Rev.3.8

EXTERNAL BLOCK DESCRIPTION MANUAL FOR LED DISPLAY CONTROLLER MCA2X16

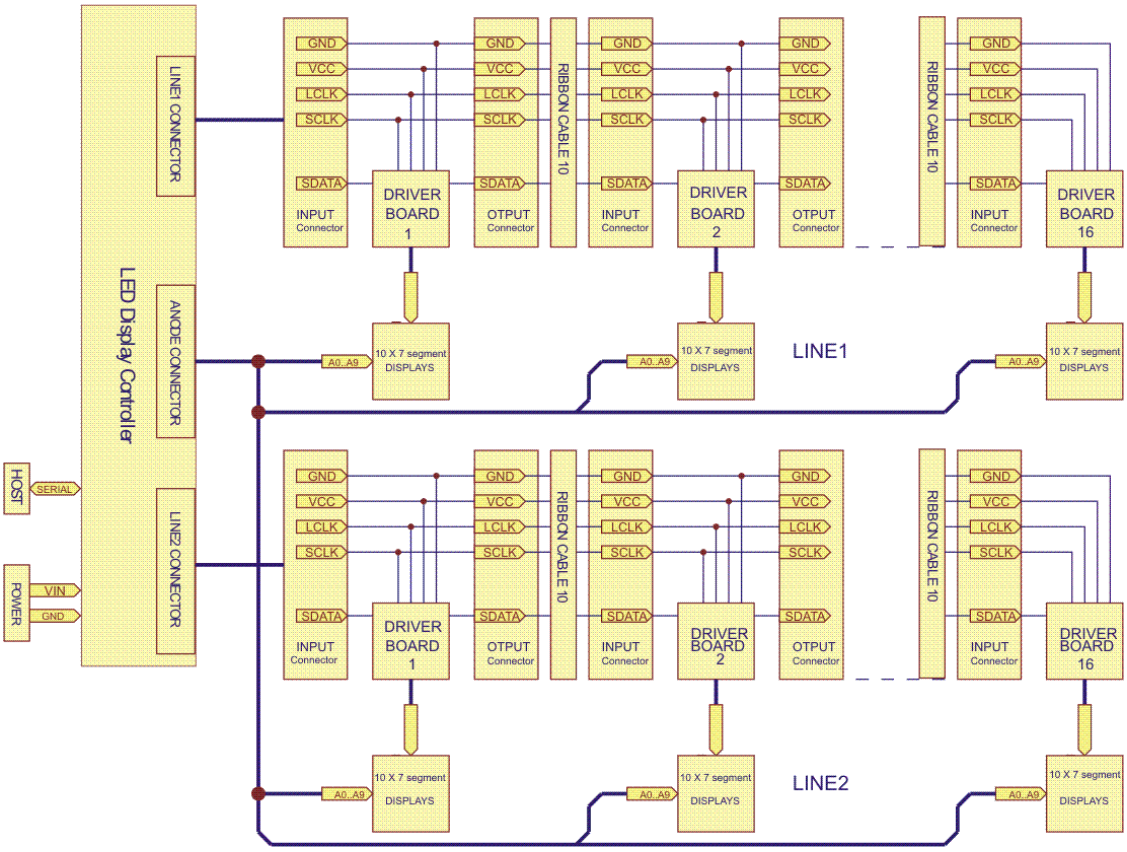


Figure 1

I. LINES, Blocks and LED Buffers

Controller can drive two independent LINES. LINES consist from 1 up to 16 external serial shift registers called blocks. Concept of block is defined of different simple external components that are selected mainly from desired load current to drive indicators. This technique takes ability of connection of different types of indicators in each LINE. Block schematic of two lines are shown in figure 1. Driving of all kind of LED displays is performing via writing of bytes or block of bytes in LED buffer. All data that receive controller, are saved first in internal receive buffer and after that are written in Selected locations in LED Buffer. This "HIDEN WRITE and REFRESH "principal of operation void flicker effect of the LED display and loss of refresh cycles. LED Buffers are two RAM memory blocks 160 bytes each that used for dump data to LED display. Relationship between LINES and LED buffers is very simple. When is written some byte in LED buffer in appropriate address, this data will be appears on the LED Display in the next refresh cycle. As shown on figure 4 seven segment indicator Q1 that are connected to anode A0 refers in LED Buffer to address 00h, A1 to address 01h and etc. First indicator in the next block will start from address 0Ah and etc. The last seven segment indicator Q10 in the last sixteen block will be refers
To address 9Fh in LED Buffer. By this way each line can drive up to 160 seven segment indicators. Sample from LED buffer.

Bl1	Bl2	Bl3	Bl4	Bl5	Bl6	Bl7	Bl8	Bl9	Bl10	Bl11	Bl12	Bl13	Bl14	Bl15	Bl16

00h-First address Addresses in LED Buffer Last Address-9Fh

II. Defining Character Buffer

User can read and write any data in any address in LED buffer using command No 4, 5, 6, 7 in selected LINE, but this takes lot of efforts to design each shape of sign and deciding appropriate position in LED buffer. Using internal uploadible / downloadable character generator takes ability to construct any kind of large LED displays without any additional software, only using digit and character pointers for each line to control seven segment and 14/16 segment indicators connected by the specific user needs in each LINE. Sense of Defining of character buffer in LED buffer is to use internal character generators for alphanumeric 14/16 segment indicators. Character buffer occupied least two consecutive blocks (Refer to command No 18 in command reference manual). As example if we needs from 10-14/16 segment displays and 90 seven segment indicators.
User can define buffer where he wish. In this example character buffer is defining using Blocks 2 and 3. Boundary address is a start from 0Ah up to 13h, all other blocks are reserved now for seven segment indicators.

