



# Fixed & User-Defined Character Sets Application Note

# TABLE OF CONTENTS

- General Description.....3**
- Character Sets.....3**
  - Fixed Character Sets 0 and 1.....3
  - User-Defined Character Set 2.....4
  - User-Defined Character Set 3.....12
- Command Examples.....18**
- Combining Text and Graphics.....19**
- Notices.....20**
  - Copyright Notice.....20
  - Technical Notice.....20
  - Warranty Disclaimer.....20
- Change History.....21**

# GENERAL DESCRIPTION

The CC0064 Controller allows you to control Sxnnnn switches. Besides downloading bitmap images you may wish to write text directly to the display or a portion thereof. To allow this the CC0064 controller contains four character sets, 2 fixed and up to 2 user-defined.

The fixed internal character sets are called **Charset0** and **Charset1**. The two optional user-defined character sets are designated **Charset2** and **Charset3**.

The character sets allow you to write text strings of a given length to a specific location on the LCD display by using the 0x64 command (see below).

## CHARACTER SETS

Character sets are defined as fixed segment text displays within the LCD pixel matrix as either a 5x8 or 10x16 pixel font. Characters are written in the position specified with the increment value of the command (default increment = 0 must be specified as part of the 0x64 command) and auto-incremented. See Command Examples below.

The character sets are valid for all Sxnnnn LCD keys regardless of resolution. Consequently, you can fit 6 characters of Charset0 and Charset2 into the 36x24 pixel matrix of a SA3624 key while 10 characters of the same character sets fit into the matrix of a SA6432 switch.

### Fixed Character Sets 0 and 1

!"#\$%&'()*+,-./ 0123456789:;<=>? @ABCDEFGHIJKLMNO PQRSTUVWXYZ[\]^_ `abcdefghijklmno pqrstuvwxyz{ }~  i¢£¥¦§¨©ª«¬®¯ °±²³´µ¶·¸¹º»¼½¾¿ ÀÁÂÃÄÅÆÇÈÉÊËÌÍÎ ÏÐÑÒÓÔÕÖ×ØÙÚÛÜÝÞß	!"#\$%&'()*+,-./ 0123456789:;<=>? @ABCDEFGHIJKLMNO PQRSTUVWXYZ[\]^_ `abcdefghijklmno pqrstuvwxyz{ }~  i¢£¥¦§¨©ª«¬®¯ °±²³´µ¶·¸¹º»¼½¾¿ ÀÁÂÃÄÅÆÇÈÉÊËÌÍÎ ÏÐÑÒÓÔÕÖ×ØÙÚÛÜÝÞß
--	--

Character sets 0 and 1 (see below) are fixed and cannot be changed. The 0x64 command (see Command Examples) is used to write the text to the specified location in the switch.

