Rev.1.8

THIS DOCUMENT DESCRIBES COMMAND SET FOR MULTICOMMUNICATION LED DISPLAY CONTROLLER MCA2X16 DRIVING 7 SEGMENT, ALPHANUMERIC 14, 16, 17 SEGMENT, DOT MATRIX, LED BARS AND SINGLE DIODES

PROTOCOL

MCA2X16 receive commands via serial channel. Host sends commands and data to LED controller. When command is completed, LED controller sends answer to HOST notifying that current command is completed and can accept new command. Until current command is not completed MCA2X16 do not accept any additional data. Time completion of commands varies, depending of type of command and amount of received or transmitted data.

START UP

After power up LED Controller **MCA2X16** sends message depending of mounted interface board as follows:

a. When is mounted interface board **CB232**, message to HOST after restart or power up is "MCA2X16-RS232C".

b. When is mounted interface board **CB485**, no message to HOST is issued to prevent other **MCA2X16** LED controllers from buss collision. Activity led only will blink once for short time (typically 200 mS) or may blink with frequency of about 1 Hz if cable is disconnected.

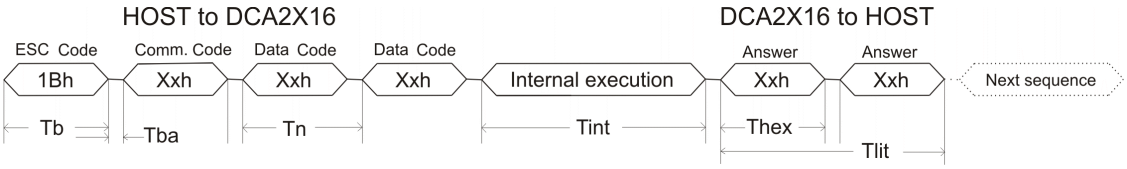
c. When is mounted interface board **CB100**, message to HOST after restart or power up is "MCA2X16-RFCOMM" that notifies HOST that Bluetooth profile RFCOMM is loaded.

d. When is mounted interface board **CB203**, message to HOST after restart or power up is "MCA2X16-TTL".

Start up messages notifies HOST that LED controller is supplied, start up correctly and ready to accept command. User must follow exact sequence for transmitting of command and data described below.

LED controller maintains two types of timeouts. First one is more than 500 mS and second one is 50 mS. As example if HOST sends command 'TEST ' 1Bh 30h, if time between sign ESCAPE and command code 30h exceed 500mS, LED DISPLAY will send to HOST message "TIMEOUT ERROR" and will cancel completing of current command and will be ready to accept new command. All commands is consists least from two bytes. Second timeout is used to define end of received block of data or ASCII strings. Time for transmit between two sequential bytes in string must be least 50 ms. If this time is larger than 50 mS, LED Controller will accept this delay as end of received block. If the next command is issued immediately after transmitting the string, MCA2X16 will treat command byte as a part of string and will display this code.

Command newer be performed. Good practice is to set a delay least 50ms when string is transmitted before issuing the next command. Next time diagram point main periods for completion of commands.



Tb – Period for transmitting one byte (depended from baud rare selected)
Tba - min period time between two transmitted bytes
Tn - number of transmitted byte depending of typical command
Tint - Internal execution time after last received byte (this period vary from command to command and is published for each command separately and not dependent from selected baud rate.
Thex=Tb+Tba – Hexadecimal answer period. This is answer byte, issued from LED Controller , to inform the HOST, that current command is completed and MCA2X16 can receive another one. This period is useful to set delay, when HOST does not perform check for completed command.
Tlit=(Tb+Tba) x Tn - Literal answer period. LED Controller sends answer to HOST like ASCII string.

