FMod-I2C5LedDriver SLP 58/0.5

User's Manual

Version 1.1

five co

innovative engineering

Version: **1.1**
Last revision: **10.01.2018**
Printed in **Switzerland**

© Copyright 2002-2018 FiveCo Sàrl. All rights reserved.
The contents of this manual may be modified by FiveCo without any warning.

Trademarks
Windows® is a registered trademark of Microsoft Corporation.
Ethernet® is a registered trademark of Xerox Corporation.
Java® is a registered trademark of Sun Microsystem.

Warning
This device is not intended to be used in medical, life-support or space products.

Any failure of this device that may cause serious consequences should be prevented through the implementation of backup systems. The user agrees that protection against consequences resulting from device system failure is the user's responsibility. Changes or modifications to this device not explicitly approved by FiveCo will void the user's authority to operate this device.

Support
Web page: http://www.fiveco.ch/leds-drivers-products.html
e-mail: support@fiveco.ch

Revision history

Revision	Date	Author	Note	Firmware version		
1.0	10.04.2017	NS	- First version	Since v1.0		
1.1	10.01.2017	PHD	- Hardware spec - Detailed description - Registers organisation	v2.6		

Table of Contents

1. Overview

The FMod-I2C5LedDriver SLP 58/0.5 is a very small control device for LEDs lightning. It is particularly interesting because of its small size, the quality of its regulation and the power that can be delivered (up to 140 W continuous).

The driver receives consigns, thanks to its I2C bus, to regulate the 5 different currents of its 5 LEDs outputs. For each output, the user specifies an independent goal current and a speed at which the driver will get to the goal. This allows the user to have a full control of a lighting system, easily configurable and with an automatic ultra-low power mode when all the LEDs are switched-off. In standby and at ambient temperature, the device consumption is maximum of 50nA on logic 5V and 100nA on power supply.

This SLP daughter board ("SLP" stands for Soldered Low Power) can easily be soldered directly to a motherboard without any cables through its 32 + 2 plated holes on board edge (1.27mm spacing). In addition to that, 1.27 mm spacing male connector can be soldered to the board, making it plugable, horizontally or vertically, to a dedicated motherboard.

Up to 112 devices can be connected to the same I2C bus in daisychain configuration. Access through I^2C can be made by the user or with the help of another FiveCo device, FMod-TCP DB or FMod-TCP Box 2 which are bridges between TCP/IP and I^2C. Refer to the chapter "*I2C Interface*" to have more information on how to communicate with the board.

2. Hardware Specifications

Operating conditions

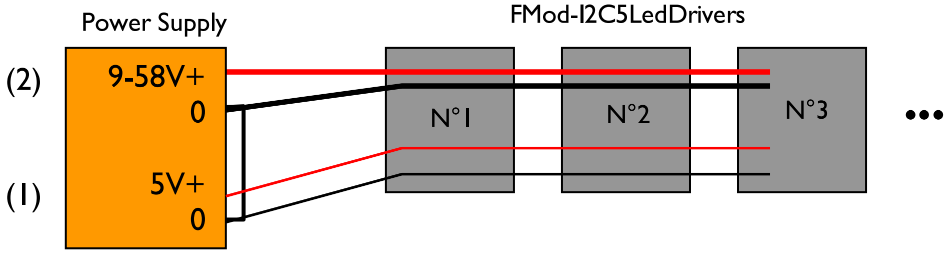
LED power	
Supply voltage	9 - 58 VDC
Supply current[1]	2.6 A max (∑ LED channel currents + 100mA)
Logic power	
Supply voltage	5 VDC
Supply current[2] (*OPTIONS.0*= 0) (*OPTIONS.0* =1)	 10 mA max 5 mA max
Standby consumption	
Logic current Power current	< 50 nA @5V, 25°C < 100nA @58V, 25°C
Input/Output	
Inputs for general purpose	5 A/D (5V)
Output LED channels	5 (LED1, LED2, LED3, LED4, LED5)
Output LED voltage (each channel)	< (LED power supply voltage -2V)
Output LED current (each channel)	10-500mA ±5mA
Other	
LED driver switching frequency	600 kHz

[1] Consumption (LED Power 9-58V) of driver electronics when all output LED channels are enable :
< 12 mA @12 V
< 31 mA @58 V
[2] Consumption (logic power 5V) when no load is connected to the ADEN pins

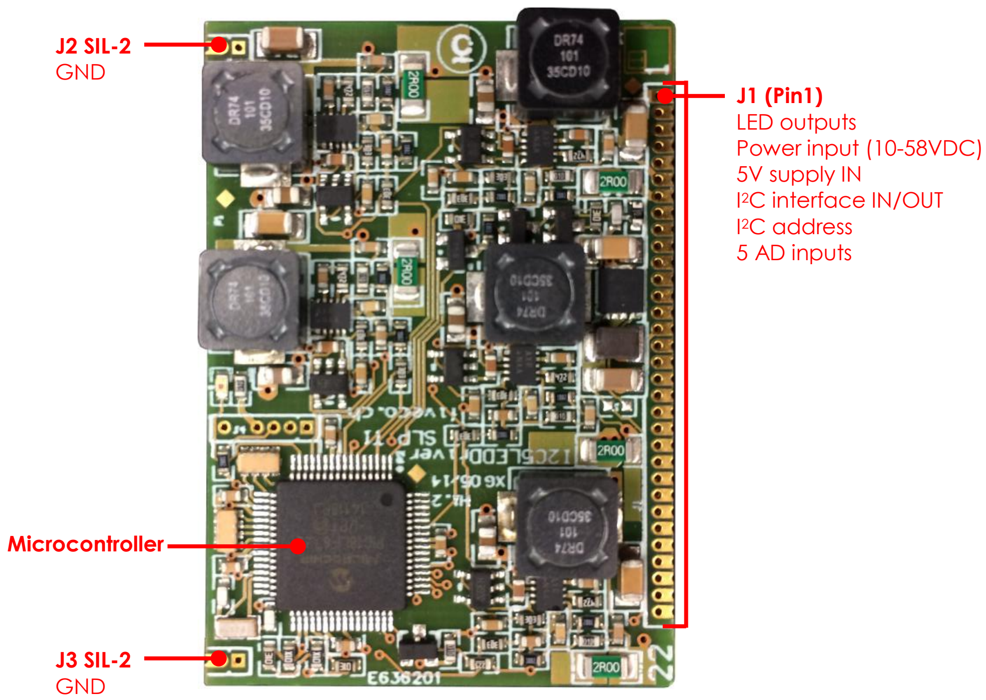
Power supply

1. Logic power: VCC (+5V) used for the electronics and processor must be applied on the Logic 5Vpins. (See details on section "Pin description")
2. LEDs power 9-58 V with the peak current minimal of ∑ LED channel currents + 100mA (2.6 A max) must be applied on the LEDs power pins.

