



# Display - NeoPixel (7-Segment)

NeoPixel 7-Segment display

## Plugin details

Type: Display

Name: NeoPixel (7-Segment)

Status: DISPLAY NEOPIXEL

GitHub: [P165\\_7SegNeopixel.ino \(https://github.com/letscontrolit/ESPEasy/blob/mega/src/\\_P165\\_7SegNeopixel.ino\)](https://github.com/letscontrolit/ESPEasy/blob/mega/src/_P165_7SegNeopixel.ino)

Maintainer: tonhuisman

Used libraries: Noiasca NeopixelDisplay (modified)

## Description

NeoPixels are very versatile, multi-color led pixels, that can be used in a lot of situations, and only need a single GPIO to be able to control multiple leds as they are connected sequentially.

This plugin offers multiple layout 7-segment displays, for use as clock, score-board, news-ticker etc.

Configurations can have from 1..7 pixels per segment, and up to 15 extra pixels, that can f.e. be used as a time-colon. Options for initial, unused, pixels and several settings for generating the number-plan for connecting the pixels in the correct order and position are available.

The minimum size for a display would have 1 digit, starting from 7 pixels with 1 pixel per segment and no decimal point, and the max. configuration possible has 16 digits, split in 4 groups that can be addressed as a single display or per group, with up to 1100 pixels, though that may have issues as that number of pixels hasn't been tested. Technically, larger displays are possible, but that might cause even more (signal-timing) issues because of the huge number of pixels needed.

## Hardware

Most of the work is probably spent in mounting the pixels in a panel or casing, so experimenting with layouts may be helpful in finding the optimal configuration, as f.e. changing the number of pixels per segment, decimal point or extra pixels later can be a lot of work.

## Signal wiring

### Note

If the strips use a VCC higher than 3.3V, f.e. 5V (and they usually do), a level converter for 3V3 to 5V should be used when connecting the display to the ESP, as the ESP might be damaged, or the logic levels don't match the requirements of the display (high level ~70% of VCC = 3.5V @ VCC 5V), and may not work properly.

ESP	Level converter		5V	Strip 1	5V	Strip 2	5V	Strip n
----	-----			-----		-----		-----
3V3 -->	LV	HV	<-->	VCC	-->	VCC	-->	VCC ...
GND ---	GND	GND	---	GND - GND	---	GND - GND	---	GND ...
GPIO -->	LV1	HV1	-[470]-	DIN - DOUT	-->	DIN - DOUT	-->	DIN ...

NB: It is advised to use a single 220 to 470 ohm resistor in the wiring from ESP (after the level converter) to the first strip **Din** for better signal stability.

## Power consumption

By the physical design of these modules, using NeoPixel leds for display that can consume up to 60 mA each, the power consumption is dependent on the number of neopixels that are activated and the brightness that is set. On a set of NeoPixels, like consisting of 64 pixels, the max. power used  $60 \text{ mA} * 64 = 3840 \text{ mA} @ 5\text{V} = 19.2 \text{ watt}$  with all leds on at max brightness (255), so it requires an adequate power supply, preferably separate from the power supply used to power the ESP.

When reducing the brightness to 40% ( $0.4 * 255 = 102$ ), that power is reduced to ca.  $1.5 \text{ A} @ 5\text{V} = \text{ca. } 7.5 \text{ watt}$ . 40% is usually enough during daylight conditions, at night the brightness can often be reduced to 20%.

For a set of 300 pixels, the max. power would count up to  $300 * 60 \text{ mA} @ 5\text{V} = 90 \text{ W}$ . At 40% brightness, that is reduced to 36 W.

Reasoning back from the available power supply max. load (f.e. 30 W), the max. allowed brightness can be calculated by dividing the max. load by the voltage (5V) and the number of pixels ( $300 = 20 \text{ mA}$ ,  $60 / 20 = 3 = \sim 33\% \text{ brightness} = 85$ ). That will still be bright enough, but will put quite a strain at the power supply.

Also, when using several strips sequentially, separate wiring should be added for VCC and GND to each strip (or group of strips), to avoid burning out the pcb traces of the strip where the power is connected. For a group of strips of 100 to 150 LEDs, it is advised to use separate wiring for power to each group, and only connect the GND and signal connections Dout/GND to Din/GND of the next group, but **never** the VCC pins.

## Configuration

## Task Settings

Device: Display - NeoPixel (7-Segment) ? i

Name: NeoPixel7Seg

Enabled:

### Actuator

GPIO → Strip: - None -

### Device Settings

Strip Type: GRB

Initial brightness: 40 [0..255]

Maximum allowed brightness: 255 [1..255]

Display Output: Manual

Start at digit: 1

Note: Min. 4 digits used for Time/Date Display Output

Font set: Default

Note: Check documentation for examples of the font sets.

### Options

Suppress leading 0 on day/hour:

Scroll text > display width:

Scroll text in from right:

Scroll speed (0.1 sec/step): 10 [1..600 = 0.1..60 sec/step]

Clear display on exit:

### Display

Note: Fields with \*: When changed will save and reload the page.

Number of Groups \*: 1

Foreground color: red

Background color: black

Note: Use Color name, '#RGB565' (# + 1..4 hex nibbles) or '#RRGGBB' (# + 6 hex nibbles RGB color).