Table describes period value in different cycles of receiving command.

	min	typical	max
Tb - 9600 bps	**889.2 mkS**	**936 mkS**	**982 mkS**
Tb - 19200 bps	**444,6 mkS**	**468 mkS**	**491 mkS**
Tb - 38400 bps	**222.3 mkS**	**234 mkS**	**245,7 mks**
Tb - 57600 bps	**148 mkS**	**155,8 mkS**	**163,5 mkS**
Tba – 9600 bps	**-**	**104 mkS**	**45 mS**
Tba – 19200 bps	**-**	**52 mkS**	**45 mS**
Tba – 38400 bps	**-**	**26 mkS**	**45 mS**
Tba – 57600 bps	**-**	**17,4 mkS**	**45 mS**

Legend:

MCA2X16	**-**	LED controller sends this message after start up depending of interface
HOST	**-**	User control system (PC or some user controller)
SET	**-**	Mean set logic level equal to 1.
CLEAR	**-**	Mean set logic level equal to 0.
LIT	**-**	Literal Message Mode (to Host)
HEX	**-**	Hexadecimal Message Mode (to Host)
h (xx)h	**-**	Hexadecimal Value
d (xx)d	**-**	Decimal value
b (xx)b	**-**	Binary value
(xx)	**-**	Value
(ad)	**-**	Address value
(db)	-	Data byte value
(xx) par	**-**	Parametric byte

Messages issued from MCA2X16:

Literal mode	Hexadecimal mode
OK	**4Fh - (O) ASC II**
SET	**53h - (S) ASC II**
DONE	**44h - (D) ASC II**
ERROR	**45h - (E) ASC II**
CLEAR	**43h - (C) ASC II**
WRONG VALUE	**57h - (W) ASC II**
TIMEOUT ERROR	**54h - (T) ASC II**
INCORRECT VALUE	**55h - (U) ASC II**
PARAMETER MISSING	**50h - (P) ASC II**
WRONG COMMAND CODE	**52h - (R) ASC II**
ADDRESS IS OUT OF RANGE	**41h - (A) ASC II**
THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG	**74h - (t) ASC II**

No	ASC II VALUE	HEX. VALUE	COMMAND
1.	**ESC 0**	**1Bh 30h**	**Communication test**

Command test communication between HOST and LED controller

Command test communication between LED controller and HOST. LED controller answer in HEX message mode with (74h) - (t) ASC II, or in Literal message mode MCA2X16 LED Controller sends to **HOST** sentence: "**THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG**"(without quotation). When **HOST** receive correct answer, this means that cables and baud rate are selected correctly.

	HOST – LED DISPLAY	**LED DISPLAY to HOST**		
ASCII	**ESC 0**	**THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG**		
HEX	**1Bh 30h**	**74h - (t) ASC II**		
		MIN	**TIPICAL**	**MAX**
Tint – internal execution period		-	1,8 mkS	-

No	ASC II VALUE	HEX. VALUE	COMMAND
2.	**ESC 1(xx)h**	**1Bh 31h (xx)h**	Set PWM Value

Command sets value of PWM

This command adjusts light intensity of the LED display. It uses PWM for driving Anodes for appropriate period for each cycle of refresh. Value of parametric byte is between (00h) to (62h) and refer as follows PWM=(00h) - dark display (PWM=0%), PWM=(62h) – Maximum Intensity (PWM=99%). Also value of PWM can be saved in non-volatile memory. After power-up LED controller will use the last saved value of PWM. Refer to command No 13. If no parametric byte is present after command code, LED Controller sends to HOST the last valid value of PWM. Parametric byte can be in range between 00h to 62h. All other values will be ignored from LED controller, and HOST will receive error message in HEX message Mode (57h) or in Literal message Mode"**WRONG VALUE!!!**" (Without quotation). LED Controller will use previous valid PWM value. Changing of PWM value acts to the both LINES.

	HOST – LED DISPLAY	**LED DISPLAY to HOST**		
ASCII	**ESC 1 (XX)h**	**(XX)h**		
HEX	**1Bh 31h (XX)h**	**(XX)h**		
		MIN	**TIPICAL**	**MAX**
Tint – internal execution period		-	3,35 mkS	-

No	ASC II VALUE	HEX. VALUE	COMMAND
3.	**ESC 2(xx)h**	**1Bh 32h (xx)h**	Clear LED Buffer

Command clear LED Buffers

This command clears LED buffers of LINE1 and LINE2, depending of selected bits in STATUS byte. Before issuing this command, please select correct bit in STATUS byte (commands No10 and 11) for LINE SELECT. Value after the command code can also be use for testing of connected segments of LED indicators. Fill value can be hexadecimal value from 00h-FFh. Actually this command fill with value (xx) selected LED buffers. After completing of command depending of selected message mode, LED controller send to HOST (44h) or '**DONE**' (without quotation).

