DIAMEX DIGI-DOT BOOSTER
Drivers for WS2812 and SK6812 LEDs with SPI interface

Functions

The DIAMEX DIGI-DOT booster drives up to 256 digital light-emitting diodes of the types WS2812 (and compatible) and SK6812. This are only a few commands to the DIGI-DOT boosters which are sent via an SPI interface, thus the microcontroller-addressing is greatly relieved (such as the ATMega328 on the Arduino) and has enough time to take care of other tasks.

Technical specifications:

Connector for LEDs and Stripes with WS2812 or SK6812 compatible LEDs

Controls RGB and RGBW LEDs

Maximum: 256 LEDs

Can be easily connected to the SPI interface the Arduino or Raspberry PI.

Maximum SPI speed: 12 megabits

Easy-to-remember instruction set (programming examples available online)

Control by a powerful 32-bit ARM Cortex-M0 microcontroller

Board dimensions: approx. 21 x 18mm

Pinout

Following is a description of the connection pins of the DIGI-DOT -Boosters. All input pins can be controlled with 3.3V or 5V who-the.

| PIN	IO	Description
GND	PWR	GND
WS2812	O	Output to WS2812, SK6812 or compatible LEDs. 5V signal level.
DEBUG	O	Debug output is only required for the development of the Bios.
RESET	I	A low level sets the controller of the DIGI-DOT Boosters back
SWDIO	I/O	SWCLK is only required for the production
SWCLK	I	SWCLK is only required for the production
+3,3V	PWR	3.3V output of the voltage regulator to the LED booster board. Please do not put tension here
+5V	PWR	5V power supply for the DIGI-DOT booster. Current consumption max. 15 mA
PA2	I/O	Reserved, without function
PA3	I/O	Reserved, without function
LED	O	Control LED cathode (anode to + 3.3V, do not forget the resistor!)
CS	I	Chip-select, Low level active
MOSI	I	SPI-Bus MOSI
MISO	O	SPI-Bus MISO
SCK	I	SPI-Bus SCK
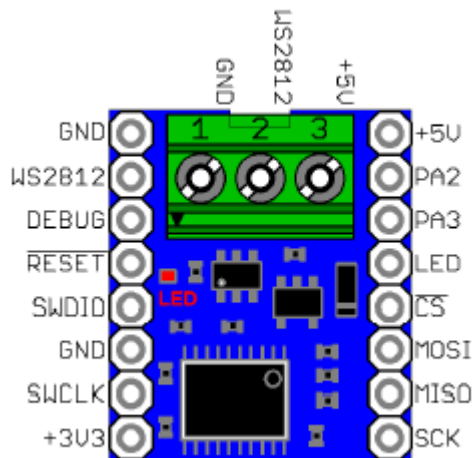
Example: connection to Arduino

Connect the DIGI-DOT Booster board with the PINs of the Arduino using the following list.

DIGI-DOT-BOOSTER PIN	ARDUINO PIN
GND	GND
+5V	+5V
RESET	9 (PB1)
CS, Chip-Select	10 (PB2)
MOSI	11 (PB3)
MISO (optional)	12 (PB4)
SCK	13 (PB5)

Even if your Arduino board provides a 3.3V voltage, please connect only 5V. This chip voltage is required for the level converter, which supplies the connected LEDs with 5V data.

The connection of the MISO line to the Arduino board is not required. Since it belongs to the SPI bus, it was located in the adjacent graph with.