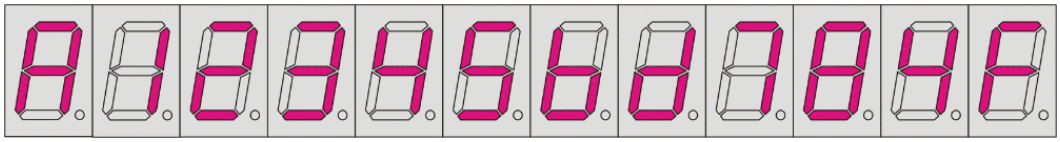
Bl1	Bl2	Bl3	Bl4	Bl5	Bl6	Bl7	Bl8	Bl9	Bl10	Bl11	Bl12	Bl13	Bl14	Bl15	Bl16

00h-First address Addresses in LED Buffer Last Address-9Fh

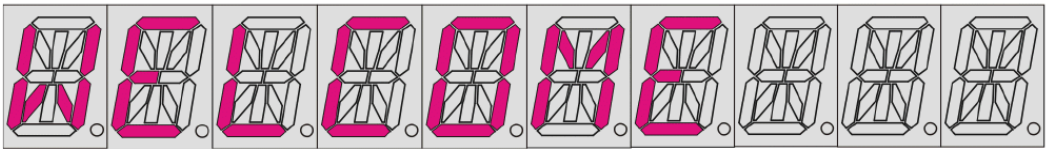
Now LED buffer are separated by two independent LED buffers. One for seven segment indicators and other for 14/16 segment indicators.
Depending of state of the bit – 2 in STATUS register (refer to command No 10) two pointers (digit and character) are incremented/decremented automatically after receiving of hexadecimal digit or ASCII character. If we send 12 digits as example below :

0A 01 02 03 04 05 0B 0D 07 08 04 0F

This digits will be appear in bl1 (10 digits) and Bl4 (2 digits) and digit pointer will show the last dumped seven segment indicator skipping character buffer area. If user read value of digit pointer he will receive value of 0Bh, but the actual address in LED buffer is a 15h. On the seven segment LED display will appears next sequence :



By the same way if a send a ASCII string WELCOME to the controller, It will appear on 14/16 segment LED Display as :



And character pointer will show value 06h. Each 14/16 segment indicator occupied two address bytes from Character buffer. As example first indicator occupied addresses 00h and 0Ah. In this case (10 character indicators) character pointer will rise up to 09h and the next value will be 00h.

III. External Components

External buffer block have two main purposes. First to buffer cathodes of displays and load shift registers. Schematic of each block is very simple. Each driver block consist from shift register 74XX595, Darlington buffer ULN 2803, 8 Load resistors and 2 connectors. Blocks in each LINE are connected serially via 10-pin ribbon cable (note **Figure 1**). All necessary signals for control of blocks are produced from controller. Cables between neigh borough blocks must be not longer than 1 m. To void a different light intensity in different Blocks, Ground must be assured to provide enough current for Darlington transistors. Each driver block occupied 10 bytes in controllers LED Buffer. Via writing of byte (bytes) in appropriate address in LED Buffer, controller drives defined segments of selected indicator. After power up two LED LINES is filled with 00h (display is blanc). Each external block can drive up to 10 seven-segment displays. All 16 external blocks can drive up to 160 seven-segment displays for each Line. All tree control signals are connected in parallel except, Dout signal that must be connected serially between each block. Refer to **Figure 1**. As noted from **Figure 4**, Q1 refers to address 00h in LED buffer, Q1 to 01h and Q10 to 09h. In the next block indicators will refer to LED Buffer as follows Q1 to 0Ah, Q2 to 0Bh and etc.

JP5 on LED Controller is used for different schematics applicable to drive LED displays. In this schematic **JP5** on LED Controller must be in open position, because ULN2803 invert data comes from shift register 74XX595. This schematic provide maximal Load current of 500 mA for each segment and is suitable for building of very large LED displays, where segments are constructed from multiple LED's.