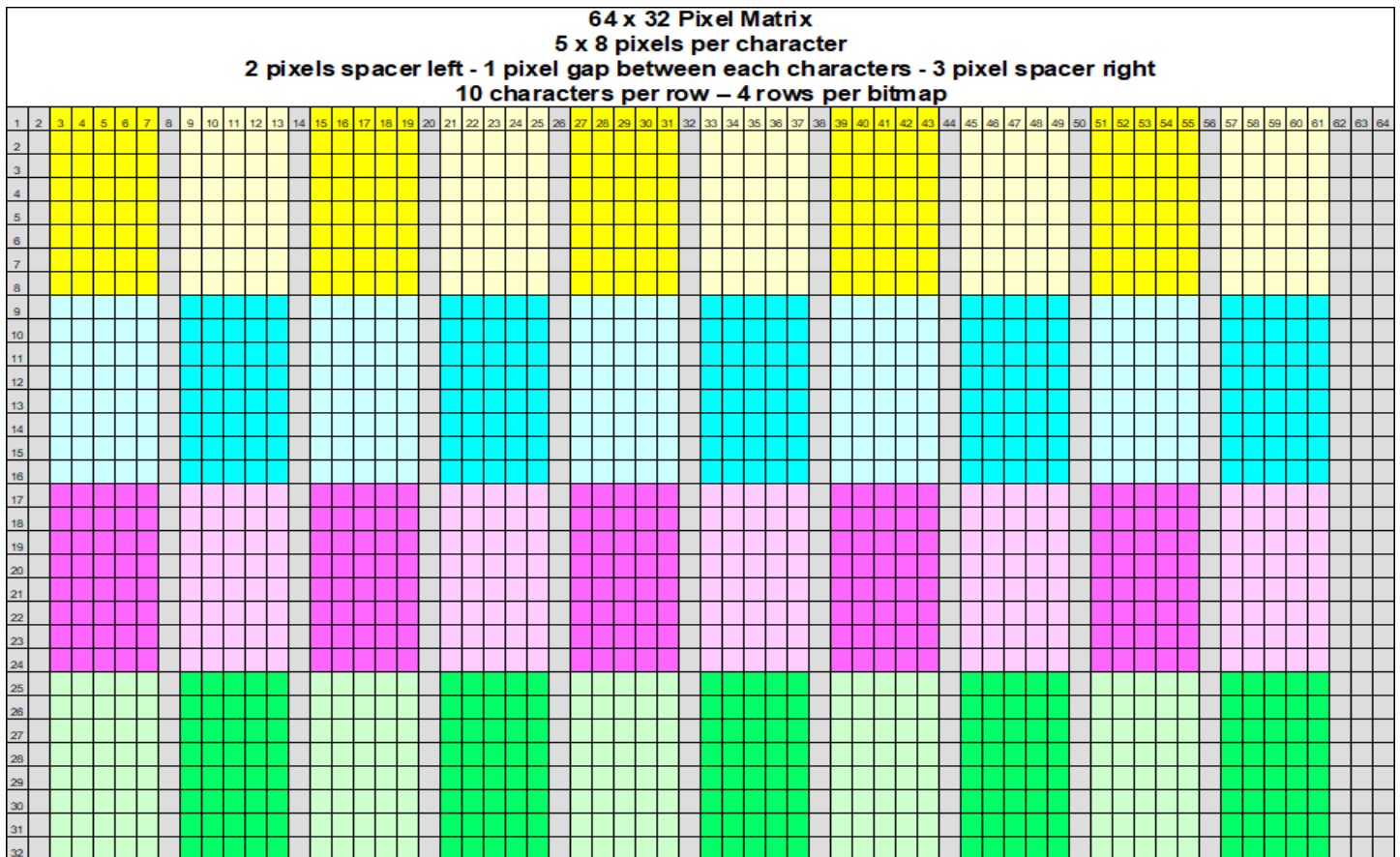
# User-Defined Character Set 2

The **Charset2** is defined as a 5 x 8 pixel character set and uses memory locations 0xEB...0xEF to store 192 characters from the **ISO/IEC 8859-1 (Latin-1)** code page.

The characters for **Charset2** are assigned by their position within a 64 x 32 pixel black & white monochrome bitmap stored in the corresponding memory locations of the controller. In the example below, the 192 characters are defined and saved in bitmaps Charset2\_EB.bmp to Charset2\_EF.bmp based on the Times New Roman Regular font.

Bitmap				
Charset2_EB.bmp	Charset2_EC.bmp	Charset2_ED.bmp	Charset2_EE.bmp	Charset2_EF.bmp
!"#\$%&'()* *+,-./0123 456789:;<= >?@ABCDEFGHI	HJKLMNO PQRSTUVWXYZ [ \ ^ _ ` abc defghijklm no	pqrstuvxy z{ }~" € ¢ ¤ ¥ ¦ § ¨ © ª « ¬ ® ¯ ± ² ³ ´ µ ¶ · ¸ ¹ º » ¼	~ ¯ ¸ ¹ º » ¼ ½ ¾ ¿ @ ¤ ¥ ¦ § ¨ © ª « ¬ ® ¯ ± ² ³ ´ µ ¶ · ¸ ¹ º » ¼ ½	AAAAAAECDE EFFFFIJD 00 0000000000 P0
Memory Location				
0xEB	0xEC	0xED	0xEE	0xEF

The layout of the bitmap follows the following pattern:





Set Key:1

Transmit Bitmap: M:\E3 GmbH\Marketing - RW\Documentation\Character Sets\charset2\_EB.bmp

```
0x40 * Set Display address and write display data
; * Data Bytes to follow 512 each containing just one nibble
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 ;* -
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 ;* -
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 ;* -
0x04 0x01 0x08 0x00 0x00 0x00 0x00 0x00 ;* - # # #
0x04 0x01 0x0B 0x03 0x0E 0x03 0x02 0x03 ;* - ##### # ## # # ## ##
0x04 0x01 0x0A 0x04 0x01 0x04 0x0D 0x04 ;* - # # # ## # # # # # #
0x04 0x01 0x01 0x04 0x00 0x00 0x09 0x04 ;* - # # # # # # # #
0x04 0x01 0x0E 0x03 0x00 0x00 0x01 0x04 ;* - # # # # # #####
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 ;* -
0x02 0x02 0x03 0x00 0x00 0x00 0x00 0x00 ;* - # # ##
0x04 0x01 0x0D 0x01 0x00 0x00 0x0E 0x06 ;* - ### ## # # # ###
0x04 0x01 0x09 0x04 0x01 0x04 0x01 0x05 ;* - # # # # # # # # #
0x04 0x01 0x0F 0x07 0x0E 0x03 0x01 0x06 ;* - ##### # ## # # #####
0x08 0x00 0x01 0x04 0x00 0x00 0x02 0x04 ;* - # # # # # #
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 ;* -
0x00 0x00 0x03 0x06 0x00 0x00 0x00 0x00 ;* - ## ##
0x00 0x00 0x0D 0x05 0x00 0x00 0x00 0x04 ;* - # # ### #
0x04 0x06 0x09 0x04 0x07 0x00 0x0F 0x07 ;* - ### ##### # ## # # #
0x00 0x0A 0x0F 0x07 0x00 0x00 0x01 0x04 ;* - # # # #####
0x00 0x00 0x01 0x04 0x00 0x00 0x00 0x00 ;* - # # #
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 ;* -
0x00 0x00 0x0C 0x01 0x0C 0x04 0x00 0x00 ;* - ## # ###
0x00 0x00 0x02 0x02 0x03 0x03 0x0E 0x03 ;* - ## ## ##### # #
0x04 0x04 0x01 0x04 0x0D 0x05 0x01 0x04 ;* - # ## # # # # # # #
0x00 0x00 0x0F 0x07 0x0E 0x04 0x01 0x04 ;* - ### # # # #####
0x00 0x00 0x01 0x04 0x00 0x03 0x0E 0x03 ;* - ## ##### # #
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 ;* -
0x00 0x00 0x03 0x02 0x00 0x03 0x00 0x00 ;* - ## ## ## #
0x0E 0x01 0x02 0x04 0x03 0x03 0x03 0x00 ;* - ## ## ## ##### # #
0x01 0x03 0x01 0x04 0x0C 0x00 0x0C 0x01 ;* - ## ### ## # # # #
0x01 0x05 0x01 0x04 0x06 0x03 0x00 0x06 ;* - ## ## ## # # # # #
0x0E 0x04 0x0E 0x03 0x06 0x04 0x00 0x00 ;* - ## # ### # #####
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 ;* -
0x00 0x00 0x06 0x03 0x04 0x02 0x00 0x00 ;* - # # ## ##
0x06 0x03 0x09 0x04 0x02 0x05 0x00 0x00 ;* - # # # ## # # # #
0x09 0x04 0x09 0x04 0x0F 0x0F 0x00 0x04 ;* - ##### # # # # # # #
0x09 0x04 0x0F 0x07 0x0A 0x04 0x00 0x00 ;* - # # # # # # #####
0x06 0x03 0x01 0x04 0x0C 0x02 0x00 0x00 ;* - ## # ## ## # #
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 ;* -
0x00 0x00 0x00 0x00 0x07 0x07 0x01 0x00 ;* - ### # ###
0x03 0x00 0x0E 0x01 0x0C 0x07 0x08 0x00 ;* - ##### # ## #####
0x0D 0x01 0x01 0x01 0x04 0x01 0x08 0x00 ;* - # # # # ## # #
0x01 0x06 0x0E 0x01 0x0F 0x01 0x08 0x00 ;* - ##### # # ## #####
0x02 0x00 0x00 0x00 0x07 0x04 0x07 0x00 ;* - # ### # #####
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 ;* -
0x00 0x00 0x0E 0x05 0x00 0x00 0x00 0x00 ;* - ##### #
0x09 0x03 0x05 0x05 0x07 0x00 0x00 0x00 ;* - ### # ## # # # #
0x05 0x04 0x09 0x05 0x00 0x00 0x00 0x06 ;* - ## # # # # ## #
0x06 0x04 0x02 0x02 0x07 0x00 0x00 0x0A ;* - ### # # ## # # #
0x0C 0x03 0x0C 0x01 0x00 0x00 0x00 0x00 ;* - ##### ##
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 ;* -
0x00 0x00 0x00 0x00 0x00 0x00 0x08 0x00 ;* - #
0x09 0x03 0x06 0x00 0x00 0x00 0x08 0x00 ;* - # # ### ##
0x05 0x04 0x09 0x05 0x0F 0x05 0x0E 0x03 ;* - ##### # ##### # # # # # #
0x06 0x04 0x02 0x00 0x00 0x00 0x08 0x00 ;* - # ## # #
0x00 0x04 0x00 0x00 0x00 0x00 0x08 0x00 ;* - # #
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 ;* -
0x00 0x00 0x08 0x00 0x00 0x00 0x0A 0x00 ;* - # # #
0x00 0x01 0x04 0x01 0x00 0x00 0x04 0x00 ;* - # # # #
0x0F 0x07 0x04 0x01 0x00 0x00 0x0F 0x00 ;* - ##### ##### # #
0x06 0x01 0x04 0x01 0x00 0x00 0x04 0x00 ;* - # ## # # #
0x08 0x01 0x02 0x02 0x00 0x00 0x0A 0x00 ;* - # # ## # #
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 ;* -
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 ;* -
* - End of Bitmap
```

0xF5 0xEB 0xFE ; store image in display buffer in bitmap library location 0xEB and terminate command

Repeat for the other bitmaps of the character set to store them in the respective memory locations..