Tip: It can be operated simultaneously several DIGI-DOT booster when the chip select lines are connected to various IO pins of the Arduino. MOSI, SCK and MISO optionally can be connected in parallel.
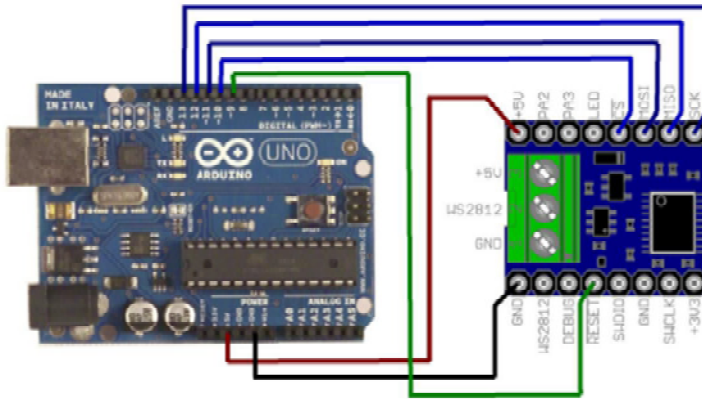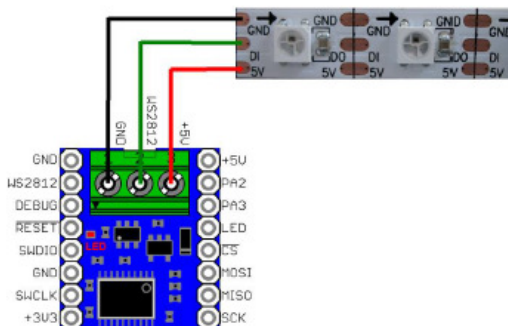
Power supply

LEDs need individually a relatively small current. This can, however, add up in long stripes or large panels to high values. As an example: the current consumption of a WS2812 LED. At 100% white this LED 60mA required for 5V. It seems not to be very much, but for 10 LEDs 600mA a required, that means that a normal USB 2.0 port is already overloaded because it can supply a maximum of 500mA normally (according to USB specification). For this reason, we always recommend the use of a LEDs designed power supply.

Option 1: supply Some LEDs on Arduino

The DIGI-DOT-booster is supplied via the 5V line of Arduino. If only a few LEDs are connected (up to 4 pieces), the power supply generally can be taken directly from the Arduino. For safety, please measure the voltage across the LEDs at 100% white. If this breaks strong, supply the LEDs please with an external 5V power supply


Option 2: supply many LEDs on power supply

The DIGI-DOT-booster is supplied via the 5V line of Arduino, the LEDs are powered by an external power supply.

In this variant may the 5V terminal on the DIGI-DOT Boosters not be connected.
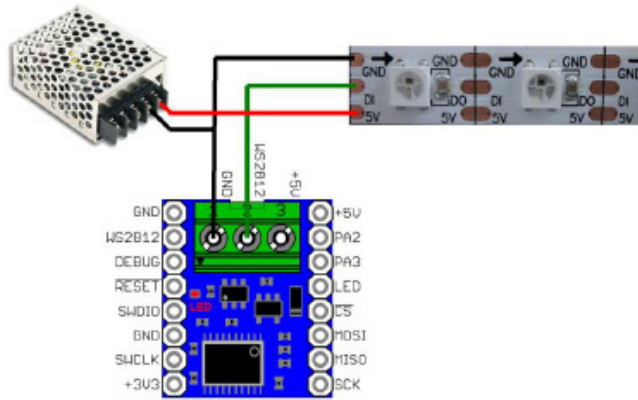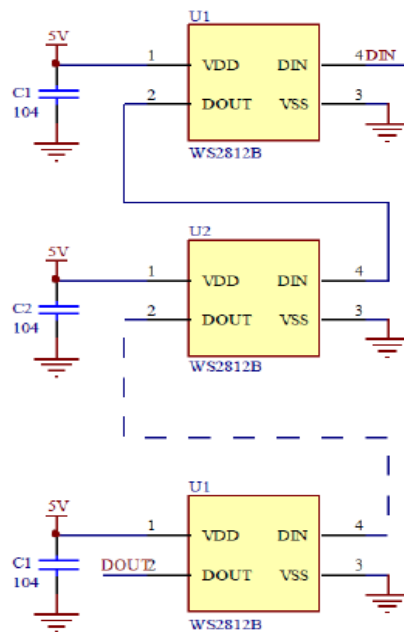
LED connection

