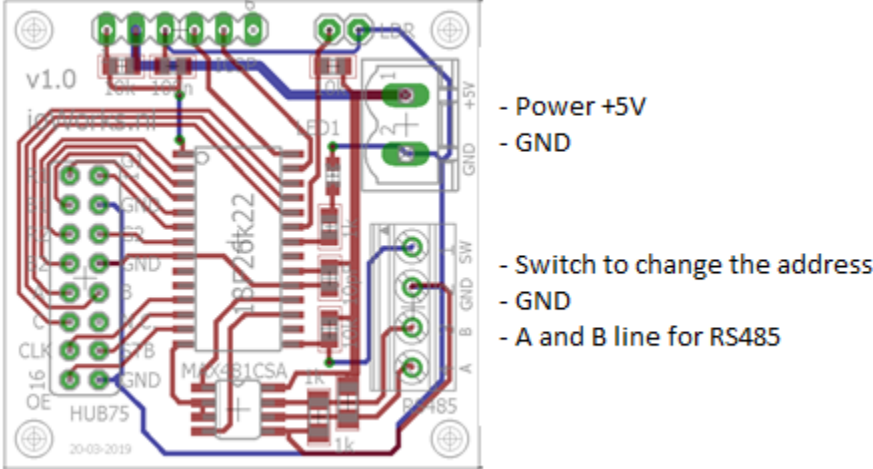


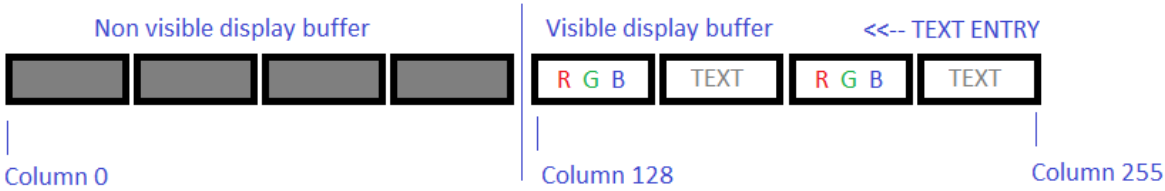
EMT Display driver EL128.64-P6S



The display driver supports up to 8 display modules. For EMT only 4 display modules are implemented. This means that only 1/2 the display buffer is visible at one time.

A single LED display module has 32 columns and 16 rows. The EMT display combines 4 display modules providing a resolution of 128 * 16 RGB pixels.

The first column (zero) is on the left side of the display. The first (implemented) visible display module starts at column 128 (0x80). The last visible column is located at position 255 (0xFF).



It is possible to change the offset of the display pointer such that column zero starts at a different position. Increasing the display offset pointer moves the display buffer to the LEFT. The display buffer is a circular buffer. The characters in the display buffer scroll out of the display buffer at column zero (0x00) and reappear at column 255 (0xFF)

Setting the offset to 128 (0x80) will make text characters positioned at column 0 appear in the middle of the display. That is exactly left of the visible buffer. The “position” in the display buffer and the “offset” are two different things.

Commands - display driver

The following paragraph specifies the instructions that are supported by the displays. The recipient address field allows addressing every display separately using their unique address. The addresses can range from "01".."32". Recipient address "00" is a broadcast address that every display listens to. The Sender address field is not used in this implementation. The displays communication is unconfirmed. No acknowledgements are send back to the originator.

Standard EMT text message: (0x02)

This instruction will clear the display buffer, set the display offset to 128 (0x80) and write the text at position 0 in the display buffer

Field	Description	Possible value	Length	Unit
SOH	Start of Header	0x01	1	Byte(s)
Sender	Sender address	"00"	2	ASCII char
Recipient	Recipient address	"00".."32"	2	ASCII char
Command	Start Of Text	0x02	1	Byte(s)
Text	Payload text	"Hello World"	0..14	ASCII char
EOT	End Of Text	0x03	1	Byte(s)

Note: The Sender address is not used in this implementation and can be left blank or "00"

Example:

```
01 30 30 30 30 02 48 65 6C 6C 6F 20 77 6F 72 6C 64 2E 2E 2E 03
```

Write text message. Commando: "W"

This instruction will clear the display buffer and write the text into the display buffer. Additionally this command can set the color of the display text as well as the position in the display buffer.