Note: NB: Colors stored as RGB565 value!

Show Pixel number-plan \*: No

Note: When set to Yes will block digit-changes.

### Groups and Digits

Number of Digits \*: 1 Digit 1 (Extra)

Segment Width pixels: 2

Segment Height pixels: 2

Segment Corners overlap:

Decimal dot pixels: 1

Extra pixels after: 0

Group 1: Pixels in group: 15

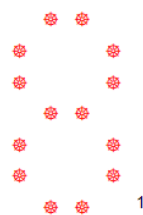
Pixel-offset before: 0

Starting pixel: Left-top

Split g-segment pixels:

Decimal dot last segment:

Numbering counter-clockwise:



Right to Left digits:

Total digits:

Total pixels to connect:

Close

Submit

Delete

Powered by [Let's Control It](#) community

- **Name:** Required by ESPEasy, must be unique among the list of available devices/tasks.
- **Enabled:** The device can be disabled or enabled. When not enabled the device should not use any resources.

## Actuator

- **GPIO -> Strip:** Select the GPIO pin where the NeoPixel strip will be connected. This has to be a pin that allows output (not input-only).

## Device Settings

- **Strip Type:** Select the type of the LEDs mounted on the strip:

Strip Type:

GRB

GRB

GRBW

- **GRB** : Most commonly, the LEDs are of GRB type.
- **GRBW** : If the LEDs also support Brightness control, then the GRBW type should be selected.

- **Initial brightness:** The brightness that the display is initially set to. Default is 40, to protect both the eyes and the power-supply powering the display. Range: 0..255.
- **Maximum allowed brightness:** The brightness that is the maximum it can be set to, either from UI or via the `7db` command, to optionally help protect both the eyes and the power-supply powering the display. Default is 255. Range: 1..255.
- **Font set:** Select the font set from this list: (These fonts are the same fonts as available in Display - 7-segment display (P073.html#p073-page))

Font set:

Default

Default

Siekoo

Siekoo with uppercase 'CHNORUX'

dSEG7

**Default:** The original font as initially included in plugin P073. Includes digits 0..9, special characters: space, dash, degree (when using a `^`), equal sign, slash, underscore and letters A..Z. Uppercase and lowercase characters are shown exactly the same, but as the possible shapes are quite limited, some can be somewhat hard to recognize at first.

**Siekoo:** The character set as documented here [Fakoo.de: Siekoo alphabet \(https://fakoo.de/siekoo.html\)](https://fakoo.de/siekoo.html) (See below for the extra special characters supported)

**Siekoo with uppercase 'CHNORUX':** This font is the same as the normal Siekoo character set, but supports uppercase versions

for characters 'CHNORUX', even though they duplicate other characters in the set.

dSEG7: the character set as documented here [Keshikan.net: dSEG7 \(https://www.keshikan.net/fonts-e.html\)](https://www.keshikan.net/dSEG7) special characters: space, dash, degree (when using a <sup>^</sup>), equal sign, slash, underscore and letters A..Z. (Again single-case, like the default font)

If the **Font set** configuration option is not available, then a different font layout, default from the used library, is used.

**Siekoo:**

**das Siekoo-Alphabet für eine verwechslungsfreie 7-Segment-Anzeige**

The image displays a grid of 7-segment displays representing the Siekoo alphabet. The characters are arranged in rows: Row 1: A-K; Row 2: L-V; Row 3: W-Z, a highlighted box containing Ä, Ö, Ü, ß, and @; Row 4: 1-0, %; Row 5: ., , ; : + - = \* # ! ?; Row 6: \_ - ' " ° < > / \ ( )

[www.fakoo.de/siekoo](http://www.fakoo.de/siekoo)

The four marked characters, Ä, Ö, Ü and ß, are *not* included, as they can not reliably be sent to the unit because of conversion issues from ASCII/UTF-8/ISO character sets. And they are possibly not often used, except in German, and some closely related, languages.

**dSEG7:**

Character	DSEG7	Character	DSEG7	Character	DSEG7
"	N/A	B	b	b	b
\$	N/A	C	c	c	c
%	N/A	D	d	d	d
&	N/A	E	E	e	E
'	N/A	F	F	f	F
(	N/A	G	G	g	G
)	N/A	H	h	h	h
*	N/A	I	i	i	i
+	N/A	J	J	j	J
,	N/A	K	K	k	K
-	-	L	L	l	L
.	.	M	M	m	M
/	N/A	N	n	n	n
0	0	O	o	o	o
1	1	P	P	p	P
2	2	Q	Q	q	Q
3	3	R	r	r	r
4	4	S	S	s	S
5	5	T	T	t	T
6	6	U	u	u	u
7	7	V	V	v	V
8	8	W	W	w	W
9	9	X	X	x	X
:	:	Y	Y	y	Y
<	N/A	Z	Z	z	Z
=	N/A	\	N/A		N/A
>	N/A	^	N/A	~	N/A
?	N/A	_	-	¥	N/A
@	N/A	`	N/A		N/A
A	A	a	A	°	◻
				±	N/A

The  character is again used to display the degree symbol, and the degree symbol is not recognized!

**Note**

Any periods used in tekst to be displayed by the plugin will be placed at the previous digit decimal point position and removed from the text, regardless of a decimal point is defined for the digit!

