIOD Address: **0x1F**							
7	6	5	4	3	2	1	0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PDPUSH <7 - 0>							

2.7.3 Fade RGB encoder LED

This register is used for setting the fade speed of the RGB LED of the rotary encoder. The value is in ms and indicate the time of one step. If a standard rotary encoder is used, this register is not used.

FADERGB Address: **0x20**							
7	6	5	4	3	2	1	0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
FADERGB <7 - 0>							

2.7.4 Fade GP ports

This register is used for setting the fade speed of the GP ports. The value is in ms and indicate the time of one step. This register is working only if the GP port are configured in PWM

FADEGP Address: **0x21**							
7	6	5	4	3	2	1	0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
FADEGP <7 - 0>							

2.8 Gamma correction

This register is used for applying a gamma correction of the PWM output It can be very useful when a LED is connected.
This three registers are only for the RGB led of the RGB encoder

GAMRLED Address: **0x27**							
7	6	5	4	3	2	1	0
-	-	-	-		R/W-0	R/W-0	R/W-0
					GAMRLED <2 - 0>		

GAMGLED Address: **0x28**							
7	6	5	4	3	2	1	0
-	-	-	-		R/W-0	R/W-0	R/W-0
					GAMGLED <2 - 0>		

GAMBLED Address: **0x29**							
7	6	5	4	3	2	1	0
-	-	-	-		R/W-0	R/W-0	R/W-0
					GAMBLED <2 - 0>		

❑ **GAMxLED** Gamma curve selector for RGB led of the rotary encoder

000: Gamma OFF, the PWM register work like the V2.0
001: Gamma 1
010: Gamma 1.8
011: Gamma 2.0
100: Gamma 2.2
101: Gamma 2.4
110: Gamma 2.6
111: Gamma 2.8

This other three registers are used only when the corresponding GP pins is set to PWM.

GAMMAGP1 Address: **0x2A**							
7	6	5	4	3	2	1	0
-	-	-	-		R/W-0	R/W-0	R/W-0
					GAMMAGP1 <2 - 0>		

GAMMAGP2 Address: **0x2B**							
7	6	5	4	3	2	1	0
-	-	-	-		R/W-0	R/W-0	R/W-0
					GAMMAGP2 <2 - 0>		

GAMMAGP3 Address: **0x2C**							
7	6	5	4	3	2	1	0
-	-	-	-		R/W-0	R/W-0	R/W-0
					GAMMAGP3 <2 - 0>		

❑ **GAMMAGPx** Gamma curve selector for the PWM pins

000: Gamma OFF, the PWM register work like the V2.0
001: Gamma 1
010: Gamma 1.8
011: Gamma 2.0
100: Gamma 2.2
101: Gamma 2.4
110: Gamma 2.6
111: Gamma 2.8

2.9 I2C Encoder V2.1 unique code

This register contains an unique code that it's used to identify the I2C Encoder V2.1. This register is only readable and not writable.

IDCODE Address: **0x70**							
7	6	5	4	3	2	1	0
R-0	R-1	R-0	R-1	R-1	R-0	R-1	R-1
IDCODE = 0x53							

2.10 I2C Encoder V2.1 version

This register contains the version of the I2C Encoder V2.1. This value will change in case of a new hardware or firmware release. This register is only readable and not writable.

VERSION Address: **0x71**							
7	6	5	4	3	2	1	0
R-0	R-0	R-0	R-1	R-0	R-0	R-0	R-1
Version = 0x21							

3. Reference

This project is open source, the HW and the FW as well as some example can be found on GitHub:
https://github.com/Fattoresaimon/I2CEncoderV2.1

The HW project can be found on Circuit Maker site:
https://workspace.circuitmaker.com/Projects/Details/Simone--Caron/I2C-Encoder-V2

4. Programming

The I2C Encoder V2.1 can be easily programmed with a customized firmware or with an official updated version. For programming the I2C Encoder V2.1 it's required an external programmer, such us the PICkit 3 or the PICkit 4. The I2C Encoder V2 can be flashed with the FW of the I2C Encoder V2.1.