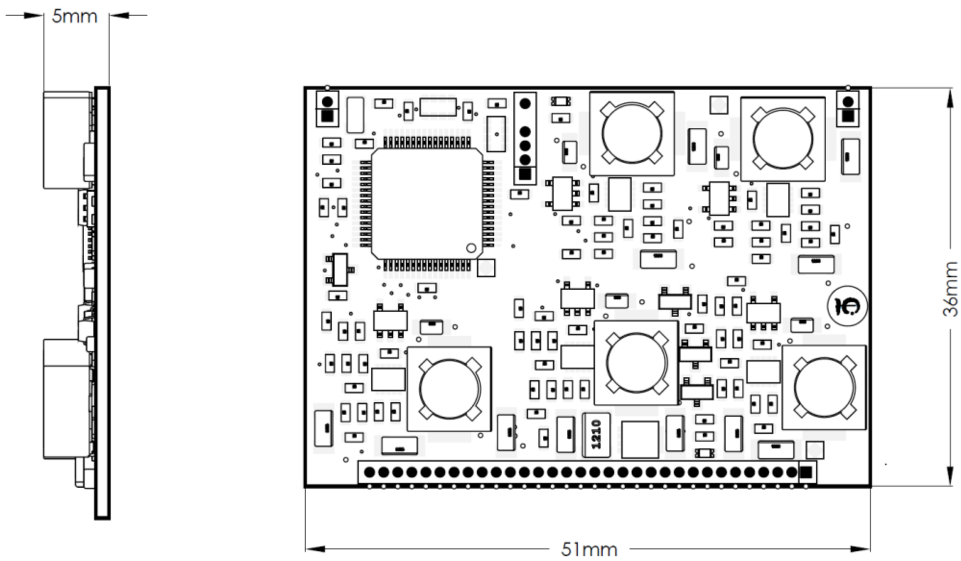
!!! Both power supplies must be connected to the same ground. !!!



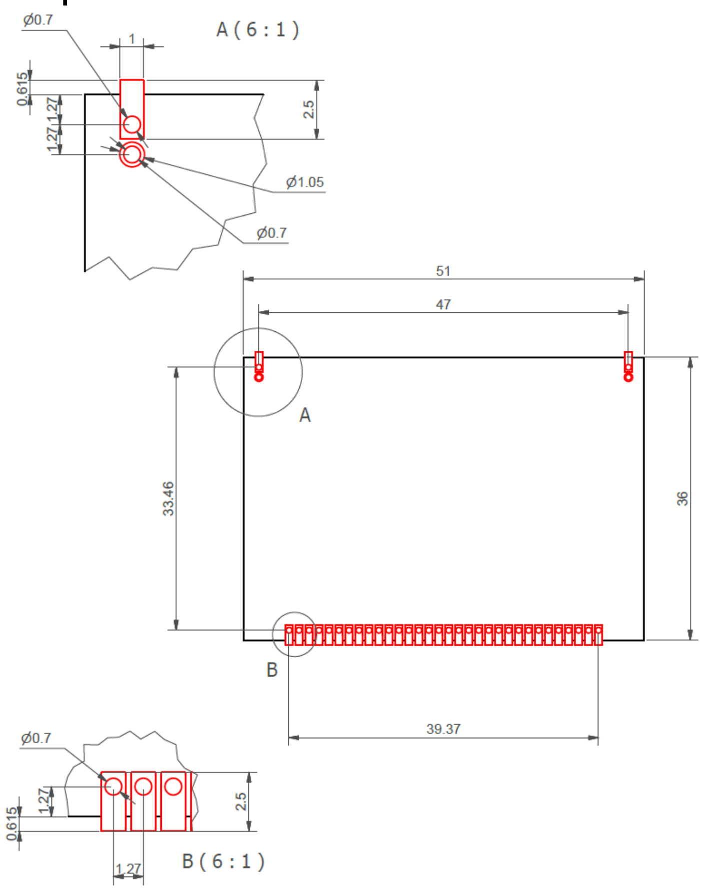
Hardware descritpion



Physical Dimensions [mm]



Footprint



Holes (ø0.7) are optional and only needed if you want to use 1.27mm inline pin connectors.

Dimensions in mm

<u>Pin description</u>

CONNECTOR J1				
PIN	Function		PIN	Function
1	GND (0V)		17	I^2C-SCL
2	LED5 +		18	I^2C-SDA
3	LED5 –		19	I^2C-Add0
4	LED4 +		20	I^2C-Add1
5	LED4 –		21	I^2C-Add2
6	LED3 +		22	I^2C-Add3
7	LED3 –		23	I^2C-Add4
8	LED2 +		24	I^2C-Add5
9	LED2 –		25	I^2C-Add6
10	LED1 +		26	AD1
11	LED1 –		27	AD2
12	PWR (10-58VDC)		28	AD3
13	PWR (10-58VDC)		29	AD4
14	GND (0V)		30	AD5
15	GND (0V)		31	ADEN (out)
16	+5V0 (in)		32	GND (0V)

Notes:

- The device's logic power supply (electronics and processor) is on pin 16 (Logic +5V) and must be applied externally. A second power supply (+9-58V) for the LEDs must be connected to pins 12 and 13.
- All GND pins must be connected together (ideally to a ground plan) on the mother board.
- The power supply for A/D inputs (ADEN) is on pin 31 and is off in ultra-low power mode allowing power saving. So, electronics used to produce AD signal is automatically shutdown when AD that are powered by ADEN are unused.

Warning

During the installation process, it is important to make sure that the output lines are not crossed (example: a LED being plugged to LED1+ and LED2-). This could permanently damage the LEDs as well as the module itself.

Under 5.5 V on PWR pins, the driver is automatically switched off.

3. Detailed description

Overall system

The FMod-I2C5LedDriver SLP 58/0.5 is composed of a microcontroller, 5 LED driver modules, a temperature sensor and a voltage sensor. Commands and settings are sended to the microprocessor throw the I^{2}C interface. The microntroller process the data and generates endpoints (in the form of a PWM) for the 5 independent LED driver modules. These will drive LED outputs in order to achieve the desired LED currents.

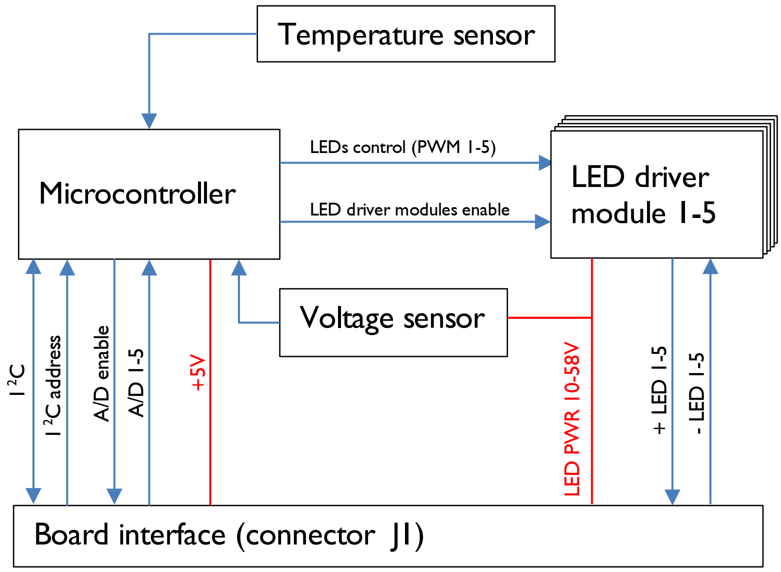


Figure 1 - Bloc diagram of the FMod-I2C5LedDriver SLP

Typical application

Each of the 5 LED outputs drives up to 500mA. The current can be distributed in several branches of LEDs. In this case, an additional resistor is mandatory on each branch to get a proper distribution of the current. The number of LEDs per branch is limited by the maximal forward voltage of the LED (this value is usually found on datasheets of the LEDs) and the maximal LED driver output voltage (LED driver input power – 2V). The ADEN pin can power analogic system (potentiometer, switch, sensor,...) at 5V and 24mA max. Acquisition is done independently on each AD inputs.