**Options**

- **Suppress leading 0 on day/hour:** When enabled, will show the hours of the time and days of the date without a leading 0

when < 10.

- **Scroll text > display width:** Normally the `7dtext, <text>` command only show the left n characters the display can hold. This option enables the Scroll Text feature, that will scroll text sent using the `7dtext` or `7dbin` commands from right to left when the content is longer than the display can show at once.
- **Scroll text in from right:** Normally the Scroll Text feature starts with the display filled with the left part of the text to scroll, with this option enabled, the display starts empty and the text is scrolled in from the right side of the display to the left, until all text is scrolled off. Then the scrolling restarts.
- **Scroll speed (0.1 sec/step):** Determines the speed of scrolling the text. Default value is 10, so 1 character per second.
- **Clear display on exit** When checked, will clear the display when the task is disabled, either from settings or via the `TaskDisable` command. This will fill the strip with black pixels, turning all NeoPixels off.

## Display

---

### Note

Some field-labels show an asterisk `*` as the last character, to indicate that when a different option is selected on these fields, the current settings are stored and the page is reloaded, so the changed configuration will reflect in the page.

- **Number of Groups \***: Select the number of digit-groups to use. With current configuration, extra pixels can only be applied at the end of a group, so if f.e. a time display is to be made with a colon between the hours and minutes (that can optionally blink every other second), 2 groups should be defined, 1 with 2 digits and extra pixels, and a second with 2 digits only. The max. number of groups is 4, for settings-technical reasons.
- **Foreground color**
- **Background color**

These are the default colors, used to display the content.

Colors can be specified in 3 ways:

- **Color name** Some named colors are recognized: White, Black, Red, Green, Blue, Yellow, Orange, Navy, Darkgreen, Cyan, Darkcyan, Maroon, Purple, Olive, Magenta, Lightgrey, Darkgrey, Pink and Greenyellow. If any of these colors is recognized, the name will be shown.
- **#RGB565 A #** prefixed, 1 to 4 digit, hex-coded color value in RGB565 format (5 bits red, 6 bits green, 5 bits blue) giving the max. number of 65535 colors. This is the value that is shown after the settings are saved, and it is not one of the known color names.
- **#RRGGBB A #** prefixed, 6 digit, hex coded RGB color value. Will be transformed to an RGB565 value on save!
- **Show Pixel number-plan \***: When changed to Yes, the digit preview display will change from showing pixels, to showing the number-index of each pixel as wired in the strip. This also blocks changing any of the digit options below (calculation of the number-plan is not implemented in the front-end Javascript code).

## Groups and Digits

---

This section shows 1 to 4 groups of digit configurations, depending on the **Number of Groups** selection. All groups have the exact same settings options.

The group-order is reverted (4..1) if all groups have the **Right to Left digits** option enabled.

These options determine the layout of the digits:

- **Number of Digits \***: Select the number of digits in this group. Currently limited to 4.
- **Segment Width pixels**: The number of pixels to use in the horizontal segments, range 1..7.
- **Segment Height pixels**: The number of pixels to use in the vertical segments, range 1..7.
- **Segment Corners overlap**: When enabled, 6 extra pixels, where the segments (can) overlap, will be added. These pixels will light up if either of the segments it belongs to is lit.
- **Decimal dot pixels**: The number of pixels to use for a decimal point, to the lower-right of each digit, range 0..7.
- **Extra pixels after**: The number of pixels to be used, either as a 'bridge' to the next digit, or as an optional colon or other

shape, that will light up when using a time or date display output, and can also be controlled via the `7dextra` command.

- **Pixels in group:** A read-only field that shows the currently needed number of pixels in this group. This will be updated dynamically when any of the above settings is changed.

These options determine the number-plan of the pixels in this group:

- **Pixel-offset before:** The number of pixels to skip before the first digit is planned. Sometimes used to 'fix' the signal level for the pixels if the strip is connected to the ESP without a level converter, but that's an unreliable fix, so a level converter should best be used when powering the pixels with the 'suggested' (strongly advised) 5V. These pixels can also be used as an extra indicator, as the `7dbefore` command can be used to set these pixels to a color.
  - **Starting pixel:** By default the number-plan starts at the left-top pixel of a digit, and flows clock-wise along all pixels for a single digit. With this setting you can select to start at the right-top of the digit, for assigning the number-plan. (See remark when using **counter-clockwise** option, below.)
  - **Split g-segment pixels:** The default number-plan assigns pixel-indexes to the g-segment between the e and f segment. When using a larger number of Width pixels, wiring the g-segment can be challenging. When enabled, this option assigns pixel-indexes to the right-half set of pixels between the b and c segment, and the remaining pixels between the e and f segment.
  - **Decimal dot last segment:** Similar to the g-segment, also the dp-segment can be challenging to wire, as by default the f-segment is assigned the last pixel-index. So the default here is to assign pixel-indexes to the dp-segment between the c and d-segments. If you explicitly want to have these dp-pixels assigned as the last pixel-indexes, this option can be enabled.
  - **Numbering counter-clockwise:** By default the pixel-indexes are assigned in a clock-wise order, but if your wiring-plan works better when numbering counter-clockwise, this is the setting to enable. This also changes how the **Starting pixel** is interpreted: Top-left will start from the f-segment, and end with the a-segment, and top-right will start assigning pixel-indexes from the right side of the a-segment, and work down the f, g, e, d, dp, c and b segments.
  - **Right to Left digits:** When creating larger displays, or building a 2 1/2 or 3 1/2 digit display for displaying f.e. numbers up to 100 or 1000, starting the digits at the left is not efficient, as the entire left digit will get pixel-indexes assigned. When using the Right to Left digits option, the digits will be virtually reverted, and pixel-indexes assigned from the right digit to the left digit. The extra pixels, not needed to show anything else than `1` can be left out of the display. An example config is included below, showing this.
- 
- **Total digits:** Shows the number of digits, as currently configured.
  - **Total pixels to connect:** Shows the number of pixels to connect, as currently configured. If building a 3 1/2 or 2 1/2 digit display, the excess pixels of the left most digit can be manually subtracted.