# User-Defined Character Set 3

The **Charset3** is defined as a 10 x 16 pixel character set and uses memory locations 0xF0...0xFF to store up to 192 ASCII characters from the **ISO/IEC 8859-1 (Latin-1) code page**.

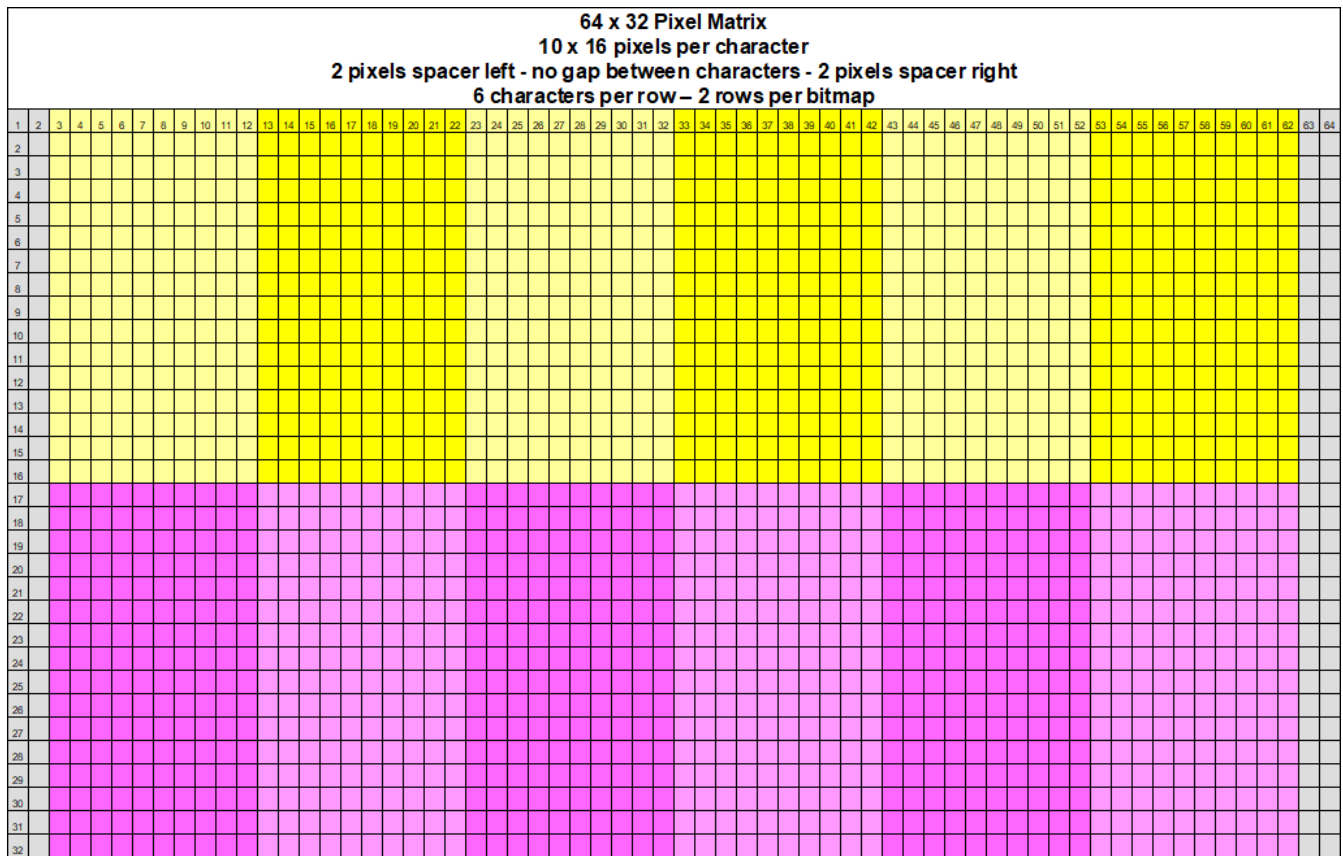
The characters for **Charset3** are assigned by their position within a 64 x 32 pixel black & white monochrome bitmap stored in the corresponding memory locations of the controller. In the example below, the 192 characters are defined based on the *Bahnschrift SemiBold Condensed Italic* font and saved in bitmaps Charset3\_F0.bmp to Charset3\_FF.bmp.