	HOST – LED DISPLAY	**LED DISPLAY to HOST**		
ASCII	**ESC 2 (XX)h**	**DONE**		
HEX	**1Bh 32h**	**44h - (D) ASC II**		
		MIN	**TIPICAL**	**MAX**
Tint – internal exec. period one LINE		-	222 mkS	225,5 mkS

No	ASC II VALUE	HEX. VALUE	COMMAND
4.	**ESC 3(ad)h (db)h**	**1Bh 33h (ad)h (db)h**	Writes byte in LED Buffer

Command writes byte in LED Buffers

This command takes ability to write data directly into LED buffers, issuing address and data from HOST. Before issuing this command, please select desired bit in STATUS byte for LINE SELECT for correct selection. Address can be in range from (00h) to (9Fh). LED controller accepts only hexadecimal values of address and data. When address exceeds (9Fh), LED controller sends message "**ADDRESS IS OUT OF RANGE**" (without quotation) in Literal message mode or (41h) - (A) ASC II in Hex message mode and cancel completing of command. User can use this command for variety purposes including single diodes or LED bars and for generating Graphical signs and pictures. After completing of command LED controller sends to HOST value of written in LED Buffer data. If no one LED Buffer is selected, MCA2X16 do not send to HOST any value.

	HOST – LED DISPLAY	**LED DISPLAY to HOST**		
ASCII	**ESC 3 (ad)h (db)h**	**(db)h**		
HEX	**1Bh 33h (ad)h (db)h**	**(db)h**		
		MIN	**TIPICAL**	**MAX**
Tint – internal exec. period one LINE		-	3,35 mkS	-

No	ASC II VALUE	HEX. VALUE	COMMAND
5.	**ESC 4 st (ad) h (db... db) h**	**1Bh 34h st (ad) h (db... db) h**	**Writes block in LED Buffer**

Command writes block of bytes in LED Buffers

Using this command, user can write block of data directly into selected LED buffer, issuing Start address and send data block. Before issuing this command, please select correct bit in STATUS byte for LINE SELECT for correct selection. Start address can be in range from (00h) to (9Fh). LED controller accepts only hexadecimal values of start address. If the received block is a larger than LED buffer boundary, LED controller will cancel receiving string and will completing command. Period between two data bytes must be lower than 50 mS. when period is larger than 50 mS, command is automatically completed and LED controller is ready to accept new command. When address exceeds (9Fh), LED controller send message "**ADDRESS IS OUT OF RANGE**" in Literal message mode or (41h) - (A) ASC II and cancel command execution. User can use this command, when controls dot matrix indicators and single diodes or LED bars for generating values of segments and downloading graphical signs and pictures. This command can also be use for clear part of LED buffer. When the last byte is received, depending of selected message mode, LED controller send to HOST (4Fh) in HEX mode or '**OK**' in literal mode.

	HOST – LED DISPLAY	**LED DISPLAY to HOST**		
ASCII	**ESC 4 st(ad)h (db..db)h**	**OK**		
HEX	**1Bh 34h st(ad)h (db..db)h**	**4Fh - (O) ASC II**		
		MIN	**TIPICAL**	**MAX**
Tint – internal exec. period one LINE 1byte		-	2,75 mkS	-

No	ASC II VALUE	HEX. VALUE	COMMAND
6.	**ESC 5 (ad)h..(db)h**	**1Bh 35h (ad)h ..(db)h**	**Read byte from LED Buffer**

Command read byte from LED Buffers

This command read byte from LED buffer and sends value to HOST. Address can be in range from (00h) to (9Fh). LED controller accepts only hexadecimal values of address. When address exceeds (9Fh), MCA2X16 send message "**ADDRESS IS OUT OF RANGE**" in Literal message mode or (41h) - (A) ASC II and cancel completing command. For the end of command is treated when data byte is transmitted from LED controller. LED controller sends byte depending of line select bits in STATUS byte. No message is issued from LED controller. If no one LED Buffer is selected, MCA2X16 do not send to HOST any value.

