Introduction to Driving LED Matrices

Application Note 1216

Introduction

LEDs are current driven devices. It is relatively simple to drive several LEDs individually. However, as the number of LEDs increases, the amount of resources needed to operate these LEDs will grow to an unmanageable level. As such, LEDs are often arranged in matrices in order to make efficient use of resources.

In a matrix format, LEDs are arranged in rows and columns. This arrangement will be discussed in more detail later. What should be noted here is that the matrix arrangement demands that LEDs be driven in multiplex. The multiplex sequence inevitably requires more complex processing but is more efficient compared to individually driving each LED.

This application note will also describe how the brightness of each individual LED can be controlled in multiplex mode. It involves dividing the LED driving sequence into three levels in the time domain. The last section will introduce several ICs that are widely used in driving LEDs.

This note is intended to support the design of messaging and video systems using LED tiles. However, the concepts and techniques introduced here apply to any LED matrix including arrays formed using discrete LEDs.

This application note is especially relevant to these products and applications:

- Single Color and Bi-Color Tiles
- 8x8 Rich color tiles (HDSP-R881/R883)
- LED arrays (composed of LED lamps, chip LEDs, etc.)
- LED video screens
- Moving message panels

Basic Structure of an LED Matrix

We will confine our discussion, initially, to the 4x4 matrices as shown below in Figure 1. The underlying principle here is that each LED can be addressed by specifying its location in terms of rows and columns. For example, the top-left LED is addressed as (A,1) i.e. row A, column 1. This method of addressing also indicates the flow of electrical current. In order to turn LED (A,1) on, current is caused to flow from A to 1. If switches are attached to each port A to D and 1 to 4, then, to turn the top-left LED on, switches A and 1 are made to conduct. The other LEDs will not have any current flowing because either their row or column switch is non-conducting.

Figure 1 shows two different configurations. The difference is in the method that is used to drive the LEDs. With the common-row anode configuration, current sinks are attached to ports 1 to 4. With the common-row cathode, current sources are attached to ports 1 to 4.

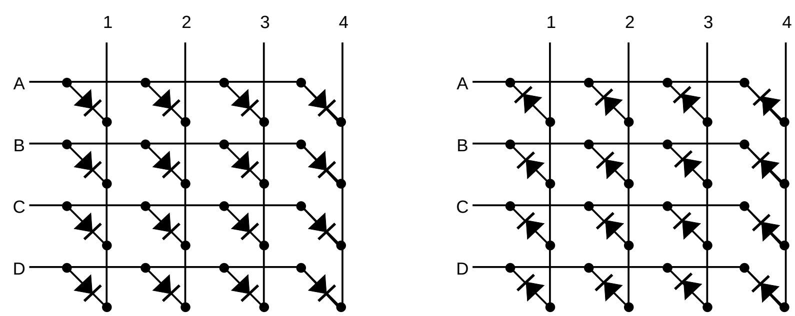


Figure 1. Common-row anode (left) and common-row cathode (right) matrix arrangements.

Multiplexing an LED Matrix

Multiplexing is the technique employed to operate LED matrices. By multiplexing, only one row of the LED matrix is activated at any one time. This approach is required because one end of the LED (either the anode or the cathode) is tied to a single row. From Figure 2, we can see that if current is applied to both rows A and B at the same time, it becomes impossible to address an individual LED within those two rows.

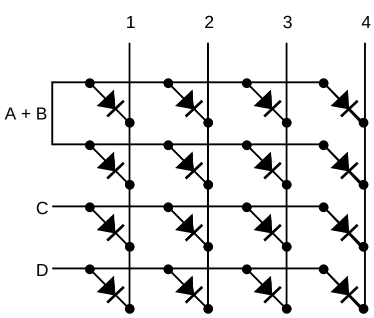


Figure 2. If we energize both A and B at the same time, it becomes impossible to address individual LEDs within those two rows. For example, if line 1 is made to conduct when (A+B) is conducting, two LEDs will light up simultaneously. Note: this is not a recommended method of operation as the LEDs are driven in parallel.