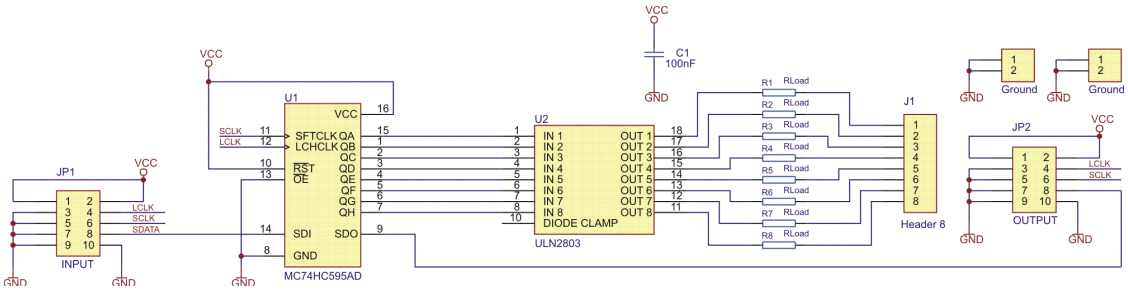


Figure 2

Another and probably better version of the external block is shown on **Figure 3.** It uses TPIC6B595DW power logic shift register. This schematic is very similar as drawn in **Figure1**, but simpler and takes ability of brilliant trace PCB indicator boards including large amount of small size LED indicators. Also is important that this schematic takes possibility to use maximal clearance factor between tracks and pads in PCB route process. JP5 on LED Controller must be in short position to SDATA coming from Controller be a positive. This schematic provides maximal Load current of 150 mA for each segment.

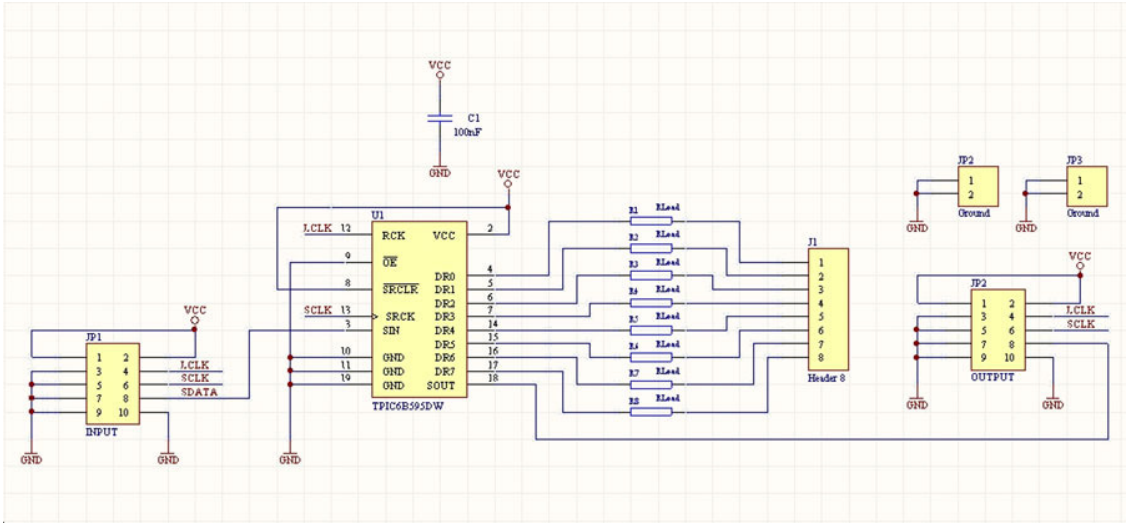


Figure 3

IV. Load Resistors

Load resistors – Load resistors limits LED's current. It varies depending of selected LED indicator and input voltage. Segments in some large Indicators are constructed from serially connected Diodes. To be Light up intensity equal for all segments, can use the formula noted below:

Vin – Input Supply Voltage
Vf - Forward Voltage each segment
Ifc - Forward current for each segment
Nled- Serially connected LED's in each segment
Rl - Load Resistor Value
Rp - Required resistor Power.
Vsat – 0.85V-1.2V Saturation voltage from ULN2003 Specifications

Resistance formula for schematic **Figure 2**[1]:

$$Rl = \frac{Vin - (Nled \cdot Vf) - Vsat}{Ifc}$$

Resistance formula for schematic **Figure 3**[2]:

$$Rl = \frac{Vin - (Nled \cdot Vf)}{Ifc}$$

Required power for load resistors formula:

$$Rp = Ifc^2 \cdot Rl$$

[1] Note: This formula is available for schematic in **Figure 2**.

[2] Note: This formula is available for schematic in **Figure 3**.

Example:

On **Figure 4** is shown schematic of one driver block. In this example we use 7 segment displays SA23-11SRWA manufactured from Kingbright company. From data sheet, we see that each segment from **a** to **g** are constructed from 4 serially connected LED's, but **dp** is constructed from two LED's. DC Forward current for each segment - **Ifc** – 30 мA, Forward voltage (high efficiently red) **Vf** – 2.0V. Input voltage **VIN** to controller is e 12V. Then for segments from **a** to **g** resistance and power is:

$$\frac{12 - (4 \cdot 2{,}0) - 1{,}2}{0{,}03} = 93.66 \text{ ohms}$$

Standard value is 100 ohms.

Required resistor power is: $0{,}03^2 \cdot 100 = 0{,}09$ W

We can use resistors with power of 100 mW.

For dp value is:

$$\frac{12 - (2 \cdot 2{,}0) - 1{,}2}{0{,}03} = 226{,}66 \text{ ohms}$$

Standard value is 220 ohms.

Required resistor power is: $0{,}03^2 \cdot 220 = 0{,}198$ W

We can use resistors 250 mW.