## Commands available

---

### Command Syntax

`7don`

### Extra information

Turn the display on, if it was turned off before using the `7doff` command. (Not implemented yet)

`7doff`

Turn the display off, can later be turned on again by using the `7don` command. (Not implemented yet)

---



## Command Syntax

7db,<brightness>

<brightness> : Set the brightness between 0 and 255. Will be limited to the max. value configured in **Maximum allowed brightness** setting.

## Extra information

Changes the **Initial brightness** config setting, and the brightness of the strip. The setting is not saved unless the `save` command is used.

7output,<output option>

Change the Display Output setting, available options:

- 0 - Manual
- 1 - Clock 24h - Blink
- 2 - Clock 24h - No Blink
- 3 - Clock 12h - Blink
- 4 - Clock 12h - No Blink
- 5 - Date

7dfont,<font>

Select a different font, either by name: `Default` or `7dgt` for the original font (0), `Siekoo` for standard Siekoo font (1), `Siekoo_Upper` for the Siekoo font with uppercase 'CHNORUX' characters (2), and `dSEG7` for the dSEG7 font (3). Numbers can also be used, 0, 1, 2, or 3 as noted. Fontnames are not case-sensitive.

## Command Syntax

## Extra information

`7dtext,<text>`

Examples:

`7dtext,OUT [BLK#Temperature#d2.1]^`

`7dtext,HU. [BLK#Humidity#d2.1]^o` Will result in 'HU. 57.2°o' being displayed

To display a text on the display. Any variable can be used. Unsupported characters (like accented letters) will show as a space (empty digit).

With the Scroll Text option disabled, the first n characters the display can show are displayed, if the Scroll Text option is enabled, longer texts will scroll from right to left across the display, at the set speed.

`7digit,<grp>,<dgt>,<char>[.]`

Draw a single character in the exact group & digit. When followed by a period, the decimal point will also be used.

`7dbin,<byte>,...`

Example:

`7dbin,0x40,0x20,0x10,0x08,0x04,0x02,0x01,0x80`

To display any bit pattern on the display.

The example shows each digit with a different segment on, including the dot/colon, and assuming an 8 digit configuration, or Scroll Text enabled.

See explanation on how the bits map to segments, below.

`7dgroup,<group>`

`<group>` Range 0..<number of configured groups>

Set the default group any command should be applied to. Group 0 is the entire display, and the default.

## Command Syntax

```
7color,<fg_r>,<fg_g>,<fg_b>[,<fg_w>],<bg_r>,<bg_g>,<bg_b>[,<bg_w>]
```

## Extra information

Set the foreground, and/or background colors for the entire display. When a GRBW strip is configured, the brightness/white values have to be provided too. For only setting the background colors, leave out values for the foreground colors, but keep the commas.

Color values can be in range 0..255.

When group or digit colors are configured, then those will get priority.

```
7digitcolor,<grp>,<dgt>,<fg_r>,<fg_g>,<fg_b>[,<fg_w>],<bg_r>,<bg_g>,<bg_b>[,<bg_w>]
```

Set the foreground, and/or background colors for a specific digit of a group. When a GRBW strip is configured, the brightness/white values have to be provided too. For only setting the background colors, leave out values for the foreground colors, but keep the commas.

Both <grp> (group) and <dgt> (digit) are required, and must be in range 1..[available groups/digits for that group](#).

Color values can be in range 0..255.

To clear a digit color, to use the group or global defaults, provide `-` instead of the <fg\_r> value. The rest of the arguments will then be ignored.

Digit colors have highest priority.

---

## Command Syntax

```
7groupcolor, <grp>, <fg_r>, <fg_g>, <fg_b>[, <fg_w>], <bg_r>, <bg_g>, <bg_b>[, <bg_w>]
```

## Extra information

Set the foreground, and/or background colors for a all digits of a group. When a GRBW strip is configured, the brightness/white values have to be provided too. For only setting the background colors, leave out values for the foreground colors, but keep the commas.

Group <grp> is required, and must be in range 1..<available groups>.

Color values can be in range 0..255.

To clear a group color, to use the global defaults, provide `-` instead of the <fg\_r> value. The rest of the arguments will then be ignored.

Group color will be overridden by a digit color, when set.