Note: When the bit-0 and bit-1 in STATUS byte are set together, (both LINE1 and LINE2 are selected), LED controller sends only data byte from LED buffer1.

	HOST – LED DISPLAY	**LED DISPLAY to HOST**		
ASCII	**ESC 5 (ad)h.. (db)h**	**(db)h**		
HEX	**1Bh 35h (ad)h ..(db)h**	**(db)h**		
		MIN	**TIPICAL**	**MAX**
Tint – internal exec. period one LINE		-	2,35 mkS	-

No	ASC II VALUE	HEX. VALUE	COMMAND
7.	**ESC 6 st(ad)h Nmb(XX)h**	**1Bh 36h st(ad)h Nmb(XX)h**	**Read block from LED Buffer**

Command read block of bytes from LED Buffers

This command read block of data from LED buffer. Next byte after command code is Start address of block in LED buffer and the last one is the number of bytes that user wish to receive. When Start address exceeds (A0h), LED controller sends message "ADDRESS IS OUT OF RANGE" in Literal message mode or (41h) - (A) ASC II in Hex message mode and cancel command. For the end for completing of command can be treated when the last data byte is transmitted. No additional message is issued from LED controller. When the bit-0 and bit-1 in STATUS byte are set together, (both LINE1 and LINE2 are selected), LED controller sends only data byte from LED buffer1.

	HOST – LED DISPLAY	**LED DISPLAY to HOST**		
ASCII	**ESC 6 st(ad)h Nmb(xx)h**	**data block (db..db)h**		
HEX	**1Bh 36h st(ad)h Nmb(xx)h**	**data block (db..db)h**		
		MIN	**TIPICAL**	**MAX**
Tint – internal exec. period one LINE		-	2,85 mkS	-

No	ASC II VALUE	HEX. VALUE	COMMAND
8.	**ESC 7**	**1Bh 37h**	**Not used command**

Not used

This command is reserved for future developments.

	HOST – LED DISPLAY	**LED DISPLAY to HOST**
ASCII	**ESC 7**	**Not respond**
HEX	**1Bh 37h**	**Not respond**

No	ASC II VALUE	HEX. VALUE	COMMAND
9.	**ESC 8**	**1Bh 38h**	**Not used command**

Not used

This command is reserved for future developments.

	HOST – LED DISPLAY	**LED DISPLAY to HOST**
ASCII	**ESC 8**	**Not respond**
HEX	**1Bh 38h**	**Not respond**

No	ASC II VALUE	HEX. VALUE	COMMAND
10.	**ESC 9 (XX)h par**	**1Bh 39h (XX)h par**	**WRITE STATUS byte**

Command writes (change) current state of STATUS byte

This command changes a STATUS byte. Command consists of three bytes, issued from HOST. Third one is the new value of STATUS byte. STATUS byte can be saved in non-volatile memory (Refer to command No13). After power up, LED controller will use the last value of STATUS byte, saved in non-volatile memory. If HOST transmit only ESC (1Bh) and command code (39h), LED controller will transmit value of current used STATUS register. When command is completed, MCA2X16 transmits to HOST new value of STATUS byte.

	HOST – LED DISPLAY	**LED DISPLAY to HOST**		
ASCII	**ESC 9 (xx)h par**	**(XXh)**		
HEX	**1Bh 39h (xx)h par**	**(XXh)**		
		MIN	**TIPICAL**	**MAX**
Tint – internal exec. period		-	1,35 mkS	-

Description of STATUS byte:		
bit7	-	Set=Literal message mode, Clear=Hexadecimal message mode
bit6	-	Not used
bit5	-	Enable - Disable Power LED
bit4	-	Not used
bit3	-	Not used
bit2	-	Increment/ Decrement Digit and Character pointer mode.
bit1	-	Select LINE2
bit0	-	Select LINE 1

No	ASC II VALUE	HEX. VALUE	COMMAND
11.	**ESC : (XX)h Sl**	**1Bh 3Ah (XX)h Sl**	**Slave Select**

Command selects Slave LED controller that will communicate with HOST

This command Select desired LED controller with HOST wants to communicate. This command is used only when MCA2X16 is connected via RS485 interface board. Otherwise
Issuing of this command never be discarded and no any message to HOST will be transmitted.