Bitmap							
Charset3_F0.bmp	Charset3_F1.bmp	Charset3_F2.bmp	Charset3_F3.bmp	Charset3_F4.bmp	Charset3_F5.bmp	Charset3_F6.bmp	Charset3_F7.bmp
Memory Location							
0xF0	0xF1	0xF2	0xF3	0xF4	0xF5	0xF6	0xF7

Bitmap							
Charset3_F8.bmp	Charset3_F9.bmp	Charset3_FA.bmp	Charset3_FB.bmp	Charset3_FC.bmp	Charset3_FD.bmp	Charset3_FE.bmp	Charset3_FF.bmp
Memory Location							
0xF8	0xF9	0xFA	0xFB	0xFC	0xFD	0xFE	0xFF



The layout of the bitmap follows the following pattern:



Here is an enlarged view at the corresponding bitmaps that define this character set:



Charset3\_F0.bmp



Charset3\_F1.bmp



Charset3\_F2.bmp



Charset3\_F3.bmp



Charset3\_F4.bmp



Charset3\_F5.bmp



Charset3\_F6.bmp



Charset3\_F7.bmp



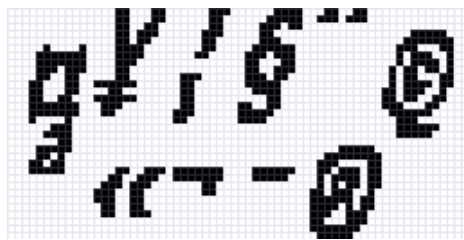
Charset3\_F8.bmp



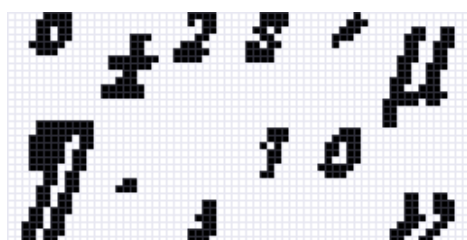
Charset3\_F9.bmp



Charset3\_FA.bmp



Charset3\_FB.bmp



Charset3\_FC.bmp



Charset3\_FD.bmp



Charset3\_FE.bmp



Charset3\_FF.bmp

The bitmaps must be stored in the correct memory location. In order to do this, the bitmap must first be sent to an LCD key using the 0x40 command. Once the bitmaps has been transmitted (and while still in the image buffer of the controller) it can be stored in the bitmap library using the 0xF5 command.