## Command Syntax

```
7dextra,<grp>,<state>[,<r>,<g>,<b>[,<w>]]
```

```
7dbefore,<grp>,<state>[,<r>,<g>,<b>[,<w>]]
```

<state> : 0, 1, 2, 3, 4 or 5

## Extra information

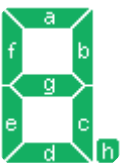
Set the extra pixels after or before the digit On or Off, according to the state. The optionally supplied color will be used for the state. If not used, the the global background color will be used for Off, and the global foreground color for On.

State:

- 0 : All Off
- 1 : All On
- 2 : Low-numbered half of pixels Off
- 3 : Low-numbered half of pixels On
- 4 : High-numbered half of pixels Off
- 5 : High-numbered half of pixels On

## Bit to segment mapping for 7dbin command

The `7dbin` command allows to show any combination of segments on the display according to a (sequence of) bit pattern(s).



The mapping from bits to segments is: `0Bhabcdefg` (Similar to the segment mapping used by Display - 7-segment display (P073.html#p073-page))

ESPEasy allows decimal, hexadecimal and binary notation for numbers. This makes creating the desired display pattern easy when using the binary notation (starting with `0B` or `0b`).

Switching on all horizontal segments for a digit can be done by the command `7dbin,0b01001001`, or in hexadecimal notation:

```
7dbin,0x49
```

## Example configurations

## Example 1

---

For displaying the time in a 24 hour format, and an optionally blinking colon as the hours/minutes separator, you can define this Group/Digit setup:

**Display**

Note: Fields with \*: When changed will save and reload the page.

Number of Groups \*:

Foreground color:

Background color:

Note: Use Color name, '#RGB565' (# + 1..4 hex nibbles) or '#RRGGBB' (# + 6 hex nibbles RGB color).

Note: NB: Colors stored as RGB565 value!

Show Pixel number-plan \*:

Note: When set to Yes will block digit-changes.

**Groups and Digits**

Number of Digits \*:  Digit 1 Digit 2 (Extra)

Segment Width pixels:

Segment Height pixels:

Segment Corners overlap:

Decimal dot pixels:

Extra pixels after:

Group 1: Pixels in group:

Pixel-offset before:

Starting pixel:

Split g-segment pixels:

Decimal dot last segment:

Numbering counter-clockwise:

Right to Left digits:

Number of Digits \*:  Digit 1 Digit 2 (Extra)

Segment Width pixels:

Segment Height pixels:

Segment Corners overlap:

Decimal dot pixels:

Extra pixels after:

Group 2: Pixels in group:

Pixel-offset before:

Starting pixel:

Split g-segment pixels:

Decimal dot last segment:

Numbering counter-clockwise:

Right to Left digits:

Total digits:



Total pixels to connect:

90

Close

Submit

Delete

Powered by [Let's Control It community](#)

(../\_images/P165\_example1\_config.png)

This defines 2 Groups of each 2 Digits. The height and width of the segments can be configured as preferred, this example uses 3 pixels for both. When using more pixels per segment, like 5, 6 or 7, it is probably better to define a larger decimal point, if that is to be used, 3 seems appropriate in that case, and also the extra pixels, used to add the colon separator, could have 4 to 8 pixels to get a matching size with the large segments.

The number-plan for this layout is presented like this:

**Display**

Note: Fields with \*: When changed will save and reload the page.

Number of Groups \*:

Foreground color:

Background color:

Note: Use Color name, '#RGB565' (# + 1..4 hex nibbles) or '#RRGGBB' (# + 6 hex nibbles RGB color).

Note: NB: Colors stored as RGB565 value!

Show Pixel number-plan \*:

Note: When set to Yes will block digit-changes.

**Groups and Digits**

Number of Digits \*:  Digit 1 Digit 2 (Extra)

Segment Width pixels:

Segment Height pixels:

Segment Corners overlap:

Decimal dot pixels:

Extra pixels after:

Group 1: Pixels in group:

		1	2	3		23	24	25	
		22		4		44		26	
		21		5		43		27	
		20		6		42		28	
			17	18	19		39	40	41
		16		7		38		29	45
		15		8		37		30	46
		14		9		36		31	
			13	12	11	10		35	34
								33	32

Pixel-offset before:

Starting pixel:

Split g-segment pixels:

Decimal dot last segment:

Numbering counter-clockwise:

Right to Left digits:

Number of Digits \*:  Digit 1 Digit 2 (Extra)

Segment Width pixels:

Segment Height pixels:

Segment Corners overlap:

Decimal dot pixels:

Extra pixels after:

Group 2: Pixels in group:

			47	48	49		69	70	71
		68		50		90		72	
		67		51		89		73	
		66		52		88		74	
			63	64	65		85	86	87
		62		53		84		75	
		61		54		83		76	
		60		55		82		77	
			59	58	57	56		81	80
								79	78

Pixel-offset before:

Starting pixel:

Split g-segment pixels:

Decimal dot last segment:

Numbering counter-clockwise:

Right to Left digits:

Total digits:

Total pixels to connect:

90

Close

Submit

Delete

Powered by [Let's Control It](#) community

(../\_images/P165\_example1\_number.png)

## Example 2

---

For displaying the time in a 12 hour format, and an optionally blinking colon as the hours/minutes separator, you can define this Group/Digit setup:

### Options

- Suppress leading 0 on day/hour:
- Scroll text > display width:
- Scroll text in from right:
- Scroll speed (0.1 sec/step):  [1..600 = 0.1..60 sec/step]
- Clear display on exit:

### Display

Note: Fields with \*: When changed will save and reload the page.