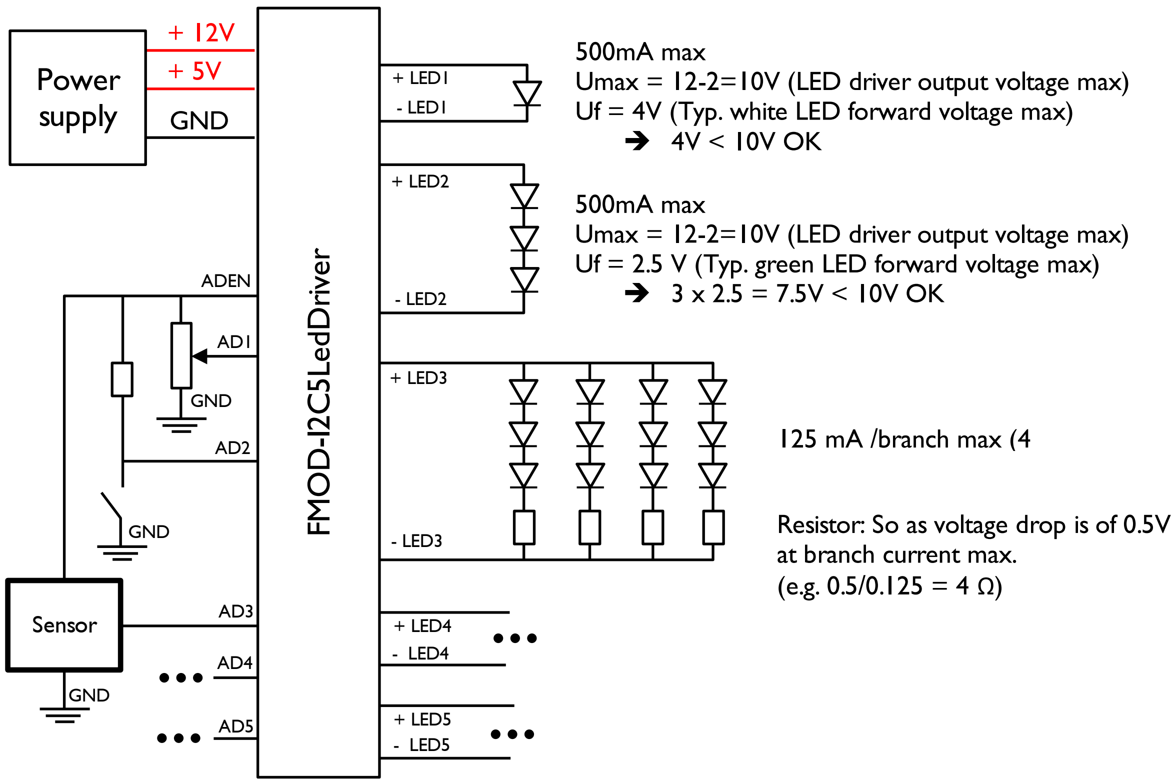


Figure 2 - Typical application of the FMod-I2C5LedDriver SLP.

How to use the driver?

The LED driver is very easy to use since it needs only two steps to work.

1. Configure settings for the current application.
2. Send commands to the driver.

(3.) Monitor the driver.

Settings can be saved so as they are automatically restored when the driver reboots (e.g. if power is cut off). Moreover, the driver will automatically goes in low power mode when all LEDs are turned off.

1. Configure settings for the current application

1. First, write in DRVVOLTAGEMIN the minimal input voltage above which the driver will operate. If you don't mind about this parameter, you can leave the default value: 0x00058000 (5.5V).
2. Then, the maximal LED output current has to be defined for each output. For example, if the maximal current must not exceed 40mA on the LED2 output, write 0x0A3D in LED2CURRENTMAX. (0.04x65'536=2621.44 = 0x0A3D).

3. Finnally, you can save settings thanks to the SAVEUSERPARAMETERS function. Once it is done, settings will automatically be restored at startup. (See Section 5 for more detail)

2. Send commands to the driver

Now you can start to use the driver by sending I2C commands.

Example

If you want to create a smooth oscillation of light on LED2 output (as described in Figure 3) you have to follow the sequence shown in Table 1.

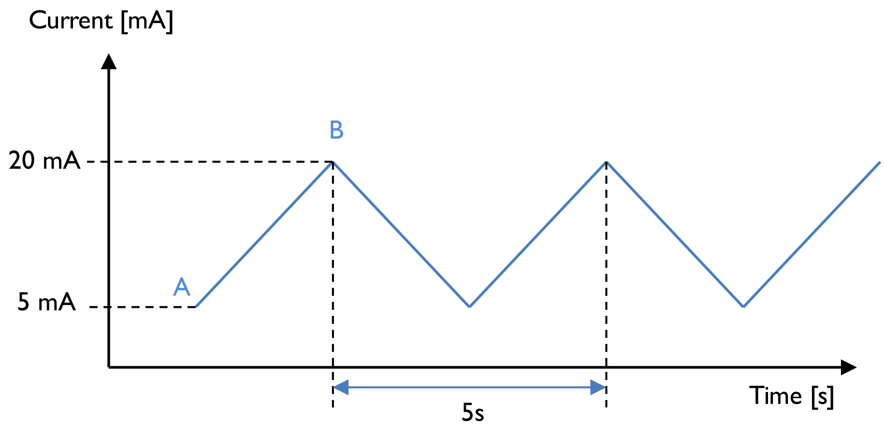


Figure 3 - LED oscillation example.

STEP	ACTION
1[3]	Write 0x2000FFFF in *LED2GOAL*
2	Write 0x8000000A in *LED2GOAL*
3	Wait 2.5 s
4	Write 0x2000000A in *LED2GOAL*
5	Wait 2.5 s
6	Write 0x8000000A in *LED2GOAL*
…	…

Table 1 – LED osciallation sequence example (LED2CURRENTMAX = 0x0A3D).

[3] This command makes sur that the sequence starts at the good point. Whitout it, you could have a problem if there is not enough time to reach the first goal of the sequence (if previous goal was 0xFFFFFFFF by example)

Explanation:

In this example, LED2CURRENTMAX = 0x0A3D (40 mA).

The LED2 output is controlled by writing on the LED2GOAL register.

LED2GOAL.[00-15]: *SPEED2* (% per milliseconde)
LED2GOAL.[16-31]: *LUM2* (%)

Goal luminosity is simply the ratio between the desired current and the max current:

LUM2_A = 5 / 40 = 12.5% = 0x2000 (0.125 x 65'536 =8'192)
LUM2_B =20 / 40 = 50% = 0x8000 (0.5 x 65'536 = 32'768)

Goal speed is an amount of percent per millisecond. In this case, we want 50%-12.5% =37.5% in 2.5 seconds. The desired speed is therefore 0.375 / 2.5s = 0.15 /s = 0.00015 /ms.

SPEED2 = 0x000A = 10 (0.00015 x 65'536 = 9.8304)

Note:

The driver goes automatically in ultra-low power mode when all LEDs are turned-off. (e.g. If you write 0x0000FFFF in LEDxGOAL*)*.

(3) Monitor the driver

Several parameters can be read during operation of the driver.

- You can verify that everything works well thanks to the WARNING register.
- You can read the input voltage thanks to VOLTAGE register.
- You can read the temperature of the board with the TEMPERATURE register.
- You can monitor the actual current sended to the output thanks to LEDxCURRENT
- registers (computed values).
- You can read values of the AD acquisition module.
- Etc.