All decisions are made on base of PWM = 99 % (maximal load). Required current for segments from **a** to **g** is:

$$\frac{12 - (4 \cdot 2{,}0) - 1{,}2}{100} = 28 \text{ mA}$$

For decimal point

$$\frac{12 - (2 \cdot 2{,}0) - 1{,}2}{220} = 30{,}09 \text{ mA}$$

Required supply current if all segments are light up for 10 indicators is:

$$(7 \times 28) + 30{,}09 = 226{,}09 \text{ mA}$$

If we use one full line with 160 seven segment displays SA23-11, required current of power supply must be:

$$16 \cdot 226{,}09 = 3617{,}44 \text{ mA}$$

We must provide Power Supply 12 V, least 4 A current, enough to drive all 160 indicators.

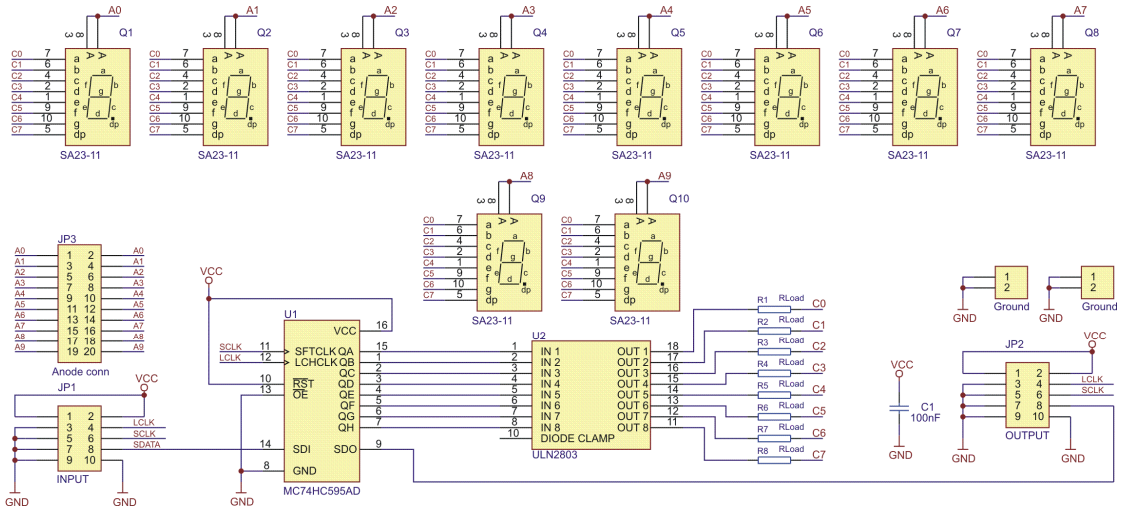


Figure 4

V. Driver boards

Driver boards takes ability to produce end product for very short time, improves simplicity of PCB boards routing and save space and flexibility of the end product. Often when indicator displays are very small, is a difficult to route all components in one PCB board. Using of separate external driver board is good practice. Driver boards are two types.

DRA1 PCB board from schematic shown on **Figure 2**. In one line can be connected 16 **DRA1** boards as shown in **Figure 1**. **Figure 5** shown topside of board.

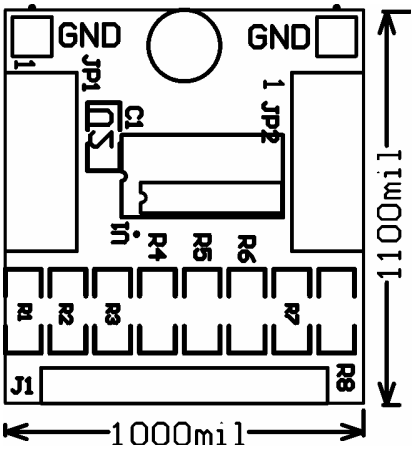


Figure 5 Top view

J1 – Segment buss connector pin 1 – **D0**…pin 8 – **D7**

pin1	pin2	pin3	pin4	pin5	pin6	pin7	pin8
DA0	**DA1**	**DA2**	**DA3**	**DA4**	**DA5**	**DA6**	**DA7**

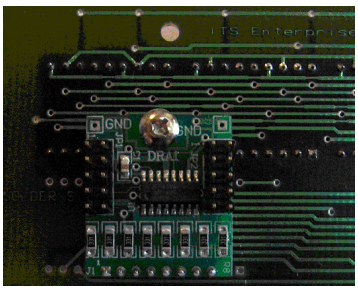
JP1 – Input connector

pin1	pin3	pin5	pin7	pin9
+5V	**GND**	**GND**	**GND**	**GND**
pin2	pin4	pin6	pin8	pin10
+5V	**LCLK**	**SCLK**	**SDIN**	**GND**

Note: SDIN input comes from controllers LINE connector, or from output connector from previous driver board.

JP2 – Output connector
Note: SDOUT output goes to the **SDIN** input of the next driver board. In the last 16th board this signal is not connected.

pin1	pin3	pin5	pin7	pin9
+5V	**GND**	**GND**	**GND**	**GND**
pin2	pin4	pin6	pin8	pin10
+5V	**LCLK**	**SCLK**	**SDOUT**	**GND**



This picture shown rear side of the indicator board. Indicator board contains only LED indicators, cathode and Anode connectors.