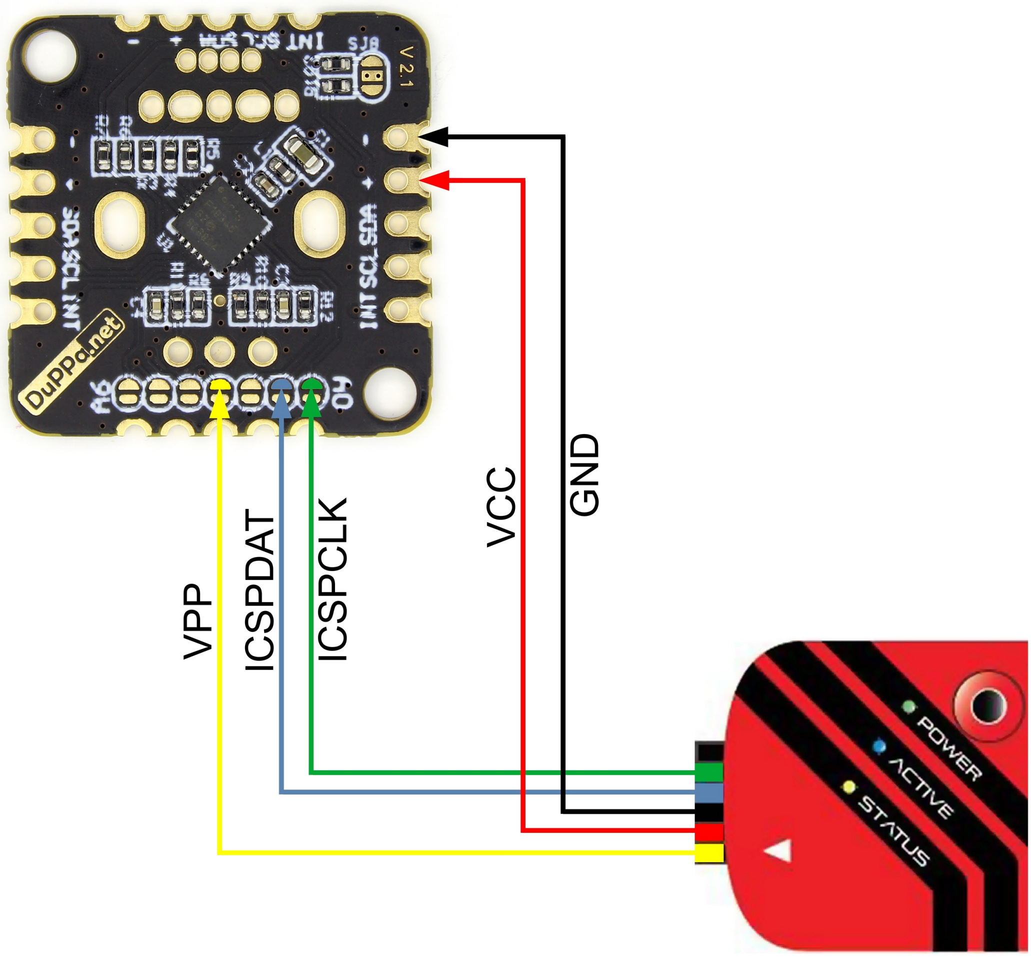


Figure 4.1: Schematic for programming with the PICkit 3

5. Changelog

In this chapter there are the hardware and firmware changes between the board version.

I2C Encoder V2.1:

Hardware changes:

- Added the jumper SJ8 for enable the pull-up resistor in the I^2C bus.
- R1 and R2 are already soldered on the board with a 0402 footprint.
- Changed the value of R3 and R4 with 42.2Ω.
- Changed the value of R5 with 107Ω.

Firmware changes:

- Added the gamma correction feature.
- Added the unique code and the version registers.
- Added the possibility to enable/disable the I^2C clock stretch.
- Added the possibility to count absolute or relative steps.
- Added and extra delay after the detection of the double-push.
- Changed the PWM frequency from 125kHz to 31,25kHz
- Decreased of the ∼33% the power consumption.
- Changed the default value of ISTEP from 1 to 0.