Analog to digital input

The FMOD-I2C5LedDriver comes with 5 analog inputs. It allows conversion of an analog input signal to a corresponding 10-bit digital number. The result is stored and left justified on a 2 Bytes register: *IOxAD*

IOxAD Max	0xFFC0 = 65'472
IOxAD Min	0x0000 = 0
IOxAD Step	0x0040 = 64

All AD measurement are refreshed 142 times per second (every 7ms)

<u>Example1</u>

A potentiometer is connected to the module as shown in Figure 4

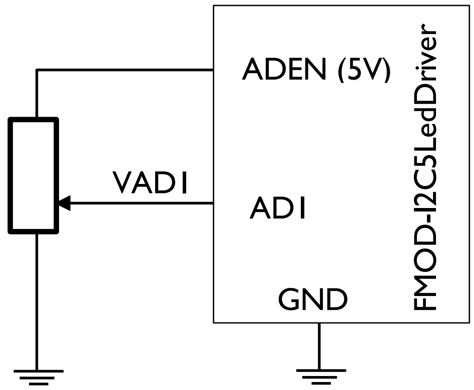


Figure 4 - Analog to digital module : example 1

- If *IO1AD* = 6'528, it means that VADI = 0.498V because (6'528 / 65'536) x 5V = 0.498V

  ➔ The slide pot is therefore at 10% of the range (0.498 / 5)

<u>Example 2</u>

A temperature sensor is connected to the module as shown in Figure 5

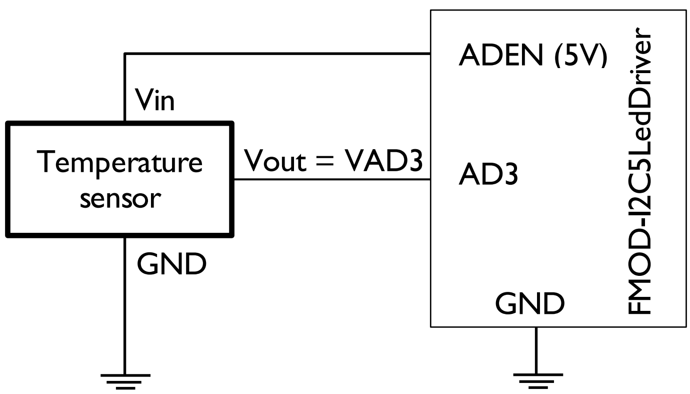


Figure 5 – Analog to digital module: example 2

Sensor's caracteristics: Vout [mV]=(10 [mV/°C] x Temp [°C]) + 500 [mV]

- If *IO3AD*= 6'528, it means that VAD3 = 0.498V, because (6'528 / 65'536) x 5V = 0.498 V

  ➔ Temperature is therefore : (498-500) / 10 = -0.2°C

- If *IO3AD* =49'152, it means that VAD3 = 0.75 V, because (49'152 /65'536) x 5V = 0.75 V

  ➔ Temperature is therefore : (750-500) / 10 = 25°C

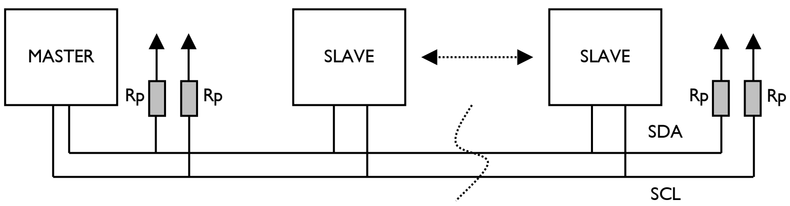
4. I2C Interface

Description

Registers are written to and read from the devices through the I2C bus; the controller is an I2C slave device. It is controlled by the I2C clock (SCL), which is driven by the I2C master. Data is transferred into and out of the cards through the I2C data (SDA) line. Either the slave or master device can pull the SDA line down; the I2C protocol determines which device is allowed to pull the SDA line down at any given time.

The maximum speed of the fast I2C interface is **400 kHz.**

WARNING:
A **pull up resistor Rp** (off-card) has to be placed between SDA and VCC (+5V) and between SCL and VCC. This is usually done at the beginning (near the master) and at the end (near the last device of the daisy chain) of the SDA and SCL lines.



(Rp = 4.7kΩ)

Protocol

The I2C bus defines several different transmission codes, as follows:

- a start bit
- the slave device 8-bit address
- an (no) acknowledge bit
- an 8-bit message
- a stop bit

Sequence

A typical read or write sequence begins by the master sending a start bit. After the start bit, the master sends the slave device's 8-bit address. The last bit of the address determines if the request will be a read or a write, where a '0' indicates a write and a '1' indicates a read. The slave device acknowledges its address by sending an acknowledge bit back to the master. If the request was a write, the master then transfers the 8-bit register address to which a write should take place. The slave sends an acknowledge bit to indicate that the register address has been received. The master then transfers the data 8 bits at a time, with the slave sending an acknowledge bit after each 8 bits.

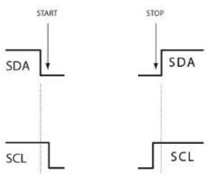
The LED driver use data length of 1 byte up to 16 bytes for their internal registers. The master stops writing by sending a restart or stop bit.
A typical read sequence is executed as follows:
First the master sends the write-mode slave address and 8-bit register address just as in the write request. The master then sends a (re)start bit and the read-mode slave address. It clocks out the register data 8 bits at a time. The master sends an acknowledge bit after each 8-bit transfer. The data transfer is stopped when the master sends a no-acknowledge bit.

Bus Idle State

The bus is idle when both the data and clock lines are HIGH. Control of the bus is initiated with a Start bit, and the bus is released with a Stop bit. Only the master can generate the start and stop bits.

Start Bit and Stop Bit

The start bit is defined as a HIGH to LOW transition of the data line while the clock line is HIGH. The stop bit is defined as a LOW to HIGH transition of the data line while the clock line is HIGH.



Start & Stop sequence

I2C address selection

On each controller with I2C interface, its 7bits of I2C address must be defined in hardware. Each line needs to be connected to +5v logic (1) or logic ground (0). Do not leave any pin of address floating!

As example, if you want to set one FMod-I2C5LedDriver SLP to address 40 (0x28), convert 40 (decimal) in binary code (b'00101000'), the 7 least significant bits (→0101000) are the values to be set to the corresponding pins of J1 "I2C Address selection →6-0".

Slave AddressWrite/Read

The 8-bit address of an I2C device consists of 7 bits of address and 1 bit of direction. A '0' in the LSB of the address indicates write-mode, and a '1' indicates read-mode.

If we want to do a **write-sequence** to the address 0x55**,** the AdressWrite is :

AdressWrite = 0xAA [i2c address (0x55)<<1 + direction bit (0)]

and if we want to do **a read-sequence**, the address used is:

AdressRead = 0xAB [i2c address (0x55)<<1 + direction bit (1)]

The **I2C address** of the device is the one that is hard-coded through the 7 address pins : [I2C Address selection bit 0,1,2,3,4,5,6,7 (in)]. With 7bits of address, 128 values are possible, but the first (8) ones (0x00-0x07) and last (8) ones (0x78-0x7F) are reserved for I2C protocol specific actions, these values must not be used as an I2C address. If the device is set with one wrong (reserved) address, it will take its default address 0x55 (85decimal, binary'1010101').

Therefore you can define the address you want between 0x08 to 0x77 (decimal 8-119).