DRA2 is the same as **DRA1**, but can drive directly 20 seven - segment indicators. In one line can be connected up to 8 **DRA2** boards. **Figure 6** shows **DRA2** schematic. Connectors **JP1** and **JP2** are the same as this of **DRA1**.

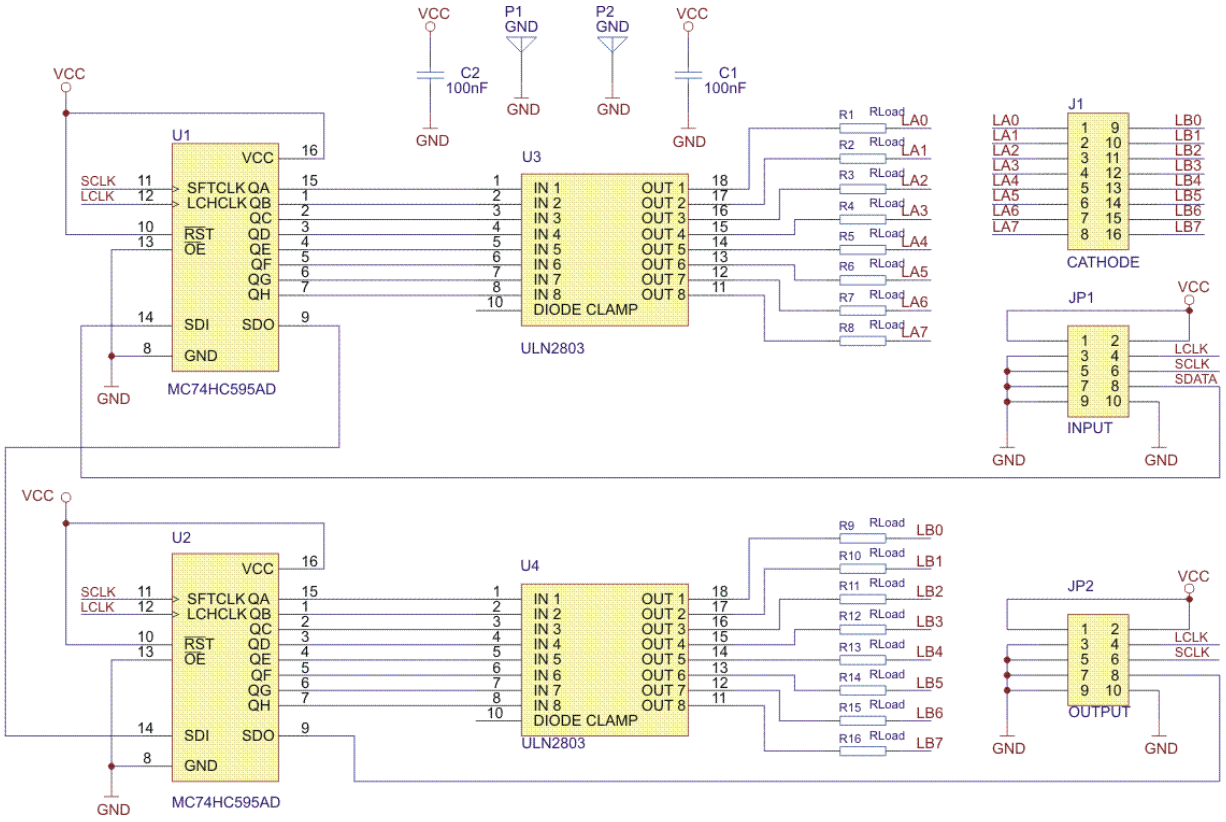

VCC
P1
GND
P2
GND
VCC
C2
100nF
C1
100nF
GND
GND
GND
U1
U3
U2
U4
SCLK
LCLK
SFTCLK
LCHCLK
RST
OE
SDI
SDO
GND
MC74HC595AD
IN 1
IN 2
IN 3
IN 4
IN 5
IN 6
IN 7
IN 8
DIODE CLAMP
OUT 1
OUT 2
OUT 3
OUT 4
OUT 5
OUT 6
OUT 7
OUT 8
ULN2803
RLoad
LA0
LA1
LA2
LA3
LA4
LA5
LA6
LA7
LB0
LB1
LB2
LB3
LB4
LB5
LB6
LB7
J1
CATHODE
JP1
INPUT
JP2
OUTPUT
LCLK
SCLK
SDATA