Parallel drive of LEDs is discouraged because of "current-hogging". This phenomenon occurs if the dynamic resistance of the LEDs in parallel differs by a large amount (see Application Brief D-007).

We will use the common-row anode configuration to illustrate the concepts of multiplexing.

The staircase sequence (A to D) shows that time division multiplex is employed here. Only one row is energized at any one time. During the period in which a given row is energized, the desired LEDs are lit by energizing the appropriate columns. Sometimes this process is known as scanning.

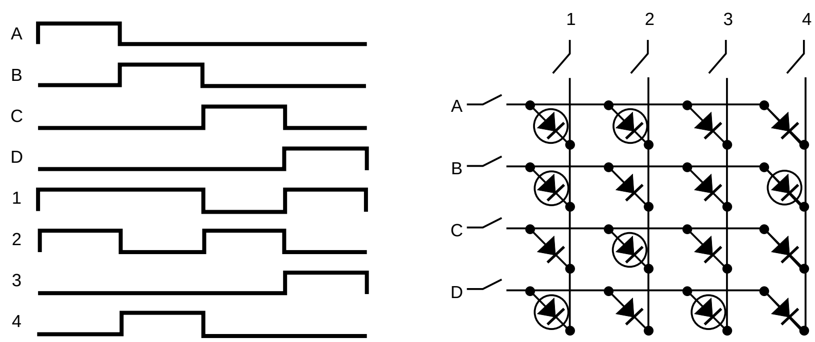


Figure 3. Multiplexing an LED matrix. Current flows when the switches are pressed. The figure on the left is a time chart showing when and which switches are pressed. The circles in the figure on the right indicate which LEDs are lit when the sequence is deployed.

Basic Structure of a Driving System

The figure only shows a section of the matrix. The driving scheme shown in Figure 4 can be extended to very large arrays of LEDs. The maximum size depends on the maximum rate at which the electronics can distribute and process data. For a common-row cathode configuration, the driving system will need constant-current sources and sink drivers instead.

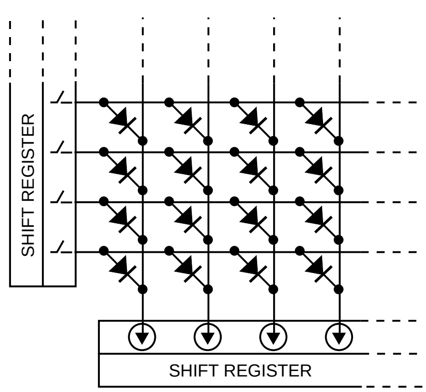


Figure 4. Implementation of a driving system. Electronic switches are used at the high-side while current sinks are used at the low-side. Shift registers are used to accept the switching sequence in digital form.

Brightness Control Via Pulse Width Modulation (PWM)

We know that the light output of an LED is dependent on the current flowing through it. However, that is not a recommended method of controlling brightness because we will need a very precise current source/sink. The preferred technique for brightness control is through

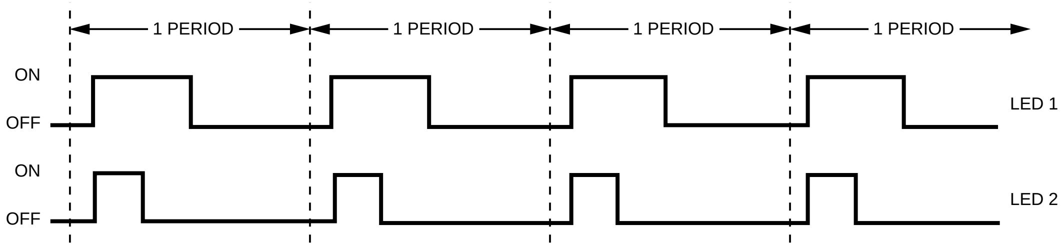


Figure 5. Switching sequence of two LEDs. LED1 will appear brighter than LED2 because it is turned on for a longer time within a period.

pulse width modulation (PWM). This concept is illustrated in Figure 5.