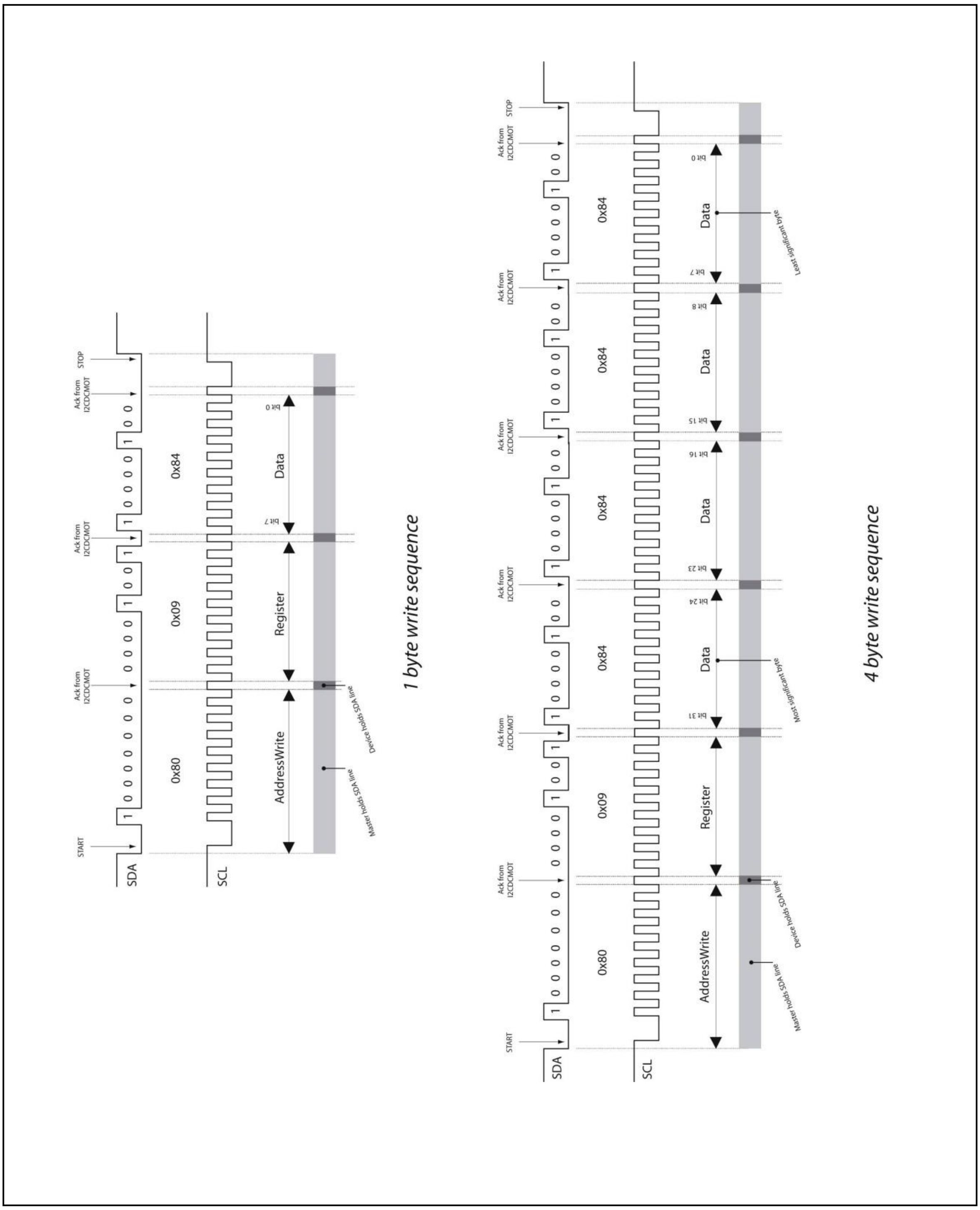
Data Bit Transfer

One data bit is transferred during each clock pulse. The I2C clock pulse is provided by the master. The data must be stable during the HIGH period of the I2C clock. It can change only when the I2C clock is LOW. Data is transferred 8 bits at a time, followed by an acknowledge bit.

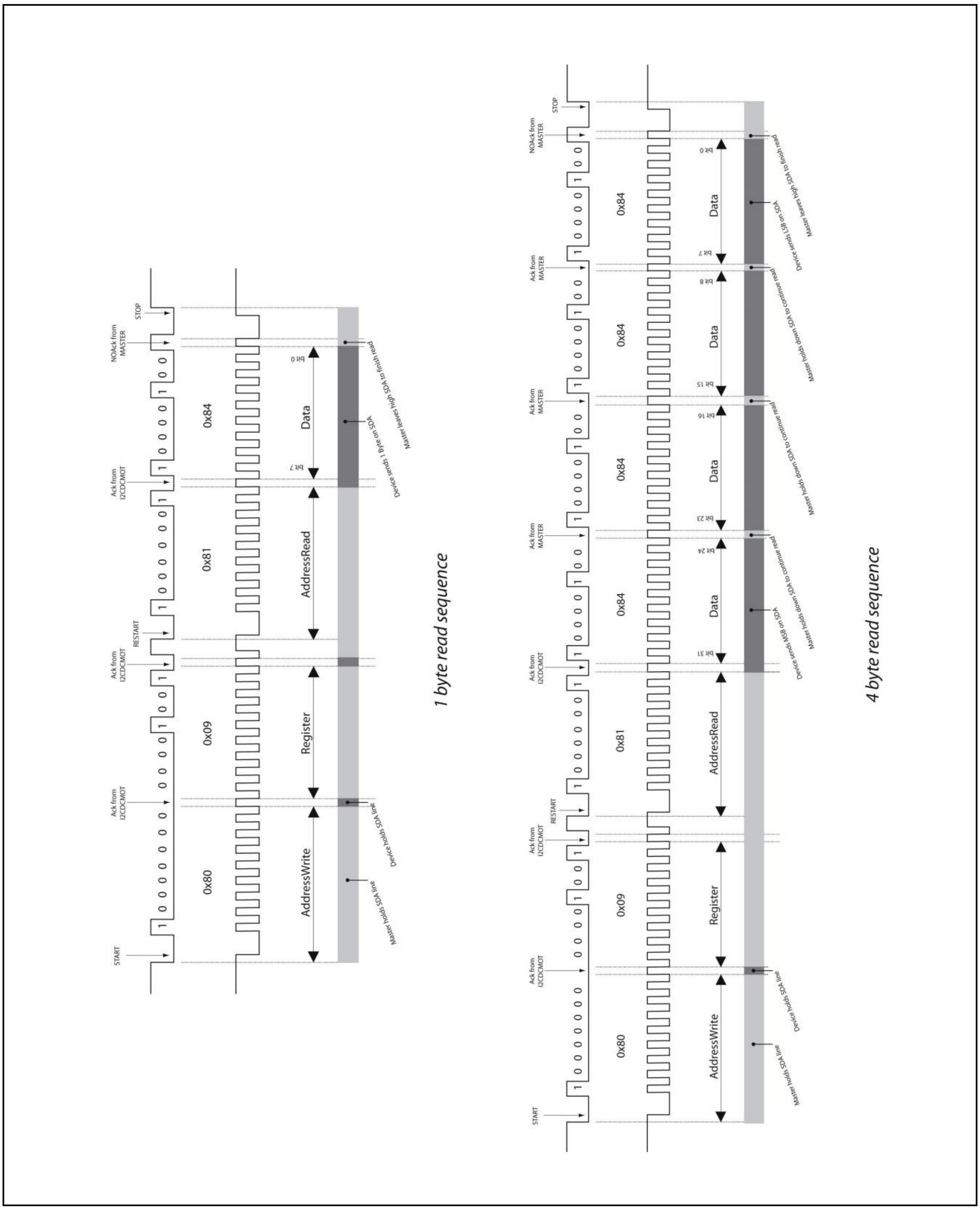
Acknowledge and No-Acknowledge Bit

The master generates the acknowledge clock pulse. The transmitter (which is the master when writing, and the slave when reading) releases the data line, and the receiver indicates an acknowledge bit by pulling the data line low during the acknowledge clock pulse. The no-acknowledge bit is generated when the data line is not pulled down by the receiver during the acknowledge clock pulse. A no-acknowledge bit is used to terminate a read sequence.