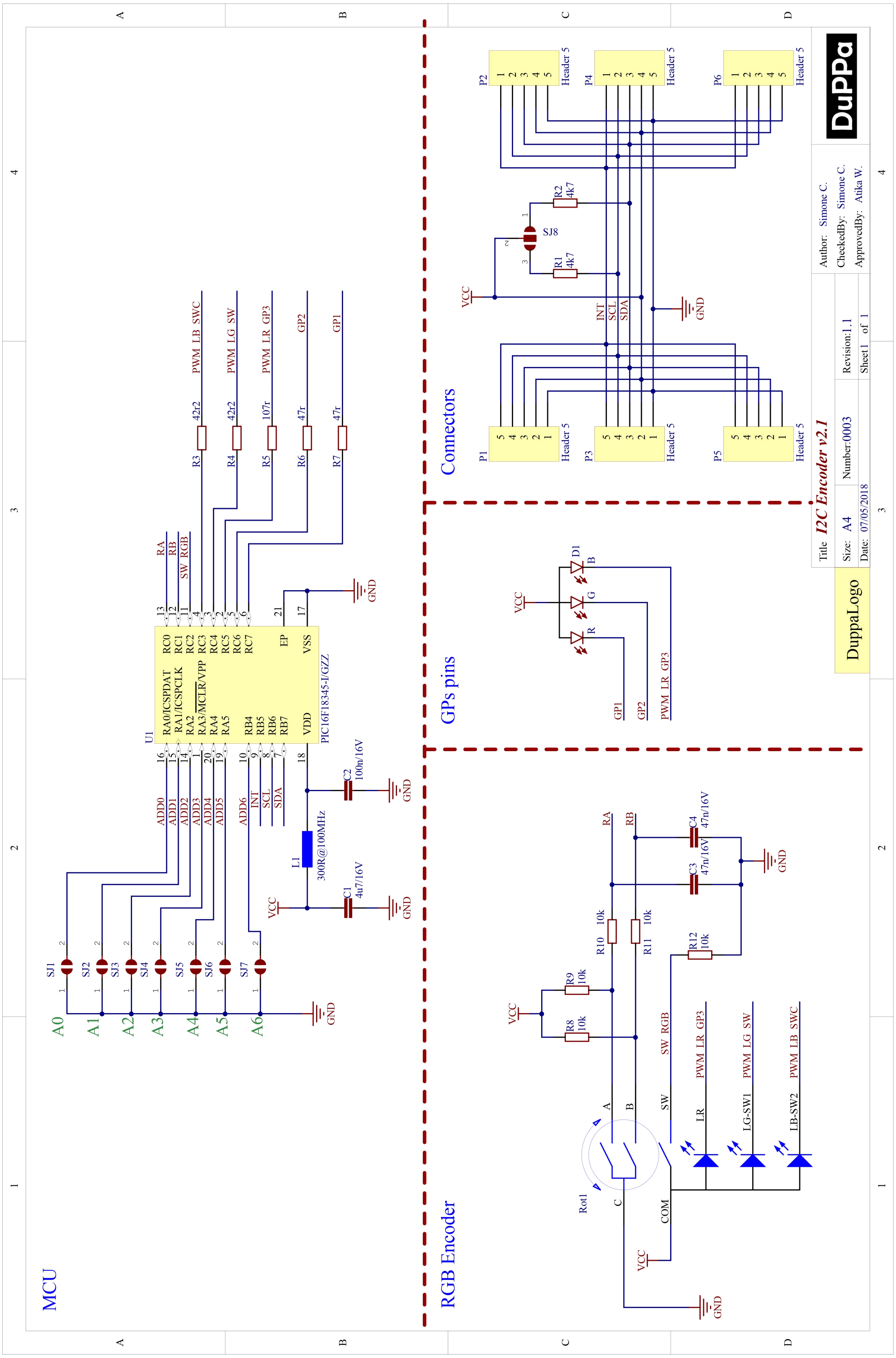
I2C Encoder V2:

First version of the I2C Encoder V2.

6. Issues

In this section, there are listed the known bugs of the I2C Encoder V2.

7. Schematic


MCU
Connectors
GPs pins
RGB Encoder
PIC16F18345-I/GZZ
Title I2C Encoder v2.1
Size: A4
Number:0003
Revision:1.1
Date: 07/05/2018
Sheet1 of 1
Author: Simone C.
CheckedBy: Simone C.
ApprovedBy: Atika W.
DuppaLogo
Header 5
VCC
GND
A0
A1
A2
A3
A4
A5
A6
PWM LB SWC
PWM LG SW
PWM LR GP3
GP2
GP1
300R@100MHz
100n/16V
4u7/16V
47n/16V
Rot1
COM
LR
LG-SW1
LB-SW2
SW RGB