However, the driving system shown in Figure 4 will activate an entire row at the same time. How do we control the brightness of each individual LED? The answer is to divide each scanning period into time slots. Thus, we now have a time domain hierarchy.

Brightness Control of an Individual LED

The PWM technique described here can be extended beyond the 4 grey scale system; the narrower the time slot, the finer the brightness control. It is limited by the switching time of the driving system, which in turn determines the minimum length of a time slot. LED switching time is not an issue since it is very short (several tens of ns).

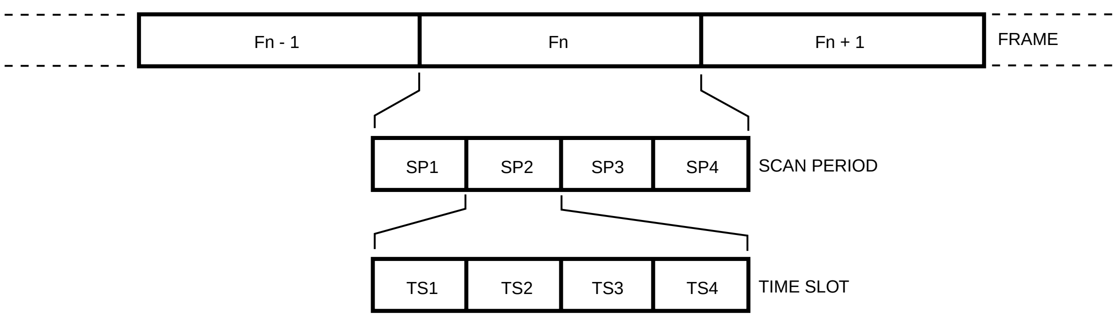


Figure 6. Hierarchy for a 4 grey scale, 1/4 duty factor system. Frame refers to a complete image on a 4x4 LED display. Frame rate is the number of frames per second (see below).

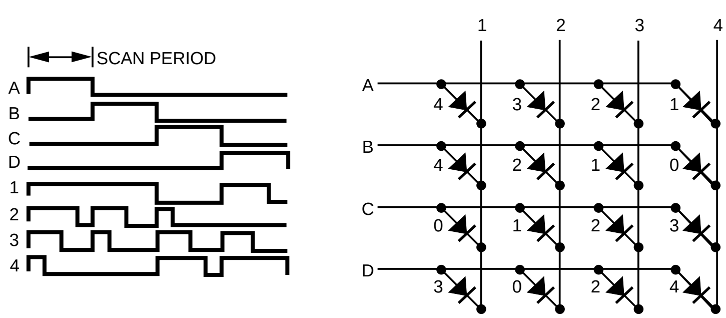


Figure 7. Individual LED brightness control technique in a multiplex scheme. The timing diagram on the left shows 4 scan periods (A to D) and 4 time slots within each scan period. Each scan period corresponds to one row of LEDs. The figure on the left shows the relative brightness of each LED. Brightness decreases from 4 to 0.

Frames and Persistence of Vision

Frames here are defined as the final image on the display that is to be presented to the observer. Frames can be simple characters or pictures. Video works by presenting a set of frames so quickly that the observer does not perceive any discontinuity. The rate at which the frames are refreshed is termed the refresh frequency. If the frequency is above a certain threshold frequency, the observer will not notice any flickering. For LED displays, a refresh rate of above 60 Hz is recommended.

Persistence of vision is the human visual phenomenon that allows video images to be viewed without flicker. When the human visual system is presented with an image, that image continues to be perceived even though it is no longer in the visual field of the observer, albeit for a short time. This phenomenon thus enables flicker-free video.

Towards a Practical LED Display System

While the previous section discussed basic concepts, this section focuses on the construction of a practical display system. We will begin by defining some of the terms used in this section. This will be followed by a description of a typical display system. Finally, the details of each element of that system will be discussed. It would be helpful to download the data sheets of the drivers mentioned here. The Appendices point to the on-line documentation store.