- Number of Groups \*:
- Foreground color:
- Background color:
- Show Pixel number-plan \*:

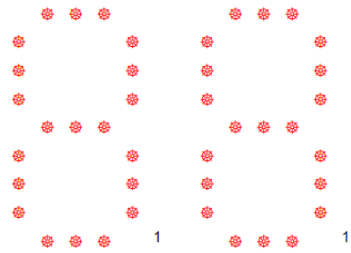
Note: Use Color name, '#RGB565' (# + 1..4 hex nibbles) or '#RRGGBB' (# + 6 hex nibbles RGB color).  
Note: NB: Colors stored as RGB565 value!

Note: When set to Yes will block digit-changes.

### Groups and Digits

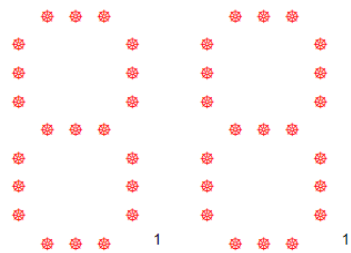
Note: Attention, Groups are shown in reverse order!

- Number of Digits \*:  (Extra) Digit 1 Digit 2
- Segment Width pixels:
- Segment Height pixels:
- Segment Corners overlap:
- Decimal dot pixels:
- Extra pixels after:
- Group 2: Pixels in group:
- Pixel-offset before:
- Starting pixel:
- Split g-segment pixels:
- Decimal dot last segment:
- Numbering counter-clockwise:
- Right to Left digits:



(../)

- Number of Digits \*:  (Extra) Digit 1 Digit 2
- Segment Width pixels:
- Segment Height pixels:
- Segment Corners overlap:
- Decimal dot pixels:
- Extra pixels after:
- Group 1: Pixels in group:
- Pixel-offset before:
- Starting pixel:
- Split g-segment pixels:
- Decimal dot last segment:
- Numbering counter-clockwise:
- Right to Left digits:



Total digits:

Total pixels to connect:

Close

Submit

Delete

Powered by [Let's Control It](#) community

\_images/P165\_example2\_config.png)

The option to suppress the leading 0 on day/hour option is enabled now, so the left-most digit will be empty most of the time.

We have defined 2 Groups of each 2 Digits, but effectively use only 3 1/2 digit. The height and width of the segments can be configured as preferred, this example uses 3 pixels for both. When using more pixels per segment, like 5, 6 or 7, it is probably better to define a larger decimal point, if that is to be used, 3 seems appropriate in that case, and also the extra pixels, used to add the colon separator, could have 4 to 8 pixels to get a matching size with the large segments.

#### Note

This layout is presented in reverse order, as all groups use the **Right to Left digits** option. The actual positioning is still with Group 1 as the left-most group, and then using the numerically next group(s). This layout is needed to make the number-plan work from right to left.

The number-plan for this layout is presented like this:

Show Pixel number-plan \*:

Yes

Note: When set to Yes will block digit-changes.

Groups and Digits

Note: Attention, Groups are shown in reverse order!

Group 2:

Number of Digits \*: 2 (Extra) Digit 1 Digit 2

Segment Width pixels: 3

Segment Height pixels: 3

Segment Corners overlap:

Decimal dot pixels: 1

Extra pixels after: 2

Pixels in group: 46

Pixel-offset before: 0

Starting pixel: Right-top

Split g-segment pixels:

Decimal dot last segment:

Numbering counter-clockwise:

Right to Left digits:

		42 43 44		20 21 22	
	41		23	19	1
	40		24	18	2
	39		25	17	3
	45 46	36 37 38		14 15 16	
	35		26	13	4
	34		27	12	5
	33		28	11	6
		32 31 30	29	10 9 8	7

Group 1:

Number of Digits \*: 2 (Extra) Digit 1 Digit 2

Segment Width pixels: 3

Segment Height pixels: 3

Segment Corners overlap:

Decimal dot pixels: 1

Extra pixels after: 0

Pixels in group: 44

Pixel-offset before: 0

Starting pixel: Right-top

Split g-segment pixels:

Decimal dot last segment:

Numbering counter-clockwise:

Right to Left digits:

		88 89 90		66 67 68	
	87		69	65	47
	86		70	64	48
	85		71	63	49
		82 83 84		60 61 62	
	81		72	59	50
	80		73	58	51
	79		74	57	52
		78 77 76	75	56 55 54	53

Total digits:

4

Total pixels to connect:

90

Close

Submit

Delete

Powered by [Let's Control It community](#)

(../\_images/P165\_example2\_number.png)

The number-plan starts at the Right-Top pixel, instead of the Left-Top pixel, so the a-segment for the left-most digit can be excluded.

To save space and a few pixels, or create a larger display with a standard strip-length, the marked numbers (76..90) can be excluded from the build, as for a 12 hour clock the left-most digit is 0 most of the time, and that 0 is being suppressed, so no pixels will light up.