Principal of this command is to set internal Flag to take possibility of the HOST to communicate with selected Slave LED controller. When command is completed, MCA2X16 sends to HOST his slave address for notification that is ready to communicate with HOST.
Activity LED is Light up until MCA2X16 is selected.
Note: Internal flag is set only when transmitted address from HOST is equal of those that
Is selected from S1 DIP switch. For more information about DIP switch settings, refer to
Hardware Reference Manual for MCA2X16 in attendant Compact disk or download document from our web page: http://www.itsdisplays.com/products.htm

	HOST – LED DISPLAY	LED DISPLAY to HOST		
ASCII	**ESC : (xx)h SL**	**(xx) SL**		
HEX	**1Bh 3Ah (xx)h SL**	(xx) SL		
		MIN	**TIPICAL**	**MAX**
Tint – internal execution		-	2,35 mkS	-

No	ASC II VALUE	HEX. VALUE	COMMAND
12.	**ESC ;**	**1Bh 3Bh**	**Measure Light intensity**

Command measure Light intensity

Using this command user can measure Light intensity of the room. Measured values are transmitted from LED controller to HOST. It takes ability to adjust light up intensity of the segments depending of Environments brightness. This command consists of two bytes, issued from HOST. After transmitting command, LED controller sends value of Light intensity value to HOST as follows, First is the Most significant byte; second one is least significant byte. Value is decimal in range from 0000d (very light) up to 1023d (full dark). After sending measured value, command is completed. Measured value is only decimal. When photo component is disconnected, HOST will receive value (1023)d. Measurement value is depending of used photo component. As example, if is used photo component LRD 07, when photo component is exposed to direct sun light, LED controller will receive values in range between 0020d – 451d, and full dark room 1023d. Depending of measured value, user can control PWM value for optimal light up intensity of the LED Display.

	HOST – LED DISPLAY	LED DISPLAY to HOST		
ASCII	**ESC**	**(xx)h (xx)h**		
HEX	**1Bh 3Bh**	**(xx)h (xx)h**		
		MIN	**TIPICAL**	**MAX**
Tint – internal exec. period		-	270 mS	-

EXAMPLE:

1B **3B** - Transmitted from Host
02 **34** - Received from HOST.

o	ASC II VALUE	HEX. VALUE	COMMAND
13.	**ESC <**	**1Bh 3Ch**	**Save values of PWM and STATUS byte**

Command save values of PWM and STATUS byte in non volatile memory

This command writes last values of PWM an STATUS byte in non-volatile memory. After power up, values of PWM and STATUS byte are automatically restored from non-volatile memory. After completing the command, depending of selected message mode, LED controller sends to HOST (44h) in HEX mode or '**DONE**' in Literal mode.

	HOST – LED DISPLAY	**LED DISPLAY to HOST**		
ASCII	**ESC <**	**DONE**		
HEX	**1Bh 3Ch**	**44h - (D) ASC II**		
		MIN	**TIPICAL**	**MAX**
Tint – internal exec. period		-	18,4 mS	-

Note: When this command is issued, may flicker appear on LED Display for time of execution of the command.

No	ASC II VALUE	HEX. VALUE	COMMAND
14.	**ESC =**	**1Bh 3Dh**	**Restart LED controller**

Command restart LED controller

This command restarts LED controller. This command acts as power up reset. LED Buffer is clear and activity led will flash for about 200 ms when CB485 is connected to MCA2X16. Otherwise Host receive message "MCA2X16-XXX" depending from interface board. After this message is received from HOST, LED controller MCa2X16 is ready to accept new command.