Write Sequence (1 byte and 4 bytes)



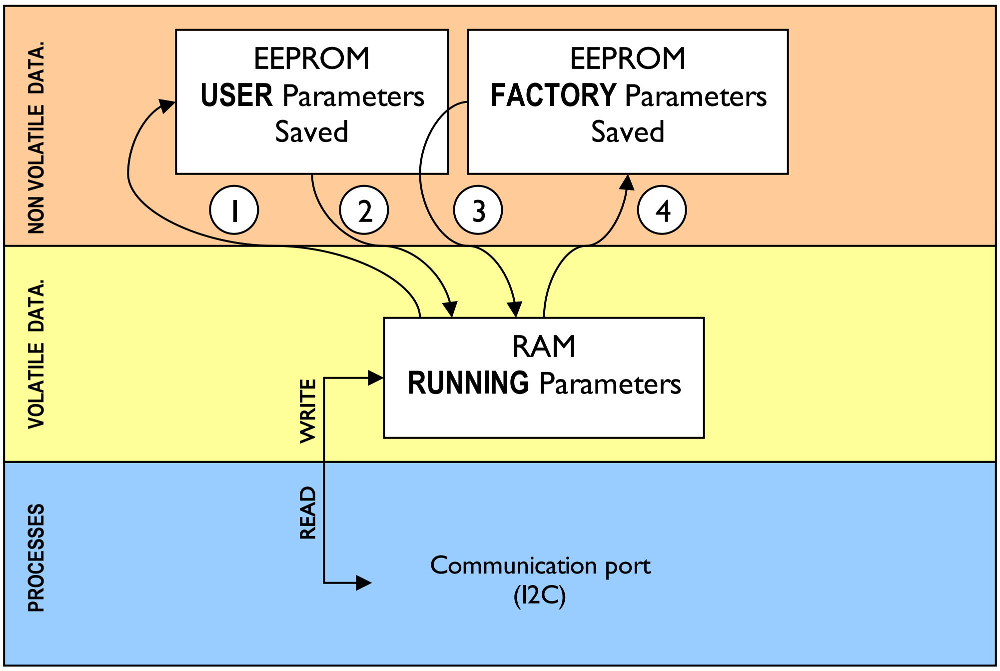
Read Sequence (1 byte and 4 bytes)



5. Register management

Memory organization

The user needs to know that a new register value sent through the communication port is loaded to the running parameters in RAM and used for the current process. All these parameters are lost at power-down. It is necessary to save them to "User Parameters" or "Factory Parameters" with the corresponding function.



Action number and description:

1	*SAVEUSERPARAMETERS* (0x03) function
2	During standard power-up or calling *RESTOREUSERPARAMETERS* (0x04) function
3	*RESTOREFACTORYPARAMETERS* (0x05) function
4 + 1	*SAVEFACTORYPARAMETERS* (0x06) function [for integrator engineers only]

Full description of registers

List of registers

Address		Bytes	Name	#Page
LED driver (continued)				
0x3F	(63)	0 (fct)	*AUTOTESTLEDS*	47
0x40	(64)	2	*IO1AD*	48
0x41	(65)	2	*IO2AD*	48
0x42	(66)	2	*IO3AD*	48
0x43	(67)	2	*IO4AD*	48
0x44	(68)	2	*IO5AD*	48

TYPE

Register Address	Register Name	Function	Read/Write control
0x00	*TYPE*	Product ID	Read only

Register Size	Register structure	
4 Bytes	Unsigned Int 16 bits (HH-HL) *TYPE*	Unsigned Int 16 bits (LH-LL) *MODEL*

Description:
Product identifier composed of a *Type* and *Model* number.
Defines the type of peripheral.
Normally different *TYPE* modules are not software compatible.

Example:
Device with *TYPE* = 0x002A0001 means *Type*=2A (42 = I2C Led Driver), *Model* = 1.

Limits:
None

Active:
Each time the processor is running.

VERSION

Register Address	Register Name	Function	Read/Write control
0x01	*VERSION*	Software ID	Read only

Register Size	Register structure	
4 Bytes	2bytes Hardware version (HH-HL)	2 bytes Firmware version (LH-LL)

Description:
Hardware identifier composed of a *Version* and *Revision* number.

Firmware identifier composed of a *Version* and *Revision* number.
Normally same *Version* with different *Revision* is backward compatible.

Example:
VERSION 0x0108050E
Hardware = 0x0108 = Version 1.8
Firmware = 0x050E = Version 5.14

Firmware 5.14 = *Version* 5, *Revision* 14 (0x0E) is compatible with all earlier revisions of the same version (ver 5.0 to 5.14) but has new functionalities (deactivated by default) or code optimizations.

Limits:
None

Active:
Each time the processor is running.

RESET CPU

Function Address	Function Name	Function	Read/Write control
0x02	*RESETCPU*	Restart processor	Write only

Register Size	Register structure	Unit
0 Byte	none	none

Description:
Reboots the card. Communication will be lost.

Active:
Each time the processor is running.

SAVE USER PARAMETERS

Function Address	Function Name	Function	Read/Write control
0x03	*SAVEUSERPARAMETERS*	Saves all in EEPROM	Write only

Register Size	Register structure	Unit
0 Byte	none	none

Description:
Saves the following parameters to EEPROM user space:

0x10 *COMOPTIONS*
0x12 *I2CADDRESS*
0x15 *DEVICENAME*

0x21 *OPTIONS*
0x22 *DRVVOLTAGEMIN*

0x30 *LED1CURRENTMAX*
0x31 LED2CURRENTMAX
0x32 LED3CURRENTMAX
0x33 LED4CURRENTMAX
0x34 LED5CURRENTMAX

Active:
Each time the processor is running.
Do not change any of these parameters while saving!

RESTORE USER PARAMETERS

Function Address	Function Name	Function	Read/Write control
0x04	*RESTOREUSERPARAMETERS*	Restores saved values	Write only

Register Size	Register Structure	Unit
0 Byte	none	none

Description:
Restores the user parameters from EEPROM.

See *SAVEUSERPARAMETERS* (0x03) register list.

Active:
Each time the processor is running.

RESTORE FACTORY PARAMETERS

Function Address	Function Name	Function	Read/Write control
0x05	*RESTOREFACTORYPARAMETERS*	Factory default	Write only

Register Size	Register Structure	Unit
0 Byte	none	none

Description:
Restores factory parameters.

See *SAVEUSERPARAMETERS* (0x03) register list.

Active:
Each time the processor is running,
SAVEUSERPARAMETERS has to be run after setting this function so that the next reboot will retain the same parameters.

SAVE FACTORY PARAMETERS

Function Address	Function Name	Function	Read/Write control
0x06	*SAVEFACTORYPARAMETERS*	Saves factory default	Write only

Register Size	Register Structure	Unit
0 Byte	none	none

Description:
This function is reserved for integrator engineers, and not for the end user.
Used when all parameters have been approved for an application. It saves in EEPROM all configurable registers for both factory parameters and user parameters.
This function also call the *SAVEUSERPARAMETERS* function.

See *SAVEUSERPARAMETERS* (0x03) register list.

Active:
Each time the processor is running.
Do not change any of these parameters while saving!

VOLTAGE

Register Address	Register Name	Function	Read/Write Control
0x07	*VOLTAGE*	Power module voltage	Read only

Register Size	Register Structure	Unit
4 Bytes	Signed (2's cplt) Int 16 (HH-HL) +16 bits fixed point (LH-LL)	Volt

Description:
Input voltage

Limits:
Max 0x7FFFFFFxx = 32'767.996
Min 0x000000xx = 0.0
Step 0x000001xx = 0.004

Example:
When read 0x00234567 = 2'311'527, Voltage = 35.27 (2'311'527/65'536)

Active:
Each time the processor is running.

WARNING

Register address	Register Name	Function	Read/Write Control
0x08	*WARNING*	Bit to bit state	R/W

Register Size	Register Structure	Unit
4 Bytes	Unsigned Int 32 bits , each bit independent	none

Description:

Each information/warning/error is made up of 2 bits: the first one shows the current state, the next one shows whether this state has appeared previously.