Figure 6

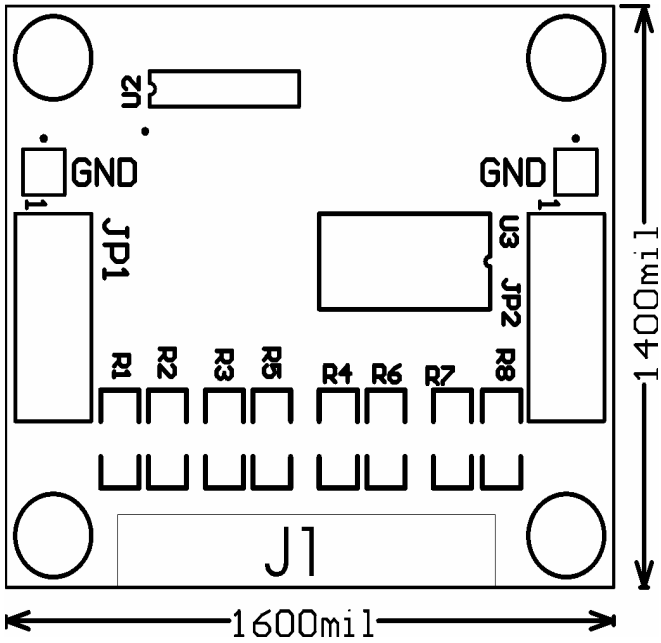


Figure 7 shows PCB board of DRA2

pin1	pin3	pin5	pin7	pin9	pin11	pin13	pin15
DA0	**DA1**	**DA2**	**DA3**	**DA4**	**DA5**	**DA6**	**DA7**
Pin2	pin4	pin6	pin8	pin10	pin12	pin14	pin16
DB0	**DB1**	**DB2**	**DB3**	**DB4**	**DB5**	**DB6**	**DB7**

J1 – Segment buss conn.

VI. Examples using driver boards

In this chapter we will be discussed variety of applications, using driver boards with different types of LED indicators.

Example 1: Seven segment indicators

In **Figure 8** is drawn typical application of driving of 10 seven-segment indicators. Common anodes of indicators are connected between controller and indicator's PCB board via 20-pin ribbon cable. All indicator segments are connected in parallel.

In LED buffer, Indicator **DS1** occupied Address 00h, next one **DS2** 01h and etc. If we want to light up digits 0,1,2,3,4,5,6,7,8,9, dp, we have tree ways to do that. First method is to use command Write byte in LED buffer ((1Bh 33h (ad)h (db)h)). But we can light up indicators, using this command 10 times for all values. This method is suitable when values are changed maze or in single positions. Second method is To use command Write block in LED buffer ((1Bh 34h (ad)h (db)h….(db)h)). Block is the string of all values that first byte refers to **DS1** and last 10nth byte refers to **DS10**. Third and simples method is to send Hexadecimal values directly to controller. If required, first must be set digit pointer and select LINE to desired position.

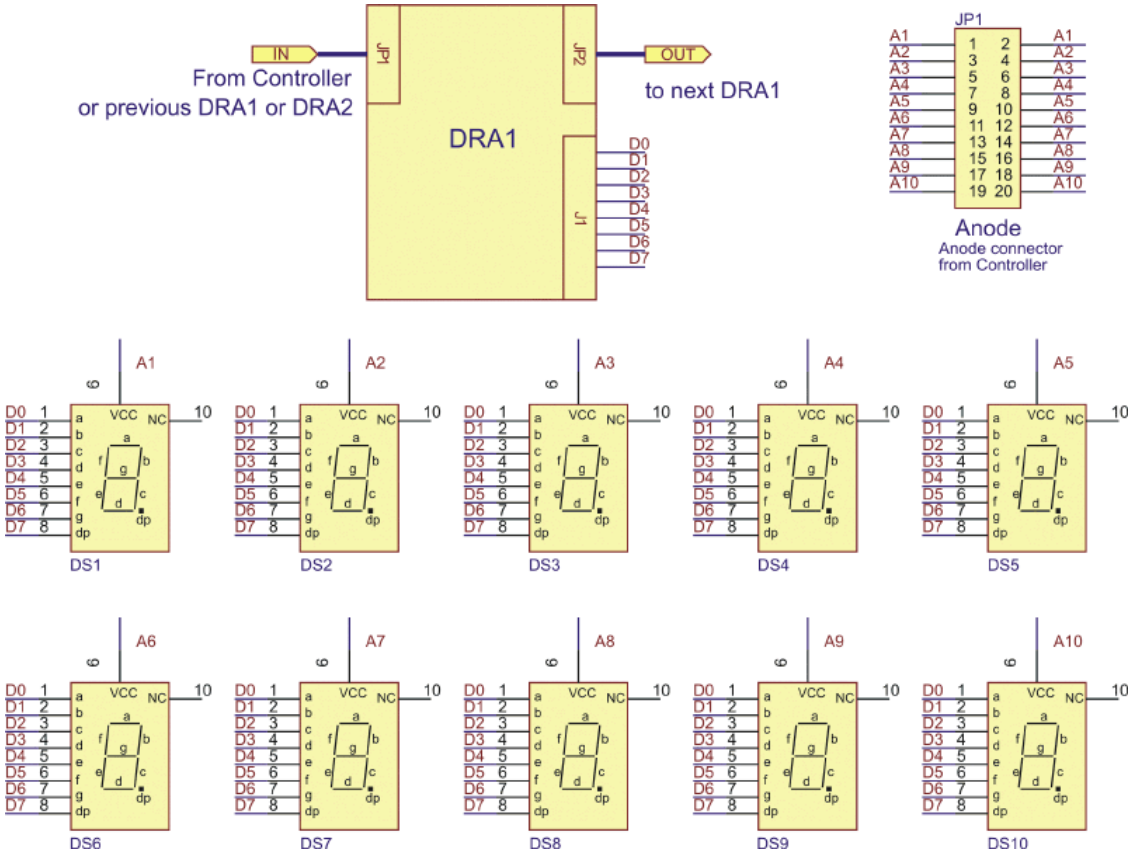


Figure 8

Example 2: Fourteen segment indicators

Figure 9 shows driving of 10 14-segment indicators. The 10 indicators occupied 20 bytes from LED buffer. Each indicator is located in two addresses in LED buffer (as example **DS1** occupied address 00h and 0A in LED buffer). By the same way user can be drive bi-color and tri-color indicators and multi color single diodes, but more convenient way is to place blocks on the same position in both LINES for bicolor indicators. By this manner user needs only to select LINE and send string to define color.