The command sequence for bitmap Charset2\_F0.bmp sent to Key number 1 and then stored in memory location 0xEB is shown below. The procedure is the same for all subsequent character set bitmaps for both Charset2 and Charset3.

```
Set Key :1
Transmit Bitmap: Charset3_F0.bmp
0x40          * Set Display address and write display data
;            * Data Bytes to follow 512 each containing just one nibble
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 ;* -
```

```

0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 ;* -
0x00 0x00 0x00 0x00 0x03 0x00 0x00 0x00 ;* - ##
0x00 0x00 0x00 0x00 0x0E 0x00 0x0E 0x00 ;* - ###
0x00 0x08 0x01 0x00 0x0C 0x01 0x0E 0x07 ;* - #####
0x00 0x0F 0x01 0x00 0x00 0x03 0x06 0x0E ;* - ## ## ## #####
0x00 0x0F 0x03 0x00 0x00 0x0E 0x0E 0x0C ;* - ### ## ## #####
0x00 0x0C 0x0F 0x00 0x0F 0x0C 0x0D 0x0F ;* - ##### ## #####
0x00 0x08 0x0F 0x00 0x0F 0x03 0x03 0x0E ;* - ##### ## ## #####
0x00 0x08 0x01 0x00 0x03 0x06 0x0E 0x00 ;* - ## ## ## ##
0x00 0x00 0x00 0x00 0x0E 0x07 0x0C 0x01 ;* - ##### ## ##
0x00 0x00 0x00 0x00 0x0C 0x07 0x00 0x03 ;* - ##### ##
0x00 0x00 0x00 0x00 0x08 0x03 0x00 0x0E ;* - ## ## ##
0x00 0x00 0x00 0x00 0x0D 0x07 0x00 0x00 ;* - # #####
0x00 0x00 0x00 0x00 0x0F 0x07 0x02 0x00 ;* - ##### #
0x04 0x00 0x00 0x00 0x00 0x0F 0x00 0x07 ;* - ##### ## #
0x07 0x03 0x00 0x00 0x0C 0x00 0x0F 0x00 ;* - ## ##### ## #
0x0F 0x03 0x00 0x00 0x08 0x01 0x09 0x01 ;* - ## # ## #####
0x0C 0x01 0x00 0x00 0x00 0x08 0x0F 0x01 0x03 ;* - ##### ## ##
0x04 0x03 0x00 0x00 0x00 0x0E 0x00 0x07 ;* - ## ## # ##
0x00 0x00 0x00 0x00 0x00 0x00 0x04 0x0F ;* - # #####
0x00 0x00 0x00 0x00 0x00 0x00 0x0C 0x0F ;* - #####
0x00 0x00 0x00 0x00 0x07 0x03 0x0C 0x01 ;* - ## ## ##
0x00 0x00 0x00 0x00 0x0E 0x07 0x00 0x00 ;* - #####
0x00 0x00 0x00 0x00 0x09 0x0F 0x0D 0x00 ;* - # #####
0x00 0x0C 0x03 0x00 0x07 0x0B 0x0F 0x00 ;* - ## ## #####
0x08 0x0F 0x0F 0x01 0x0E 0x03 0x0E 0x01 ;* - ##### ## #####
0x0E 0x03 0x0C 0x07 0x08 0x07 0x0C 0x07 ;* - ##### ##### #####
0x0F 0x00 0x00 0x0E 0x00 0x0F 0x0D 0x0E ;* - ##### ## #####
0x01 0x00 0x00 0x08 0x00 0x0B 0x0F 0x08 ;* - ## ##### ## #
0x00 0x00 0x00 0x00 0x00 0x00 0x0E 0x01 ;* - #####
0x00 0x00 0x00 0x00 0x00 0x00 0x0C 0x0F ;* - #####
0x00 0x00 0x00 0x00 0x00 0x00 0x0C ;* - ##
0x00 0x00 0x00 0x00 0x00 0x00 0x00 ;* -
0x01 0x00 0x00 0x08 0x0F 0x01 0x00 0x00 ;* - ##### # #
0x0F 0x00 0x00 0x0E 0x0E 0x01 0x00 0x00 ;* - ##### ##### ##
0x0E 0x03 0x0C 0x07 0x03 0x01 0x00 0x00 ;* - ## # #####
0x08 0x0F 0x0F 0x01 0x0F 0x01 0x00 0x00 ;* - ##### #####
0x00 0x0C 0x03 0x00 0x0E 0x01 0x00 0x00 ;* - #####
0x00 0x00 0x00 0x00 0x00 0x01 0x00 0x00 ;* - #
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 ;* -
0x00 0x00 0x00 0x00 0x00 0x00 0x00 ;* -
0x00 0x00 0x00 0x00 0x01 0x00 0x00 0x00 ;* - #
0x00 0x00 0x00 0x00 0x0F 0x00 0x00 0x00 ;* - #####
0x03 0x00 0x00 0x00 0x0F 0x07 0x00 0x00 ;* - ##### ##
0x0F 0x01 0x00 0x00 0x00 0x08 0x0F 0x03 0x00 ;* - ##### #####
0x0E 0x01 0x00 0x00 0x00 0x0C 0x0F 0x00 ;* - ##### #####
0x00 0x01 0x00 0x00 0x00 0x00 0x0E 0x0C ;* - ## ## #
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x0C ;* - ##
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x0C ;* - ##
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 ;* -
0x0F 0x01 0x07 0x00 0x00 0x00 0x00 0x00 ;* - ##### ##
0x0F 0x01 0x0F 0x09 0x00 0x00 0x00 0x00 ;* - ##### ##### #
0x03 0x00 0x08 0x0F 0x00 0x00 0x00 0x00 ;* - ## #####
0x0E 0x00 0x0F 0x0F 0x00 0x00 0x00 0x00 ;* - ## #####
0x0E 0x0F 0x0F 0x07 0x00 0x00 0x00 0x00 ;* - #####
0x08 0x0F 0x07 0x0C 0x00 0x00 0x00 0x00 ;* - ##### ##
0x00 0x00 0x03 0x0C 0x00 0x00 0x00 0x00 ;* - ## ##
0x00 0x00 0x06 0x0C 0x00 0x00 0x00 0x00 ;* - ## ##
0x00 0x00 0x0E 0x0F 0x00 0x00 0x00 0x00 ;* - #####
0x00 0x00 0x08 0x0F 0x00 0x00 0x00 0x00 ;* - #####
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 ;* -
* - End of Bitmap
0xF5 0xF0 0xFE ; store image in display buffer in bitmap library location 0xEB and terminate command

```

Repeat for the other bitmaps of the character set to store them in the respective memory locations.

You can now use the characters stored in the bitmaps to write text to the LCD keys using the 0x64 command as described in the Command Examples below. The following tables show the ASCII characters, hex codes as well as the memory locations and bitmap names for this Charset2. Please note that in the **ISO/IEC 8859-1 (Latin-1)** code page, Hex codes 80 to 9F are not defined and the corresponding characters, therefore, shifted to addresses A0 to FF