Field	Description	Possible value	Length	Unit
SOH	Start of Header	0x01	1	Byte(s)
Sender	Sender address	"00"	2	ASCII char
Recipient	Recipient address	"00".."32"	2	ASCII char
Command	Write Text	"W"	1	ASCII char

Color	Red, Green, Blue, Yellow, Magenta, Cyan, White	R,G,B,Y,M,C,W	1	ASCII char
Position	Position in the buffer (add to the right)	0x00..0xFF	1	(Byte(s))
Text	Payload text	"Hello World"	0..14	ASCII char
EOT	End Of Text	0x03	1	Byte(s)

Note: The Sender address is not used in this implementation and can be left blank or "00"

Example: Color=Red, Position =0x00, Text="Hello world..."

```
01 30 30 30 30 30 57 52 00 48 65 6C 6C 6F 20 77 6F 72 6C 64 2E 2E 2E 03
```

Example: Color=Green, Position =0x00, Text="Hello world..."

```
01 30 30 30 30 30 57 47 00 48 65 6C 6C 6F 20 77 6F 72 6C 64 2E 2E 2E 03
```

Example: Color=Blue, Position =0x01 to the right, Text="Hello world..."

```
01 30 30 30 30 30 57 42 01 48 65 6C 6C 6F 20 77 6F 72 6C 64 2E 2E 2E 03
```

Add text message. Commando: "A" or "T"

This instruction will NOT clear the display buffer and write the text into the display buffer. Additionally this command can set the color of the display text as well as the position in the display buffer.

Field	Description	Possible value	Length	Unit
SOH	Start of Header	0x01	1	Byte(s)
Sender	Sender address	"00"	2	ASCII char
Recipient	Recipient address	"00".."32"	2	ASCII char
Command	Write Text	"A" or "T"	1	ASCII char
Color	Red, Green, Blue, Yellow, Magenta, Cyan, White	R,G,B,Y,M,C,W	1	ASCII char
Position	Position in the buffer (add to the right)	0x00..0xFF	1	(Byte(s))
Text	Payload text	"Hello World"	0..14	ASCII char
EOT	End Of Text	0x03	1	Byte(s)

Note: The Sender address is not used in this implementation and can be left blank or "00"

Example: Color=Red, Position =0x00, Text="Hello world..."

```
01 30 30 30 30 30 41 52 01 48 65 6C 6C 6F 20 77 6F 72 6C 64 2E 2E 2E 03
```

Clear display. Commando: "C"

This instruction will clear the display buffer and set the offset pointer to 128 (0x80)

Field	Description	Possible value	Length	Unit
SOH	Start of Header	0x01	1	Byte(s)
Sender	Sender address	"00"	2	ASCII char
Recipient	Recipient address	"00".."32"	2	ASCII char
Command	Clear display and set offset to 0x80	"C"	1	ASCII char
EOT	End Of Text	0x03	1	Byte(s)

Note: The Sender address is not used in this implementation and can be left blank or "00"

Example: Clear display and set offset to 0x80"

```
01 30 30 30 30 43 03
```

Background color. Commando: "B"

This instruction will clear the display buffer and set the background color

Field	Description	Possible value	Length	Unit
SOH	Start of Header	0x01	1	Byte(s)
Sender	Sender address	"00"	2	ASCII char
Recipient	Recipient address	"00".."32"	2	ASCII char
Command	Set background color	"B"	1	ASCII char
Color	Red, Green, Blue, White, Zwart	R,G,B,W,Z	1	ASCII char
EOT	End Of Text	0x03	1	Byte(s)

Note: The Sender address is not used in this implementation and can be left blank or "00"

Example: Color=Red

```
01 30 30 30 30 42 52 03
```

Example: Color=Blue

```
01 30 30 30 30 42 42 03
```

Example: Color=White

```
01 30 30 30 30 42 57 03
```

Set the offset into display buffer. Commando: "O"

This instruction will set the offset pointer to the display buffer to a specific location. If the offset increments the content of the display scrolls to the left accordingly.

Field	Description	Possible value	Length	Unit
SOH	Start of Header	0x01	1	Byte(s)
Sender	Sender address	"00"	2	ASCII char
Recipient	Recipient address	"00".."32"	2	ASCII char
Command	Offset display buffer	"O"	1	ASCII char
Offset	0x81 is 129 led columns to the right	"0x00".."0xFF"	1	Byte(s)
EOT	End Of Text	0x03	1	Byte(s)

Note: The Sender address is not used in this implementation and can be left blank or "00"

Example: Offset = 0x81

01 30 30 30 30 4F 81 03

Scroll text. Commando: "S"

This instruction will make the text on the display scroll left

Field	Description	Possible value	Length	Unit
SOH	Start of Header	0x01	1	Byte(s)
Sender	Sender address	"00"	2	ASCII char
Recipient	Recipient address	"00".."32"	2	ASCII char
Command	Scroll text	"S"	1	ASCII char
Speed	"0"=still, "1"=fast .. "9"="slow"	"0".."9"	1	ASCII char
EOT	End Of Text	0x03	1	Byte(s)

Note: The Sender address is not used in this implementation and can be left blank or "00"