LEDs of types WS2812B, PL9823, APA-106, SK6812 and compatible have in addition to the power supply pins (ground and + 5V) a data input (DI) and a data output (DO). If several LEDs are required, simply the data output (DO) of the first LED has to be connected to the data input (DI) of the next LED (daisy chain principle).
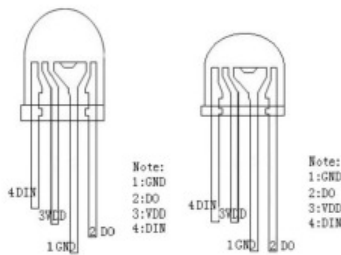
PL9823 LED

LED-Stripe mit WS2812

Number of LEDs

The DIAMEX DIGI-DOT-Booster can control a maximum of 256 LEDs. About the init command the required number (2 ... 256) is adjusted. The number must always be an even number, if 7 LEDs are connected, a number of 8 must be passed as a parameter in the init command. Since only one byte for the number of light-emitting diodes can be specified, the value 0 to the maximum number of 256 LEDS.

Arrangement of colors

The arrangement of colors may vary from manufacturer to manufacturer. In classical WS2812 LEDs arrangement green-red-blue is (GRB), this is the default setting of the DIGI-DOT Boosters. If you notice that the colors do not match, the arrangement must be changed with a command.

System-LED RED

The red LED is used to visualize the data output to the LEDs. Should you not be sure whether the connected LEDs are working, you can see from the flicker of the red LED that data to the LED are transmitted.

SPI protocol

The LED-Booster uses a standard SPI protocol 0 with CPHA = 0 and CPOL = 0th only those lines MOSI, SCK and CS are needed. Since the DIGI-DOT-Booster does not send data to the drive circuit, a connection of the MOSI line is not necessary.

The data transmission is in bytes, the number of bytes to be transmitted varies depending on the command. The CS line is LO active, during data transfer via SPI LO must be placed on the CS line. Only when the CS line goes high, the command is executed.

CS-LO	Command 1	Command 2	Command 3	Command 4	CS-HI

The DIGI-DOT Booster has a 256-byte instruction memory. During a data transfer multiple commands can be strung together and sent in a packet. Please make sure when programming by yourself that the maximum number is not exceeded.

After transmission of the data packet a short waiting time of 2-4 milliseconds should be planned to allow the DIGI-DOT Booster time to process the data.

Command set

Here is a list of the commands of the DIGI-DOT Boosters. It must always be sent the specified number of bytes. Are for a parameter valid minimum or maximum values are given, they must not be exceeded or fallen short. If this happens, the command is ignored.

BOOSTER_INIT

0xB1	COUNT	BITS

0XB1 = Command code: BOOSTER_INIT
COUNT = Number of LEDs (COUNT). Value must be an even number, valid values: 2..256 (0 = 256)
BITS = Number of LEDs (LED_COUNT). Value must be an even number, valid values: 2..256 (0 = 256)

The DIGI-DOT Booster hardware is initialized, set the number and type of connected LEDs. This command must always be executed first. Without this command no Data Out-handover is possible based on the LEDs.
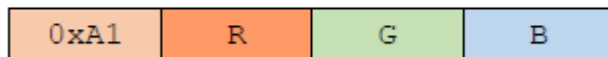
BOOSTER_SHOW

0xB2

0xB2 = Command code: BOOSTER_SHOW

Only after execution of this command changes to the LEDs are displayed. The data from the LED latch are sent to the LEDs. This command should always be the last in a SPI data packet. After sending this command a fixed time remains to be seen, to be sent to new data for DIGI-DOT booster. Depending on the indicated number of LEDs is calculated this by: * LED_COUNT 30µs. With 256 LEDs, this time of 8 ms. If this time is exceeded, it may cause unexpected flicker effects.