Only the bits that show the past states can be cleared by writing 0x00000000 to the *WARNING* register.

Bits when set

Warnings.0 Under-voltage of the power input.
Warnings.1 Previously active, it can be cleared by user.

Warnings.2 Over-voltage of the power input.
Warnings.3 Previously active, it can be cleared by user.

Warnings.4 Over-temperature state, reduce the output current to 75% of its value when temperature > 115°C, 50% when > 120°C, 25% when > 125°C, 0% when > 130°C.
Warnings.5 Previously active, it can be cleared by user.

Warnings. 6-31 Reserved

Default: bits 31 -> 0

0x00 00 00 00

Active:

Each time the processor is running.

NB POWER UP

Register address	Register Name	Function	Read/Write Control
0x0B (11)	*NBPOWERUP*	Number of power up in device's life	Read only

Register Size	Register Structure	Unit
4 Bytes	Unsigned Int 32 bits	none

Description:
The number of power up is incremented each time the controller's power supply and the logic 5V are in the specifications range.

Limits:
Min 0x00 00 00 00 = 0
Max 0xFF FF FF FF = 4'294'967'295

Active:
Each time the processor is running

TIME IN SERVICE

Register address	Register Name	Function	Read/Write Control
0x0C	*TIMEINSERVICE*	Time in service in device's life	Read only

Register Size	Register Structure	Unit
4 Bytes	Unsigned Int 32 bits	Seconds

Description:
The *TIMEINSERVICE* register is incremented every second when the driver is not in Standby

Limits:
Min 0x00 00 00 00 = 0 seconds
Max 0xFF FF FF FF = 4'294'967'295 seconds = ~136 years

Active:
Each time the processor is running.

COM OPTIONS

Register Address	Register Name	Function	Read/Write Control
0x10 (16)	*COMOPTIONS*	Communication options	Read/Write

Register Size	Register Structure	Unit
4 Bytes	32 individual bits	none

Description:
This register is reserved for future use.

This register is saved when calling the function *SAVEUSERPARAMETERS* or *SAVEFACTORYPARAMETERS*.

I2C ADDRESS

Register Address	Register Name	Function	Read/Write Control
0x12 (18)	*I2CADDRESS*	Network ID	Read(/Write)

Register Size	Register Structure	Unit
4 Bytes	4 x Unsigned Bytes	none

Description:
Network identifier of 7bits used for I2C bus, without R/W bit.
Only the least significant byte is used for the address. Value can be [8-119], [0x08-0x77].
Since the value is Hardware coded (I^2C address lines 0-7), writing in this register will not affect the effective I^2C address.

Limits of I2C ID:
Min = 0x08 (8)
Max = 0x77 (119)
Because b'0000xxx' and b'1111xxx' are reserved for I2C specific actions.
If a wrong ID is coded, the device will automatically use its default value 0x55.

DEVICE NAME

Register Address	Register Name	Function	Read/Write Control
0x15 (21)	*DEVICENAME*	Device's ASCII name	Read/Write

Register Size	Register Structure	Unit
16 Bytes	16 (only) x Unsigned Bytes (CHAR)	none

Description:
Name and/or description of the device.

This register is saved when calling the function *SAVEUSERPARAMETERS* or *SAVEFACTORYPARAMETERS*.

Example:
For the name "Hello Module"; extend to 16 Bytes the name: "Hello Module"+5x space=16 Byte.
So write 0x48656C6C 6F204D6F 64756C65 20202020.

OPTIONS

Register Address	Register Name	Function	Read/Write Control
0x21 (33)	*OPTIONS*	Bit to bit settings	Write (Read)

Register Size	Register Structure	Unit
4 Byte	Unsigned Int 32 bits , each bit independent	none

Description:
This register is saved when calling the function *SAVEUSERPARAMETERS* or *SAVEFACTORYPARAMETERS*.

Bits	*when set*
OPTIONS.0	Enable the low power PWM mode. Usefull when available logical power is limited. The CPU speed is slowed by a factor of 4 reducing therefore power consumption (on logical 5V power). Counterpart is that the PWM frequency is also divided by 4 and induces therefore 4 times higher ripple on LED current output. Except for very special cases, this bit must stay cleared.
OPTIONS.1-31	Reserved

Limits:
None

Default:

bits 31 -> 0 : 0x00, 0x00, 0x00 , b'00000001'

Active:
Each time the processor is running.

DRIVER VOLTAGE MIN

Register Address	Register Name	Function	Read/Write Control
0x22 (34)	*DRVVOLTAGEMIN*	Voltage threshold	Write only

Register Size	Register Structure	Unit
4 Bytes	Signed (2's cplt) Int 16 (HH-HL) +16 bits fixed point (LH-LL)	Volt

Description:
Minimal input voltage

Information:
When *VOLTAGE* is smaller than *DRVVOLTAGEMIN* or 5.5V the LEDs driver is swiched off.

Limits:

Max	0x7FFFFFFxx = 32'767.996
Min	0x000000xx = 0.0
Step	0x000001xx = 0.004

Example:
When read 0x00234567 = 2'311'527, Value = 35.27 (2'311'527/65'536)

Default:
0x00058000 = 360'448, Value =5.5 V (360'448/65536)

Active:
Each time the processor is running.

TEMPERATURE

Register Address	Register Name	Function	Read/Write Control
0x23 (35)	*TEMPERATURE*	Board °C	Read only

Register Size	Register Structure	Unit
4 Bytes	Signed (2's cplt) Int 16 (HH-HL) +16 bits fixed point (LH-LL)	C°

Description:
It gives temperature of the board.
This value is used inside the device to reduce the current output at the power bridge to prevent overheating destruction.

0< T° < ~90°C	Normal temperature
90°C < T° < 115°C	Critical temperature, but functioning
115°C < T° < 120°C	Over-temperature state, reduce the output current to 75% of its value
120°C < T° < 125°C	Reduce the output current to 50% of its value
125°C < T° < 130°C	Reduce the output current to 25% of its value
130°C < T°	Reduce the output current to 0% of its value

Automatic re-enabling of the power bridge to the maximum value when T° <113°C.

Limits:
Max 0x00960000 = 150°C
Min 0xFFD80000 = -40.0 °C

Example:
Other 0x00168000 = 1474560 → 22.5°C = (1474560/65536)

Active:
Each time the processor is running.

I/O STATE

Register Address	Register Name	Function	Read/Write Control
0x24 (36)	*IOSTATE*	A/D measurements	Read only

Register Size	Register Structure	Unit
10 Bytes (5 x 2 Bytes)	Unsigned Int 16 bits Unsigned Int 16 bits Unsigned Int 16 bits Unsigned Int 16 bits Unsigned Int 16 bits	None

Description:

Result of the analog to digital convertion of all ADx input in a 10 Bytes register.

IOSTATE.[00-15] : A/D acquisition of I/O 1
IOSTATE.[16-31] : A/D acquisition of I/O 2
IOSTATE.[32-47] : A/D acquisition of I/O 3
IOSTATE.[48-63] : A/D acquisition of I/O 4
IOSTATE.[64-79] : A/D acquisition of I/O 5

For more convenience, you can also get each A/D acquisition individually by using the IOxAD registers.

Limits:
The resolution of each AD acquisition is a 10-bit digital number. The result is left justified on 2 Bytes for each conversion.

IOSTATE.[00-15], IOSTATE.[16-31], IOSTATE.[32-47], IOSTATE.[48-63], IOSTATE.[64-79]

Max 0xFFC0 = 65'472
Min 0x0000 = 0
Step 0x0040 = 64

Active:
Each time the processor is running.