20				2C	,			38	8			44	D		
21	!			2D	-			39	9			45	E		
22	"			2E	.			3A	:			46	F		
23	#			2F	/			3B	;			47	G		
24	\$			30	0			3C	<			48	H		
25	%			31	1			3D	=			49	I		
26	&	0xF0		32	2	0xF1		3E	>	0xF2		4A	J	0xF3	
27	'		charset3_F0.bmp	33	3		charset3_F0.bmp	3F	?		charset3_F0.bmp	4B	K		charset3_F0.bmp
28	(			34	4			40	@			4C	L		
29	)			35	5			41	A			4D	M		
2A	*			36	6			42	B			4E	N		
2B	+			37	7			43	C			4F	O		
50	P			5C	\			68	h			74	t		
51	Q			5D	]			69	i			75	u		
52	R			5E	^			6A	j			76	v		
53	S			5F				6B	k			77	w		
54	T			60				6C	l			78	x		
55	U	0xF4		61	a	0xF5		6D	m	0xF6		79	y	0xF7	
56	V		charset3_F0.bmp	62	b		charset3_F0.bmp	6E	n		charset3_F0.bmp	7A	z		charset3_F0.bmp
57	W			63	c			6F	o			7B	{		
58	X			64	d			70	p			7C			
59	Y			65	e			71	q			7D	}		
5A	Z			66	f			72	r			7E	~		
5B	[			67	g			73	s			7F			
A0	€			AC	œ			B8	ˆ			C4	α		
A1				AD				B9	™			C5	¥		
A2				AE	ž			BA	š			C6	ı		
A3	f			AF				BB	›			C7	š		
A4	„			B0				BC	œ			C8	ı		
A5	…			B1	˙	0xF9		BD	œ	0xFA		C9	©	0xFB	
A6	†		charset3_F0.bmp	B2	˘		charset3_F0.bmp	BE	ž		charset3_F0.bmp	CA	ª		charset3_F0.bmp
A7	‡			B3	“			BF	Y			CB	«		
A8	•			B4	”			C0				CC	˘		
A9	‰			B5	•			C1	ı			CD	-		
AA	Š			B6	—			C2	¢			CE	®		
AB	‹			B7	—			C3	£			CF	ı		
D0	°			DC	¼			E8	È			F4	Ó		
D1	±			DD	½			E9	É			F5	Ô		
D2	²			DE	¾			EA	Ê			F6	Ö		
D3	³			DF	¿			EB	Ë			F7	×		
D4	´			E0	À			EC	Ì			F8	Ø		
D5	µ	0xFC		E1	Á	0xFD		ED	Í	0xFE		F9	Ù	0xFF	
D6	¶		charset3_F0.bmp	E2	Â		charset3_F0.bmp	EE	Î		charset3_F0.bmp	FA	Ú		charset3_F0.bmp
D7	·			E3	Ã			EF	Ï			FB	Û		
D8	¸			E4	Ä			F0	Ð			FC	Ü		
D9	¹			E5	Å			F1	Ñ			FD	Ý		
DA	º			E6	Æ			F2	Ò			FE	Þ		
DB	»			E7	Ç			F3	Ó			FF	ß		

You can download this Charset3 [here](#) together with an e3t file that transmits the bitmaps to the corresponding library address.

# COMMAND EXAMPLES


Write Text: 01100100(0x64)	
HEX	Comments
(0x64)	Write text string of a given length with specified offset from character set to selected key.
0xFF 0x01 0x64 0x04 0x02 0x01 0x61 0x62 0x63 0x64	Select key 1 and Write 4 characters long text "abcd" with 2 characters offset using Charset1
0xFF 0x02 0x64 0x0C 0x00 0x03 0x48 0x65 0x6C 0x6C 0x6F 0x20 0x57 0x6F 0x72 0x6C 0x64 0x21	Select key 2 and Write 12 characters long text "Hello World!" with 0 offset using Charset3
0xFF 0x03 0x64 0x05 0x00 0x01 0x48 0x65 0x6C 0x6C 0x6F 0x64 0x06 0x06 0x03 0x57 0x6F 0x72 0x6C 0x64 0x21	Select key 3 and Write 5 characters long text "Hello" with 0 offset using Charset1 and 6 characters long text "World!" with 6 offset using Charset3