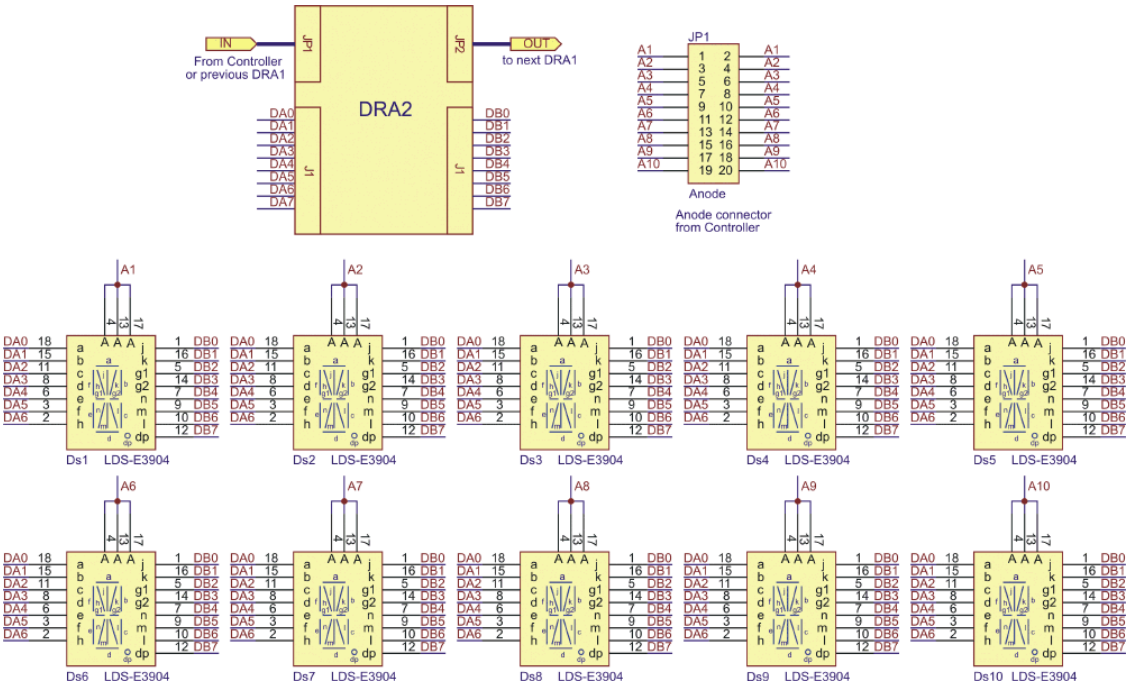


Figure 9

Example 3: Led bars or single LED's

Figure 10 points connection of 20 led bars, consist from 4 LED's each. Each two indicators occupied one address location in LED buffer. By the same way is possible connection of single diodes. Each LINE can drive 1280 single LED's. Each controller can drive total up to 2560 single LED's.