	HOST – LED DISPLAY	**LED DISPLAY to HOST**		
ASCII	**ESC =**	**MCA2X16-XXX**		
HEX	**1Bh 3Dh**	**MCA2X16-XXX**		
		MIN	**TIPICAL**	**MAX**
Tint – internal exec. period		-	4,9 mkS	-

No	ASC II VALUE	HEX. VALUE	COMMAND
15.	**ESC >**	**1Bh 3Eh**	**Download character generator**

Command downloads character generator to HOST

This command sends to HOST character generator as block of file (1024 bytes). You can copy Character generator binary file character16.bin from our web: http://www.itsdisplays.com/products.htm. Character generator can be edited as you wish in suitable hexadecimal editor (supported with most EPROM/FLASH programmers) and after that can be uploaded to MCA2X16. No answer issued from LED controller to HOST.

	HOST – LED DISPLAY	**LED DISPLAY to HOST**		
ASCII	**ESC >**	**1024 bytes block**		
HEX	**1Bh 3Eh**	**1024 bytes block**		
		MIN	**TIPICAL**	**MAX**
Tint – internal exec. period 1 byte		-	2,5 mkS	-

No	ASC II VALUE	HEX. VALUE	COMMAND
16.	**ESC ?**	**1Bh 3Fh**	**Upload character generator**

Command uploads and writes character generator into LED controller's non-volatile memory

This command sends to LED controller character generator as block of file, and writes this block in non-volatile memory. When block is received, LED controller checks file for volume, clear both LINES and writes new values. If uploaded character binary file value is different from 1024 bytes, LED controller sends to HOST in Literal Mode "**INCORRECT VALUE**" or in HEX mode 55h (U). After completing of command depending of selected message mode, MCA2X16 send to HOST (44h) in HEX mode or '**DONE**' in literal mode. Characters will appear with the new shapes. Command value must be present in first two bytes of character generator file.

Another way to upload character generator, is to open DOS prompt in Windows environment. Set appropriate mode for serial channel as example :

C:\ mode com1:57600,n,8,1

After appropriate baud rate is set simply write next command in DOS prompt :

C:\ copy character16.bin com1:/Z

When message "100% copied 1 file (s) copied." is appear in DOS prompt that mean that character generator is uploaded and written in non-volatile memory. User can verify validity of Character generator, using command No 15.

	HOST – LED DISPLAY	**LED DISPLAY to HOST**		
ASCII	**ESC ?..1024 bytes block**	**DONE**		
HEX	**1Bh 3Fh .. 1024 bytes block**	**44h - (D) ASC II**		
		MIN	**TIPICAL**	**MAX**
Tint – internal exec. write period block		-	147 mS	-

No	ASC II VALUE	HEX. VALUE	COMMAND
17.	**ESC @ (xx)h**	**1Bh 40h (xx)h**	**Set digit pointer**

Command sets start position of digit pointer

This command set position of digit pointer for displaying seven segment indicators. When user sends digital values (from 00h – 11h) digit pointer is incremented - decremented depending of number of received values. User can set start position of the string issuing this command before string is sends. Position can be set between 00h – 9Fh, except defined character buffer area in selected LINE. After reset value of digit pointer depends of status of bit 2 in STATUS register if bit2=0 pointer is set to 00h, if bit2=1 pointer is set to 9Fh. When value comes to 9Fh, pointer jumps to address 00h and rises up (increment or decrement, depending of state of bit 2 in STATUS byte) with the every received value for 7 segment indicator. If character buffer for 14 , 16, 17 segment indicators is defined, when transmitted value of digit pointer overlap this area and his boundary in LED buffer, LED controller sends message "ADDRESS IS OUT OF RANGE" in literal message mode or (41)h (A) ASCII in HEX message mode. When are transmitted only ESC and command code bytes, without value of digit pointer, MCA2X16 sends to HOST last position of the digit pointer. Values for seven segment indicators received from HOST are as follows :

HEX	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10*	11*
display	0	1	2	3	4	5	6	7	8	9	a	b	C	D	E	F	.	clr

Notes*: LED controller accepts in this mode only hexadecimal values from 00h-11h.
Value of 10h, after preceding digit will set decimal pointer in this digit.
Value 11h clears selected from digit pointer position digital indicator (display indicator is blanc).
Values from 12h to 1Ah will be discarded and values from 20h to FFh will be thread as ASCII characters for dump alphanumeric displays.