Show Pixel number-plan \*:

Yes

Note: When set to Yes will block digit-changes.

Groups and Digits

Note: Attention, Groups are shown in reverse order!

Group 2:

Number of Digits \*: 2 (Extra) Digit 1 Digit 2

Segment Width pixels: 4

Segment Height pixels: 4

Segment Corners overlap:

Decimal dot pixels: 1

Extra pixels after: 4

Pixels in group: 62

Pixel-offset before: 0

Starting pixel: Right-top

Split g-segment pixels:

Decimal dot last segment:

Numbering counter-clockwise:

Right to Left digits:

		55 56 57 58		26 27 28 29
		54	30	25 1
		53	31	24 2
		52	32	23 3
		51	33	22 4
	59 60 61	47 48 49 50		18 19 20 21
	62	46	34	17 5
		45	35	16 6
		44	36	15 7
		43	37	14 8
		42 41 40 39	38	13 12 11 10 9

Group 1:

Number of Digits \*: 2 (Extra) Digit 1 Digit 2

Segment Width pixels: 4

Segment Height pixels: 4

Segment Corners overlap:

Decimal dot pixels: 1

Extra pixels after: 0

Pixels in group: 58

Pixel-offset before: 0

Starting pixel: Right-top

Split g-segment pixels:

Decimal dot last segment:

Numbering counter-clockwise:

Right to Left digits:

		117 118 119 120		88 89 90 91
		116	92	87 63
		115	93	86 64
		114	94	85 65
		113	95	84 66
	109 110 111 112			80 81 82 83
	108	96	79	67
	107	97	78	68
	106	98	77	69
	105	99	76	70
	104 103 102 101	100	75 74 73 72	71

Total digits:

4



Total pixels to connect:

120

Close

Submit

Delete

Powered by [Let's Control It community](#)

(../\_images/P165\_example2a\_number.png)

This shows an increased layout, using 4-pixel segments and a 4-pixel colon, that can be built from exactly 100 pixels, 2 50-pixel wired-pixelchains as available from EBay, Aliexpress etc.

Again, the marked pixels (101..120) can be excluded.

### Example 3

---

For displaying a game score, with an indicator of the active player by using a half of the extra pixels, you can define this Group/Digit setup:

## Display

Note: Fields with \*: When changed will save and reload the page.

Number of Groups \*:

Foreground color:

Background color:









Note: Use Color name, '#RGB565' (# + 1..4 hex nibbles) or '#RRGGBB' (# + 6 hex nibbles RGB color).









Note: NB: Colors stored as RGB565 value!

Show Pixel number-plan \*:

Note: When set to Yes will block digit-changes.

## Groups and Digits

	Number of Digits *: <input type="text" value="2"/>	Digit 1	Digit 2	(Extra)
	Segment Width pixels: <input type="text" value="2"/>			
	Segment Height pixels: <input type="text" value="2"/>			
	Segment Corners overlap: <input checked="" type="checkbox"/>			
	Decimal dot pixels: <input type="text" value="0"/>			
	Extra pixels after: <input type="text" value="4"/>			
Group 1:	Pixels in group: <input type="text" value="44"/>			4
	Pixel-offset before: <input type="text" value="0"/>			
	Starting pixel: <input type="text" value="Left-top"/>			
	Split g-segment pixels: <input type="checkbox"/>			
	Decimal dot last segment: <input type="checkbox"/>			
	Numbering counter-clockwise: <input type="checkbox"/>			
	Right to Left digits: <input type="checkbox"/>			

	Number of Digits *: <input type="text" value="2"/>	Digit 1	Digit 2	(Extra)
	Segment Width pixels: <input type="text" value="2"/>			
	Segment Height pixels: <input type="text" value="2"/>			
	Segment Corners overlap: <input checked="" type="checkbox"/>			
	Decimal dot pixels: <input type="text" value="0"/>			
	Extra pixels after: <input type="text" value="0"/>			
Group 2:	Pixels in group: <input type="text" value="40"/>			
	Pixel-offset before: <input type="text" value="0"/>			
	Starting pixel: <input type="text" value="Left-top"/>			
	Split g-segment pixels: <input type="checkbox"/>			
	Decimal dot last segment: <input type="checkbox"/>			
	Numbering counter-clockwise: <input type="checkbox"/>			
	Right to Left digits: <input type="checkbox"/>			

Total digits:

Total pixels to connect:

84

Close

Submit

Delete

Powered by [Let's Control It community](#)

Build: [ESP\\_Easy\\_mega\\_20240831\\_max\\_ESP32\\_16M8M\\_LittleFS\\_ETH Aug 31 2024](#)

(../\_images/P165\_example3\_config.png)

We have defined 2 Groups of each 2 Digits, and enabled the Corners Overlap option to have a more explicitly defined digit. The height and width of the segments can be configured as preferred, this example uses 3 pixels for both. When using more pixels per segment, like 5, 6 or 7, the extra pixels, used to add the left/right player indicator, could have 4 to 8 pixels to get a matching size with the large segments.

The number-plan for this layout is presented like this:

Show Pixel number-plan \*:

Yes

Note: When set to Yes will block digit-changes.