Example: Scroll speed = "3"

```
01 30 30 30 30 53 33 03
```

Example: Stop scrolling = "0" offset is at an undefined position after this instruction

```
01 30 30 30 30 53 30 03
```

Flash text. Commando: "F"

This instruction will make the display flash the text

Field	Description	Possible value	Length	Unit
SOH	Start of Header	0x01	1	Byte(s)
Sender	Sender address	"00"	2	ASCII char
Recipient	Recipient address	"00".."32"	2	ASCII char
Command	Flash text	"F"	1	ASCII char
Speed	"0"=still, "1"=fast .. "9"="slow"	"0".."9"	1	ASCII char
EOT	End Of Text	0x03	1	Byte(s)

Note: The Sender address is not used in this implementation and can be left blank or "00"

Example: Flash speed = "2"

```
01 30 30 30 30 46 32 03
```

Example: Stop Flashing = "0"

```
01 30 30 30 30 46 30 03
```

Set Power. Commando: "P"

This instruction will set the display power (brightness)

Field	Description	Possible value	Length	Unit
SOH	Start of Header	0x01	1	Byte(s)
Sender	Sender address	"00"	2	ASCII char
Recipient	Recipient address	"00".."32"	2	ASCII char
Command	Power	"P"	1	ASCII char
Brightness	Linear power indicator	0x00 .. 0xFF	1	Byte(s)
EOT	End Of Text	0x03	1	Byte(s)

Note: The Sender address is not used in this implementation and can be left blank or "00"

Example: Power = 0x10

01 30 30 30 30 50 10 03

Reboot display. Commando: "X"

This instruction will reboot / restart the firmware in the display.

Field	Description	Possible value	Length	Unit
SOH	Start of Header	0x01	1	Byte(s)
Sender	Sender address	"00"	2	ASCII char
Recipient	Recipient address	"00".."32"	2	ASCII char
Command	Clear display and set offset to 0x80	"X"	1	ASCII char
EOT	End Of Text	0x03	1	Byte(s)

Note: The Sender address is not used in this implementation and can be left blank or "00"

Example: Color=Red, Position =0x00, Text="Hello world..."

01 30 30 30 30 58 03

Font data - display driver

Windows 1252 font data

The display driver has implemented the Windows 1252 font

For more details please refer to: <https://nl.wikipedia.org/wiki/Windows-1252>

Windows-1252 (CP1252)																
	-0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-A	-B	-C	-D	-E	-F
0-	<u>NUL</u>	<u>SOH</u>	<u>STX</u>	<u>ETX</u>	<u>EOT</u>	<u>ENQ</u>	<u>ACK</u>	<u>BEL</u>	<u>BS</u>	<u>TAB</u>	<u>LF</u>	<u>VT</u>	<u>FF</u>	<u>CR</u>	<u>SO</u>	<u>SI</u>
1-	<u>DLE</u>	<u>DC1</u>	<u>DC2</u>	<u>DC3</u>	<u>DC4</u>	<u>NAK</u>	<u>SYN</u>	<u>ETB</u>	<u>CAN</u>	<u>EM</u>	<u>SUB</u>	<u>ESC</u>	<u>FS</u>	<u>GS</u>	<u>RS</u>	<u>US</u>
2-	<u>SP</u>	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3-	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4-	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5-	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6-	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7-	p	q	r	s	t	u	v	w	x	y	z	{		}	~	<u>DEL</u>
8-	€		,	f	„	…	†	‡	ˆ	‰	Š	‹	Œ		Ž	
9-		‘	’	“	”	•	–	—	~	™	š	›	œ		ž	ÿ
A-	<u>NBSP</u>	ı	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	<u>SHY</u>	®	¯
B-	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C-	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D-	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E-	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F-	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Extended character test set

The following test set for extended font data writes the extended character set from address 0x80 .. 0xFF to the display. This test set is using recipient address "00" = 0x3030

```

0x08..0x8F : 01 30 30 30 30 57 47 01 30 80 81 82 83 84 85 86 87 88 99 03
0x09..0x9F : 01 30 30 30 30 57 47 01 30 90 91 92 93 94 95 96 97 98 99 03
0x0A..0xAF : 01 30 30 30 30 57 47 01 30 A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 03
0x0B..0xBF : 01 30 30 30 30 57 47 01 30 B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 03
0x0C..0xCF : 01 30 30 30 30 57 47 01 30 C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 03
0x0D..0xDF : 01 30 30 30 30 57 47 01 30 D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 03
0x0E..0xEF : 01 30 30 30 30 57 47 01 30 E0 E1 E2 E3 E4 E5 E6 E7 E8 E9 03
0x0F..0xFF : 01 30 30 30 30 57 47 01 30 F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 03
    
```

Terminal program

Convenient terminal program under windows : <http://freeware.the-meiers.org>

This terminal program supports a hexadecimal input screen that makes it easy for testing