BOOSTER_SETRGB

0xA1	R	G	B

0xA1 = Command code: BOOSTER SET RGB
R = Color value for RED
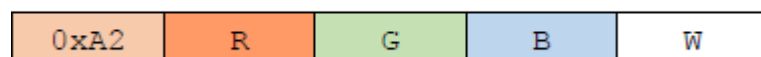Valid values: 0..255
G = Color value for GREEN
Valid values: 0..255
B = Color value for BLUE
Valid values: 0..255

The color register is set to the specified RGB values. In this register the commands BOOSTER_SETLED, BOOSTER_SETALL and BOOSTER_SETRANGE access to.

In the 32-bit RGB mode, a value for the white level is calculated from the RGB values. If this loading calculation does not meet the requirements, please use the BOOSTER_SETRGBW command.

BOOSTER_SETRGBW

0xA2	R	G	B	W

0xA1 = Command code: BOOSTER_SET_RGB
R = Color value for RED
Valid values: 0..255
G = Color value for GREEN
Valid values: 0..255
B = Color value for BLUE
Valid values: 0..255
W = Color value for WEISS
Valid values: 0..255

The color register is set to the specified RGB values. In this register the commands BOOSTER_SETLED, BOOSTER_SETALL and BOOSTER_SETRANGE access to.
In 24-bit RGB mode, the value for white is ignored.

BOOSTER_SETHSV

0xA3	HL	HH	S	V

0xA3 -Command code: BOOSTER_SETHSV
HL and HH - Value for HUE (hue)
Valid values: 0..359
S -Value for SATURATION (saturation)
Valid values: 0..255
V - Color value for VOLUME (brightness)
Valid values: 0..255

The values for HUE, SATURATION and VALUE are converted into an RGB value and set in the color register. In this register the commands BOOSTER_SETLED, BOOSTER_SETALL and BOOSTER_SETRANGE are applied. Since the value for HUE ranges from 0 to 359, for this must always 2 bytes transferred (low first, then high).

In 32-bit RGB mode, the value for white is calculated. If this calculation does not meet the requirements, please use the BOOSTER_SETRGBW command.

BOOSTER_SETLED

0xA4	LED

0xA4 - Command code: BOOSTER SET LEAD
LED - LED Number Valid values: 0..LED_COUNT – 1

A single LED is controlled by the value of the color register. Please note that the numbering starts at zero. The first LED has the number 0, the last LED corresponds to the set in the init number minus 1. For example, if 50 LEDs are set the valid range is 0 to 49.

BOOSTER_SETALL

0xA5

0xA5 - Command code: BOOSTER_SETALL

All LEDs configured in the init command are driven with the value from the color register.

BOOSTER_SETRANGE

0xA6	BEG	END

0xA6 = Command code: BOOSTER SetRange
BEG = LED number beginning Valid values: 0..LED_COUNT - 1
END = LED Number End Valid values: BEG ... LED_COUNT – 1

The LEDs from BEG to END are driven by the value of the color register. The value of END must be equal to or greater than that of his BEG. Please refer to the notes on numbering in BOOSTER_SETLED command.

BOOSTER_SETRAINBOW

0xA7	HL	HH	S	V	BEG	END	INC

0xA7 = Command code: BOOSTER SET RAINBOW
BEG = LED number beginning Valid values: 0..LED_COUNT - 1
END = LED Number End Valid values: BEG ... LED_COUNT - 1
HL and HH Value for HUE (hue) Valid values: 0..359
S = Value for SATURATION (saturation) Valid values: 0..255
V = Color value for VOLUME (brightness) Valid values: 0..255
INC = Value is added to the HUE value between successive LEDs Recommended values: 2..20

The LEDs from BEG to END are driven in a rainbow effect. The value of END must be equal to or greater than that of his BEG. Please refer to the notes on numbering in BOOS-TER_SETLED command.