Example 1 :

00 02 10 03 08

Transmitting such string to LED controller will show on display 4 digits 02.38 with decimal point in the second digit.

0	2.	3	8

Example 2 :

00 02 11 08

Will be appearing 3 digits on the seven segment display digit between 2 and 8 will be blanc.

0	2		8

Before issuing this command, please check with LINE is selected in STATUS register.
Note: If both LINES are selected, command will acts to LINE1 only.
After completing this command, LED controller sends value of new position to HOST.

	HOST – LED DISPLAY	LED DISPLAY to HOST
ASCII	**ESC @ (xx)h dp val**	**(xx)h dp val**
HEX	**1Bh 40h (xx)h dp val**	**(xx)h dp val**

Without character buffer defined	MIN	TIPICAL	MAX
Tint – internal execution write period	-	3,5 mkS	-
With character buffer defined	**MIN**	**TIPICAL**	**MAX**
Tint – internal execution write period	-	5,5 mkS	-

No	ASC II VALUE	HEX. VALUE	COMMAND
18.	**ESC A (xx)h(xx)h**	**1Bh 41h (xx)h(xx)h**	**Define character buffer**

Command defines character buffer for ASCII characters

This command defines character buffer for 14, 16, 17 segment indicators. Without defined buffer, when ASCII string or character is send to LED controller, this string will be discarded. Actually, using this command, user reserve space in LED buffer for displayed ASCII characters. Value is written Once in non-volatile memory, and is preset for the next restarts and power up's of LED controller. Character buffer consists least from two blocks that occupied 20 bytes from LED buffer area. User controls displayed characters and strings via bit 2 in STATUS register INC/DEC and character pointer position that sets position of character pointer. Character pointer can be set only in range occupied of Character buffer (note next command). In each LINE can be defined only one character buffer. Before issuing this command, refer to appropriate LINE select bits in STATUS register. If both LINES are selected, LED controller will define character buffer only in LINE1. Clearing of buffer is done via writing 00h in parametric byte. If parameter byte is equals to 00h LED controller will clear character buffer and all range of LED buffer will reserved for 7 segment indicators or other purposes. When command is completed, LED controller sends to HOST 3 hexadecimal bytes. First one is a start address, second one is the end address of the defined character buffer and third one is the current mode select for dumped alphanumeric indicators (14, 16 or 17 segment). When HOST sends 1Bh 41h without parametric bytes, LED controller transmits current values of settings. When command is completed successfully, LED controller restarts it self.

First Parameter byte :

PARAMETRIC BYTE

Upper Nible	Lower Nible
Xh	Xh
0h - Dh	0h - 8h

Upper nibble of the parametric byte points beginning of the character buffer and can vary from 0h-Dh. Value of start address is multiplied by 10 for setting real boundary. This rule is
Defined from hardware of external buffers. Table below show relationship between upper nibble of the parametric byte and real start address of character buffer.

Upper nibble value	Real Start Address in LED buffer
0h	00h
1h	0Ah
2h	14h
3h	1Eh
4h	28h
5h	32h
6h	3Ch
7h	46h
8h	50h
9h	5Ah
Ah	64h
Bh	6Eh
Ch	78h
Dh	82h
Eh	8Ch

No	ASC II VALUE	HEX. VALUE	COMMAND
19.	**ESC B (xx)h**	**1Bh 42h (xx)h**	**Set character pointer**

Command sets position of character pointer for 14, 16, 17 segment displays in Character buffers

This command set position of character pointer for displaying 14, 16, 17 segment indicators. When user sends ASCII characters and strings, character pointer is incremented or decremented depending of state of bit 2 in STATUS register of received ASCII characters. User can set start position of the string issuing this command before string is transmitted to LED controller. Position can be set in all range of defined character buffer. Each 14, 16, 17 segment indicators occupied two bytes in character buffer, but the character pointer shown number position of the indicator. After reset, value of character pointer is equals to Start Address of Character buffer or End Address of Character buffer, depending of bit 2 in STATUS register. When value of character pointer comes to the last indicator, pointer jumps to the Start or End position of the LED buffer and increment/decrement with the every received ASCII character.