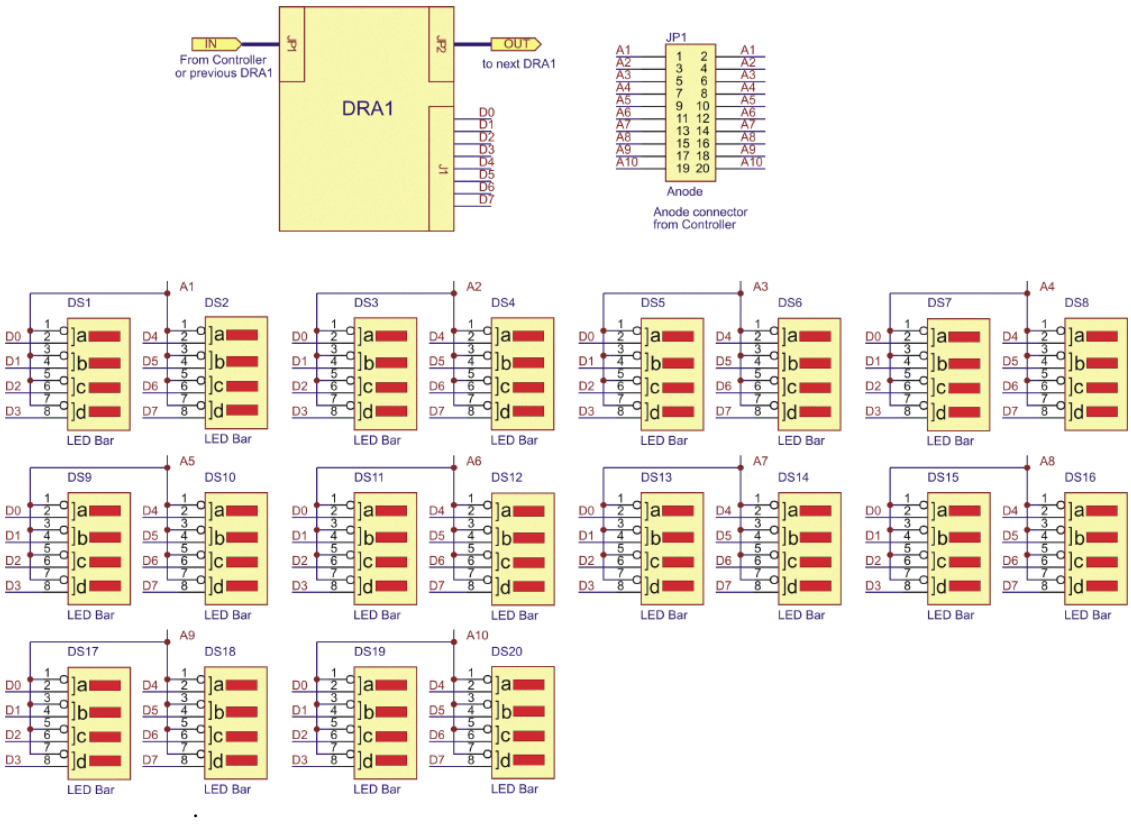


Figure 10

Example 4: Dot matrix displays

Controller can drive up to 64 pieces 5X7 dot matrix displays. In combination with seven segment indicators, is useful for time schedule boards, score tables and etc. Controller **DCA2X16** do not include internal character generator for dot matrix displays. User must send block of bytes to construct characters. **Figure 11** shows connection of the two dot matrix displays.

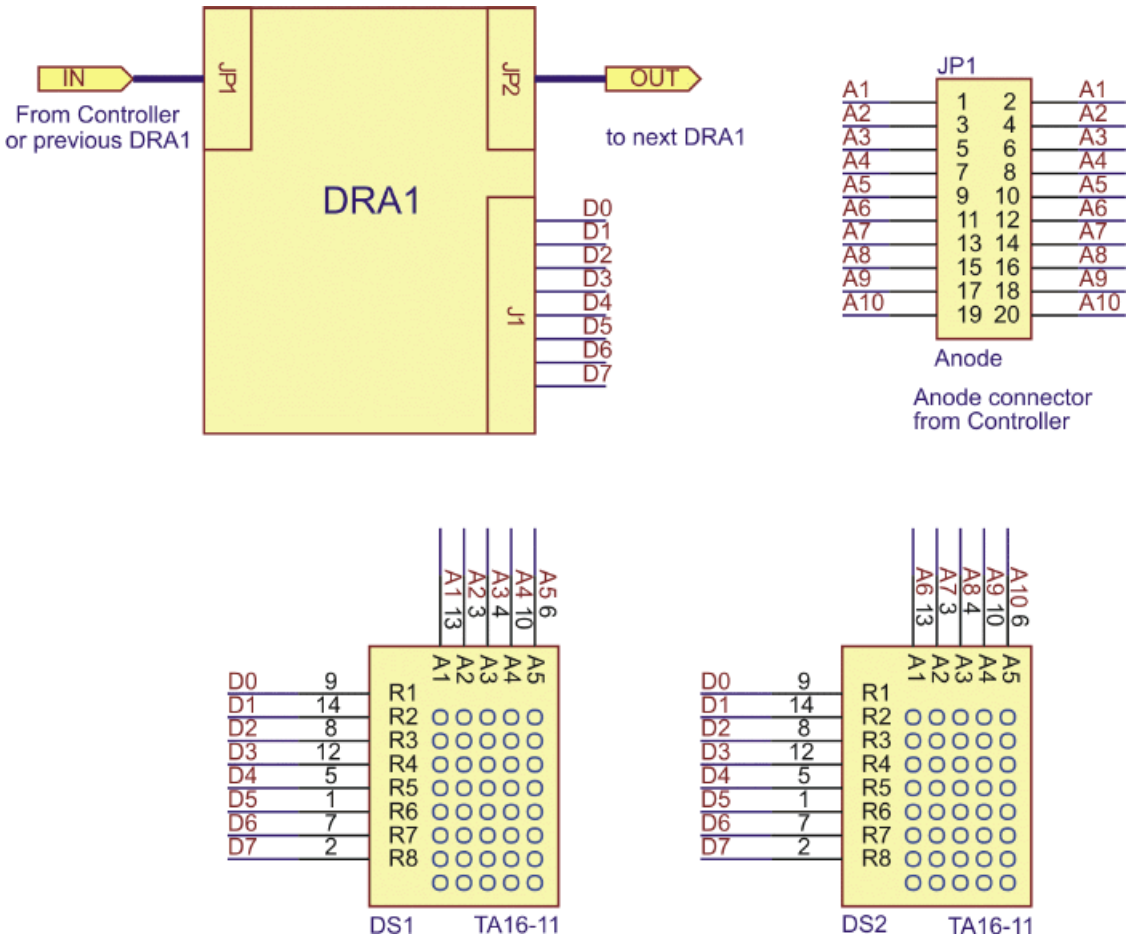


Figure11

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. Refer to **http://www.itsdisplays.com/** site to download Schematics and PCB source files.