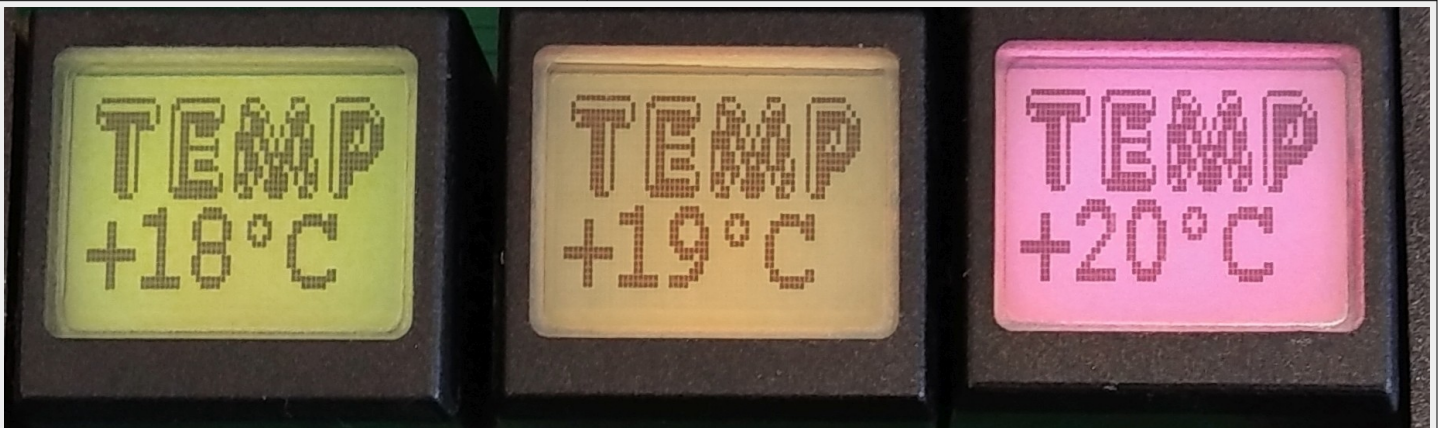
# COMBINING TEXT AND GRAPHICS

When transmitting bitmaps or text to the LCD keys the data is first stored in the graphics buffer of the controller and then sent to the LCD display of the keys. The data remains in the graphics buffer after transmission until it is overwritten. This means that you can overwrite a portion of that bitmap with text characters. The offset will not affect the previous bitmap “background”.

Below is an example of a bitmap creating a label for a temperature readout. The bitmap has previously stored in the bitmap library using the 0xF5 command. The TEMP.bmp defines a label \*top part of the bitmap) and the readout in degrees Centigrade (bottom right half of bitmap) as a graphic leaving an open area in the bottom left of the bitmap for the actual temperature readout, which can be changed dynamically.

In our example, the bitmap is stored in the image library and the incremental readings “+18”, “+19” and “+20” are written into the display after the TEMP bitmap has been displayed. We are also incrementally changing the background color to provide color context for the increase in temperature. You may want to add a delay to better see the change.

HEX	Comments
0xFF 0x04	select key 4
0x42 0x7F 0x7F 0x7F 0x43	change color
0xF4 0x45 0xFE	write bitmap from memory location 0x45 
0x42 0x40 0x40 0x00 0x43	change color
0x64 0x03 0x06 0x03 0x2B 0x31 0x38	write “+18” with 6 char offset in charset3
0x42 0x40 0x10 0x00 0x43	change color
0x64 0x03 0x06 0x03 0x2B 0x31 0x39	write “+19” with 6 char offset in charset3
0x42 0x7F 0x01 0x01 0x43	change color
0x64 0x03 0x06 0x03 0x2B 0x32 0x30	write “+20” with 6 char offset in charset3



# NOTICES

## Copyright Notice

© 2022 Copyright [E<sup>3</sup>] Engstler Elektronik Entwicklung GmbH. All rights reserved.

[E<sup>3</sup>], *The Third Evolution*<sup>™</sup> and *Legacy Mode*<sup>™</sup> are trademarks of [E<sup>3</sup>]. *The Keys to Intelligence*<sup>™</sup> is a trademark of I/O Universal Technologies, Inc. used with permission. All other trademarks are property of their respective owners.

No part of this publication may be copied, photocopied, reproduced, translated or reduced to any electronic medium or machine-readable form without the expressed written consent of [E<sup>3</sup>] Engstler Elektronik Entwicklung GmbH.

## Technical Notice

This datasheet is intended for technically qualified personnel trained in the field of electronics.

The knowledge of electronics and the technically correct implementation of the content of this datasheet are required for problem free installation, implementation and safe operation of the described product. Only qualified personnel have the required know-how to implement the specifications given in this data sheet.

For clarity, not all details regarding the product or its implementation, installation, operation, or maintenance have been included. Should you require additional information or further assistance, please contact your local [E<sup>3</sup>] distributor or [E<sup>3</sup>] Engstler Elektronik Entwicklung GmbH at [techsupport@e3-keys.com](mailto:techsupport@e3-keys.com). You may also visit our website at [www.e3-keys.com](http://www.e3-keys.com).

## Warranty Disclaimer

[E<sup>3</sup>] ENGSTLER ELEKTRONIK ENTWICKLUNG GMBH grants no warranty with respect to this data sheet, neither explicit nor implied, and it is not liable for direct or indirect damages. Some states do not grant the exclusion of incidental or consequential damages and, therefore, this statement may not be valid in such cases.

This data sheet has been produced with all due care. However, since errors cannot be excluded, [E<sup>3</sup>] Engstler Elektronik Entwicklung GmbH does not grant any warranty or accept any legal responsibility or liability in any form for erroneous statements herein.

# CHANGE HISTORY

Version	Date	Comments
0.1	10/30/19	Draft document
0.2	06/30/20	Updated formatting
1.0	01/25/22	Release version
1.1	02/04/22	Bitmaps, ASCII tables added
1.2	02/10/22	Command examples added
2.0	06/16/22	Updated release version

**[E<sup>3</sup>] Engstler Elektronik Entwicklung GmbH**  
Industriering 7 • 63868 Grosswallstadt • Germany  
[WWW.E3-KEYS.COM](http://WWW.E3-KEYS.COM)