Mode select is 01h – 14 segment indicators

When point is received, (2Eh) (.) ASCII, digital point is written in preceding 14 segment indicators. Character pointer is not incremented-decremented. When is sending string including more than two consequential digital points, LED controller always will display only one digital point in preceding indicator. If user wish to display more than two consequential digital points on the 14 segment display, must decide Effective address in LED buffer and use command No 4 (Writes byte in LED Buffers).

Mode select is 02h – 16 segment indicators

When point is received, (2Eh) (.) ASCII, digital point is constructed from segments D1, M and N and appears on LED Display like small triangle. Character pointer is increment - decrement. Digital point pin of 16 segment indicators are not connected.

Mode select is 03h – 17 segment indicators

When point is received, (2Eh) (.) ASCII, digital point is written in preceding 16 segment indicators. Character pointer is not increment-decrement. When is sending string including more than two consequential digital points, LED Controller always will display only one digital point in preceding indicator. If user wish to display more than two consequential digital points on the 16 segment display, must decide Effective address in LED buffer and use command No 4 (Writes byte in LED Buffers).

If value of character pointer exceed character buffer boundary, LED Controller sends "WRONG VALUE" in literal message mode or (57h) (W) ASCII in HEX message mode and do not change current value of character pointer. Before issuing this command, refer to appropriate LINE select bits in STATUS register.

Note: If both LINES are selected, LED Controller will set character pointer only in LINE1.

When are transmitted only ESC and command code bytes, without new value of character pointer, LED Controller sends to HOST current position of the character pointer.

	HOST – LED DISPLAY	**LED DISPLAY to HOST**		
ASCII	**ESC B (xx)h**	**(xx)h value of character pointer**		
HEX	**1Bh 42h (xx)h**	**(xx)h value of character pointer**		
		MIN	**TIPICAL**	**MAX**
Tint – internal exec. Set pointer		-	5,4 mkS	-

COMMAND SET SUMMARY for MCA2X16			
No	**ASC II VALUE**	**HEX. VALUE**	**COMMAND DESCRIPTION**
1.	**ESC 0**	**1Bh 30h**	**Test communication**
2.	**ESC 1**	**1Bh 31h**	**Sets value of PWM**
3.	**ESC 2**	**1Bh 32h**	**Clear LED Buffers**
4.	**ESC 3**	**1Bh 33h**	**Writes byte in LED Buffers**
5.	**ESC 4**	**1Bh 34h**	**Writes block of bytes in LED Buffers**
6.	**ESC 5**	**1Bh 35h**	**Read byte from LED Buffers**
7.	**ESC 6**	**1Bh 36h**	**Read block of bytes from LED Buffers**
8.	**ESC 7**	**1Bh 37h**	**Not Used**
9.	**ESC 8**	**1Bh 38h**	**Not Used**
10.	**ESC 9**	**1Bh 39h**	**Writes (change/read) STATUS byte**
11.	**ESC :**	**1Bh 3Ah**	**Slave Select for RS485 interface**
12.	**ESC ;**	**1Bh 3Bh**	**Measure Light intensity**
13.	**ESC <**	**1Bh 3Ch**	**Save values of PWM and STATUS byte**
14.	**ESC =**	**1Bh 3Dh**	**Restart LED controller**
15.	**ESC >**	**1Bh 3Eh**	**Download character generator**
16.	**ESC ?**	**1Bh 3Fh**	**Upload character generator**
17.	**ESC @**	**1Bh 40h**	**Set digit pointer**
18.	**ESC A**	**1Bh 41h**	**Define character buffer**
19.	**ESC B**	**1Bh 42h**	**Set character pointer**
For more applications, descriptions and manuals refer to: http://www.itsdisplays.com/			

If you have any questions or find some mistakes in this document, do not hesitate to contact us on the next E-mails: its@itsdisplays.com or support@itsdisplays.com , and we try to answer you as soon as possible.