LEDxCURRENTMAX

Registers Addresses	Registers Names	Function	Read/Write Control
0x30(48) 0x31(49) 0x32(50) 0x33(51) 0x34(52)	*LED1CURRENTMAX* *LED2CURRENTMAX* *LED3CURRENTMAX* *LED4CURRENTMAX* *LED5CURRENTMAX*	Defines max current	Write (Read)

Register Size	Register Structure	Unit
2 Bytes 2 Bytes 2 Bytes 2 Bytes 2 Bytes	16 bits fixed point (H-L) 16 bits fixed point (H-L) 16 bits fixed point (H-L) 16 bits fixed point (H-L) 16 bits fixed point (H-L)	Ampere

Description:
This register defines the maximal output current of LEDx when *LUMx* is set to 100%.

This register is saved when calling the function *SAVEUSERPARAMETERS* or *SAVEFACTORYPARAMETERS*.

Information:
The convertion factor between *LUMx* and *LED1CURRENT* depends on the *LED1CURRENTMAX* register.

Example:

LUM1 = 0x8000 : 50% (32'768/65'536)
LED1CURRENTMAX = 0x4000 : 250mA (16'384/65'536)

⇨ *LED1CURRENT* = 0x2000 : 125 mA (8'192/65'536)

Limits:

Max 0x8000 = 32'768 : 500 mA (32'768/65'536)
Min 0x0000 : 0.0 mA
Step 0x0001 : 0.0153 mA (1/65'536)

LED1CURRENTMAX is internaly limited to 0x8000 (500mA) if the user try to write a higher value.

Default:
0x8000 = 32'768 : Current max = 500 mA (32'768/65'536)

Active:
Each time the processor is running.

LEDxCURRENT

Register Address	Register Name	Function	Read/Write Control
0x35 (53) 0x36 (54) 0x37 (55) 0x38 (56) 0x39 (57)	*LED1CURRENT* *LED2CURRENT* *LED3CURRENT* *LED4CURRENT* *LED5CURRENT*	Current	Read only

Register Size	Register Structure	Unit
2 Bytes 2 Bytes 2 Bytes 2 Bytes 2 Bytes	16 bits fixed point (H-L) 16 bits fixed point (H-L) 16 bits fixed point (H-L) 16 bits fixed point (H-L) 16 bits fixed point (H-L)	Ampere

Description:
This register informs the user of the actual current sent to the LEDx output

Limits:

Max 0x8000 = 32‘768 : 500 mA (32‘768/65‘536)
Min 0x0000 : 0.0 mA
Step 0x0001 : 0.0153 mA (1/65‘536)

Active:
Each time the processor is running

LEDxGOAL

Register Address	Register Name	Function	Read/Write Control
0x3A (58) 0x3B (59) 0x3C (60) 0x3D (61) 0x3E (62)	*LED1GOAL* *LED2GOAL* *LED3GOAL* *LED4GOAL* *LED5GOAL*	Set LEDx goal	Write (Read)

Register Size	Register Structure	Unit
4 Byte	Int 16 + Int 16	none

Description:
This register is used by the user to controle the driver. The 2 high Bytes set the luminosity goal (in percent). The 2 low Bytes set the speed to reach the goal (in percent per milliseconde).

LUMx and *SPEEDx* :

LED1GOAL.[00-15] : *SPEED1* (% per milliseconde)
LED1GOAL.[16-31] : *LUM1* (%)

LED2GOAL.[00-15] : *SPEED2* (% per milliseconde)
LED2GOAL.[16-31] : *LUM2* (%)

LED3GOAL.[00-15] : *SPEED3* (% per milliseconde)
LED3GOAL.[16-31] : *LUM3* (%)

LED4GOAL.[00-15] : *SPEED4* (% per milliseconde)
LED4GOAL.[16-31] : *LUM4* (%)

LED5GOAL.[00-15] : *SPEED5* (% per milliseconde)
LED5GOAL.[16-31] : *LUM5* (%)

Limits:

SPEEDx
Max 0xFFFF : 100% per milliseconde (65'535/65'536)
Min 0x0000 : 0.0 % per milliseconde
Step 0x0001 : 0.0015259 % per millisecond

LUMx
Max 0xFFFF : 100% (65'535/65'536)
Min 0x0000 : 0.0 %
Step 0x0001 : 0.0015259 % (1/65'536)

<u>Default:</u>
0x0000FFFF : *LUMx* = 0%, *SPEEDx* = 100% per second

<u>Example</u>

LED1CURRENTMAX	=	0x4000 : 250mA (16'384/65'536)
LED1GOAL	=	0x20000010
		0x8000 : 50 % (32'768/65'536)
		0x0010 : 24.41 % per second (16/65536)

⇨ *LED1CURRENT* : From 0x0000 to 0x8000 : 0 to 125 mA (8'192/65'536) in 2,048 seconds

<u>Active:</u>
Each time the processor is running.

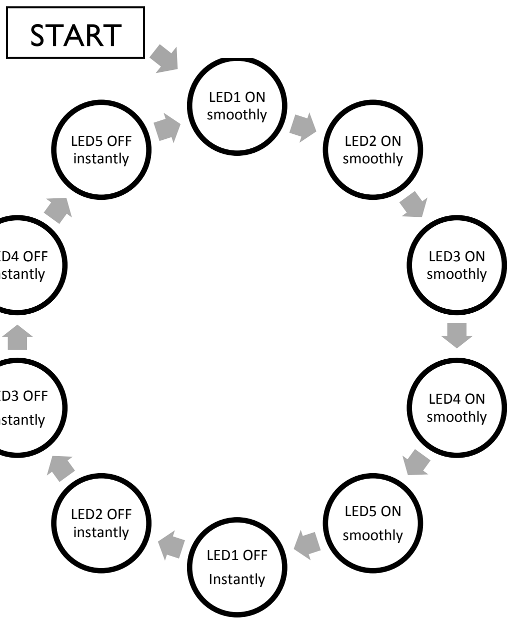
AUTO TEST LEDS

Register Address	Register Name	Function	Read/Write Control
0x3F (63)	*AUTOTESTLEDS*	Test 5 LED outputs	Write

Register Size	Register Structure	Unit
0 Byte	none	none

Description:
This function helps the user to test quickly LED outputs.

Information
The function sets LEDxGOAL each time it is called with respect to the following diagram:



LEDx ON means LEDxGOAL = 0xFFFF0005
LEDx OFF means LEDxGOAL = 0x0000FFFF

Beware:
The LEDxCURRENTMAX
registers must be configured appropriately since AUTOTESTLEDS sends maximum current to each LED output.

Active:
Each time the processor is running.

IOxAD

Register Address	Register Name	Function	Read/Write Control
0x40 (64) 0x41 (65) 0x42 (66) 0x43 (67) 0x44 (68)	*IO1AD* *IO2AD* *IO3AD* *IO4AD* *IO5AD*	A/D measurements	Read

Register Size	Register Structure	Unit
2 Byte 2 Byte 2 Byte 2 Byte 2 Byte	Unsigned Int 16 bits Unsigned Int 16 bits Unsigned Int 16 bits Unsigned Int 16 bits Unsigned Int 16 bits	none

Description:

Result of the analog to digital convertion of ADx input.

If synchronisation between the 5 A/D acquisitions is critical, use IOSTATE register instead since it guarantees that all acquisitions you read were done simultaneously (in a time window of 7ms).

Limits:
The resolution of each AD acquisition is a 10-bit digital number. The result is left justified on 2 Bytes for each ADx input

Max	0xFFC0 = 65'472
Min	0x0000 = 0
Step	0x0040 = 64

Active:
Each time the processor is running.

Contact address:

FiveCo - Innovative Engineering
En Budron H11
CH-1052 Le Mont-sur-Lausanne
Switzerland
Tel: +41 21 632 60 10
Fax: +41 21 632 60 11

www.fiveco.com
info@fiveco.com