Groups and Digits

Group 1:

Number of Digits *:	<input type="text" value="2"/>	Digit 1	Digit 2	(Extra)
Segment Width pixels:	<input type="text" value="2"/>			
Segment Height pixels:	<input type="text" value="2"/>			
Segment Corners overlap:	<input checked="" type="checkbox"/>			
Decimal dot pixels:	<input type="text" value="0"/>			
Extra pixels after:	<input type="text" value="4"/>	1 2 3 4	21 22 23 24	
Pixels in group:	<input type="text" value="44"/>	20 5	40 25	
		19 6	39 26	
Pixel-offset before:	<input type="text" value="0"/>	15 16 17 18	35 36 37 38	41 42 43 44
Starting pixel:	<input type="text" value="Left-top"/>	14 7	34 27	
		13 8	33 28	
Split g-segment pixels:	<input type="checkbox"/>	12 11 10 9	32 31 30 29	
Decimal dot last segment:	<input type="checkbox"/>			
Numbering counter-clockwise:	<input type="checkbox"/>			
Right to Left digits:	<input type="checkbox"/>			

Group 2:

Number of Digits *:	<input type="text" value="2"/>	Digit 1	Digit 2	(Extra)
Segment Width pixels:	<input type="text" value="2"/>			
Segment Height pixels:	<input type="text" value="2"/>			
Segment Corners overlap:	<input checked="" type="checkbox"/>			
Decimal dot pixels:	<input type="text" value="0"/>			
Extra pixels after:	<input type="text" value="0"/>	45 46 47 48	65 66 67 68	
Pixels in group:	<input type="text" value="40"/>	64 49	84 69	
		63 50	83 70	
Pixel-offset before:	<input type="text" value="0"/>	59 60 61 62	79 80 81 82	
Starting pixel:	<input type="text" value="Left-top"/>	58 51	78 71	
		57 52	77 72	
Split g-segment pixels:	<input type="checkbox"/>	56 55 54 53	76 75 74 73	
Decimal dot last segment:	<input type="checkbox"/>			
Numbering counter-clockwise:	<input type="checkbox"/>			
Right to Left digits:	<input type="checkbox"/>			

Total digits:

4

Total pixels to connect:

84

Close

Submit

Delete

Powered by [Let's Control It](#) community

(../\_images/P165\_example3\_number.png)

The number-plan starts at the Left-Top pixel.

For mounting the middle segment easier/more efficient, this segment can be split in a left and right half, by checking the Split g-segment pixels. The number-plan then looks like this:

Show Pixel number-plan \*:

Yes

Note: When set to Yes will block digit-changes.

Groups and Digits

Group 1:

Number of Digits \*: 2 Digit 1 Digit 2 (Extra)

Segment Width pixels: 2

Segment Height pixels: 2

Segment Corners overlap:

Decimal dot pixels: 0

Extra pixels after: 4

Pixel-offset before: 0

Starting pixel: Left-top

Split g-segment pixels:

Decimal dot last segment:

Numbering counter-clockwise:

Right to Left digits:

		1	2	3	4	21	22	23	24			
		20		5		40		25				
		19		6		39		26				
		17	18	7	8	37	38	27	28	41	42	43
		16		9		36		29		44		
		15		10		35		30				
		14	13	12	11	34	33	32	31			

Group 2:

Number of Digits \*: 2 Digit 1 Digit 2 (Extra)

Segment Width pixels: 2

Segment Height pixels: 2

Segment Corners overlap:

Decimal dot pixels: 0

Extra pixels after: 0

Pixel-offset before: 0

Starting pixel: Left-top

Split g-segment pixels:

Decimal dot last segment:

Numbering counter-clockwise:

Right to Left digits:

		45	46	47	48	65	66	67	68	
		64		49		84		69		
		63		50		83		70		
		61	62	51	52	81	82	71	72	
		60		53		80		73		
		59		54		79		74		
		58	57	56	55	78	77	76	75	

Total digits:

4

[Close](#)[Submit](#)[Delete](#)

Powered by [Let's Control It community](#)

(../\_images/P165\_example3a\_number.png)

The right half of the g-segment is placed between the b and c segments, and the left half of the g-segment between the e and f segments.

To control the player-indicator, the extra pixels 41 and 42 should be placed to indicate the digits from Group 1 being the active player score, and extra pixels 43 and 44 placed to indicate the digits from Group 2 for the other player.

These rules can be added to use 2 buttons (PlayerA and PlayerB) to select the active player:

```
On PlayerA#State=0 Do
  7dextra,0,4 // Player B Off
  7dextra,0,3 // Player A On
Endon
On PlayerB#State=0 Do
  7dextra,0,2 // Player A Off
  7dextra,0,5 // Player B On
Endon
```

## Change log

*Changed in version 2.0:* ...

added 2024-08-30 Initial release version.

Support us by using one of these alternatives:

[Patreon \(https://www.patreon.com/GrovkillenTDer\)](https://www.patreon.com/GrovkillenTDer)

[Ko-Fi](https://ko-fi.com/grovkillentder)

<https://ko-fi.com/grovkillentder>

[PayPal \(https://www.paypal.me/espeasy\)](https://www.paypal.me/espeasy)

[Back to top](#)

© Copyright 2018-2024, ESP Easy.

Created using Sphinx (<http://sphinx-doc.org/>) 7.3.7.