BOOSTER_SHIFTUP

0xB3	BEG	END	COUNT

0xB3
0XB3 = Command code: BOOSTER_SHIFTUP
BEG = LED number beginning Valid values: 0..LED_COUNT - 1
END = LED Number End Valid values: BEG ... LED_COUNT - 1
COUNT = Number of LEDs that are moved. Valid values: 1..LED_COUNT – 1

The LEDs from BEG to END to move to the number in COUNT upwards. The value of END must be equal to or greater than that of his BEG. Please refer to the notes on numbering in BOOSTER_SETLED command.

BOOSTER_SHIFTDOWN

0xB4	BEG	END	COUNT

0XB4 = Command code: BOOSTER_SHIFTUP
BEG = LED number beginning Valid values: 0..LED_COUNT - 1
END = LED Number End Valid values: BEG ... LED_COUNT - 1
COUNT = Number of LEDs that are moved. Valid values: 1..LED_COUNT – 1

The LEDs from BEG to END to move to the number in COUNT down. The value of END must be equal to or greater than that of his BEG. Please refer to the notes on numbering in BOOSTER_SETLED command.

BOOSTER_COPYLED

0xB5	FROM	TO

0xB5 = Command code: BOOSTER COMPILED
FROM = LED number Original Valid values: 0..LED_COUNT - 1
TO = LED-number target Valid values: 0..LED_COUNT – 1

The LED FROM is copied to LED TO. LED FROM retains its original value. Please refer to the notes on numbering in BOOSTER_SETLED command.

BOOSTER_REPEAT

0xB6	BEG	END	COUNT

0xB6 = Command code: BOOSTER REPEAT
BEG = LED number beginning Valid values: 0..LED_COUNT - 1
END = LED Number End Valid values: BEG..LED_COUNT - 1
COUNT = LED-number target Valid values: 1..x

This command is used for multiple repetition of a LED array.
Example: BEG = 0, END = 7, COUNT = 7

The values of the LEDs 0 to 7 are 7 times copied in succession, so there is the same pattern a total of 8 times, 64 LEDs are driven total.
Please make sure that the maximum number of specified in Initialization LEDs is not exceeded. The basis for calculation is the following formula: END + 1 + ((END - BEG + 1) * COUNT)

Please refer to the notes on numbering in BOOSTER_SETLED command.

BOOSTER_RGBORDER

0xC1	IDX-R	IDX-G	IDX-B

0xC1
Command code: BOOSTER_RGBORDER
IDX-R
Index for the LED data RED Valid values: 1..3, default: 2 (for WS2812)
IDX-G
Index for the LED data GREEN Valid values: 1..3, default: 3 (for WS2812)
IDX B
Index for the LED data BLUE Valid values: 1..3, default: 1 (for WS2812)

The arrangement of the LED data can be changed here. WS2812 are in the order of G-R-B is-controls.
Example: PL9823 need R-G-B, as a command has to be sent 0xC1-0x03-0x02-0x01.

Bibliography
Datasheet WS2812B, © World Semi
Datasheet PL9823, © Baicheng
Datasheet SK6812, © Shenzhen LED color optoelectronic Co., ltd

Links
LED Genial online shop
http://www.led-genial.de

Led'sWork blog
http://www.ledswork.de

Vertrieb

DIAMEX Produktion und Handel GmbH

Innovationspark Wuhlheide
Köpenicker Straße 325, Haus 41
12555 Berlin

Telefon: 030-65762631

E-Mail: info@diamex.de
Homepage: http://www.diamex.de

Herstellung

www.tremex.de
Köpenicker Str. 325 12555 Berlin
Tel.030-65762631

Hersteller: Tremex GmbH
DIAMEX ⚹ OBD-DIAG ⚹ TREMEX
WEE-Reg.Nr. DE 51673403