Terminology

1. **Common line** - see Figure 8.
2. **Access line** - see Figure 8.
3. **Pixel/dot** - pixel and dot refer to the same object (see Figure 9).
4. **Refresh rate** - the frequency of the images being displayed.
5. **Brightness control** - control of the overall brightness of the display.
6. **Gray scale** - control over the brightness of each LED in order to generate multiple color combinations. e.g. 8-bit gray scale per LED (red, green and blue) means that the brightness of each LED can be controlled to 256 "shades". By mixing the three colors contained in a pixel, 256x256x256 (16.7 million) or 24-bit color depth is achieved.
7. **PWM** - Pulse Width Modulation, the common method used to control light output from an LED (see Figure 10).
8. **Scan period** - the period of time a common-line is activated (see Figure 7).
9. **Peak forward current** - the maximum forward current that the LED is subjected to (see Figure 10).
10. **Average forward current** - the time averaged current the LED experiences (see Figure 10).

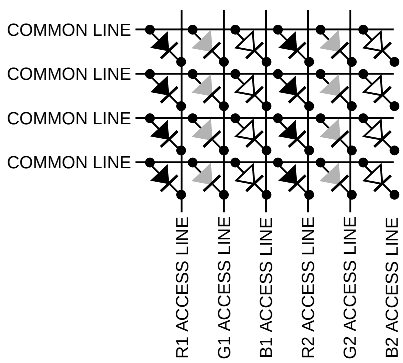


Figure 8. LED matrix indicating common and access lines. R1, G1 and B1 are red, green and blue LEDs respectively and are grouped as a single pixel. R2, G2 and B2 form another pixel.

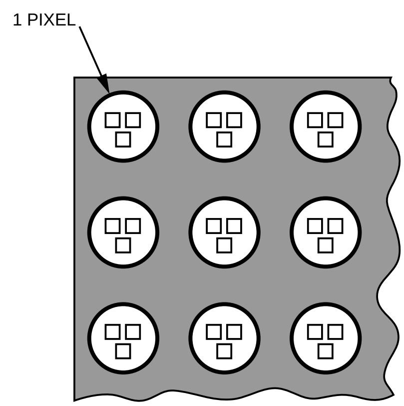


Figure 9. Definition of a pixel. One pixel can contain more than one LED.

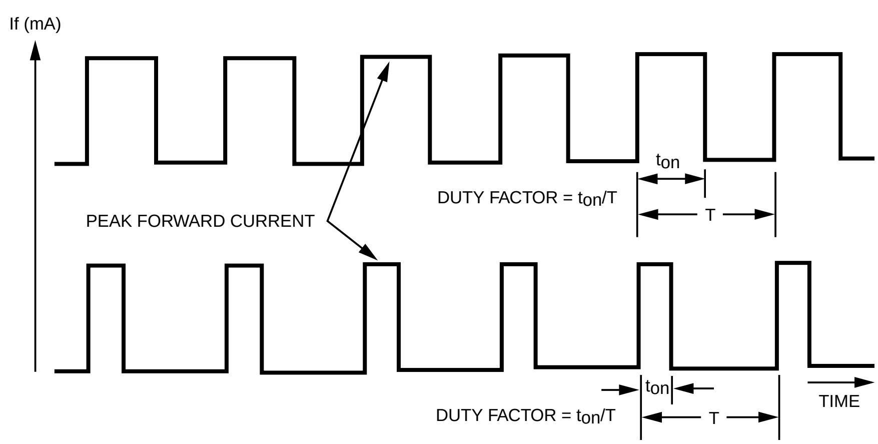


Figure 10. Definition of PWM and peak forward current. Average current, Iavg, is Ipeak x D.F. where Ipeak is peak forward current and D.F. is duty factor. Generally, higher Iavg results in higher brightness. Hence, the top pulse train will produce a brighter LED than the bottom one.

Display System Structure

This section concentrates on video displays. The drivers used here are constant current latches and intelligent drivers.

Each level contains logic to handle data distribution and generation of control signals. The signal source will provide the image data. It can take any form, eg. VCR, PC graphics card or a DVD player.

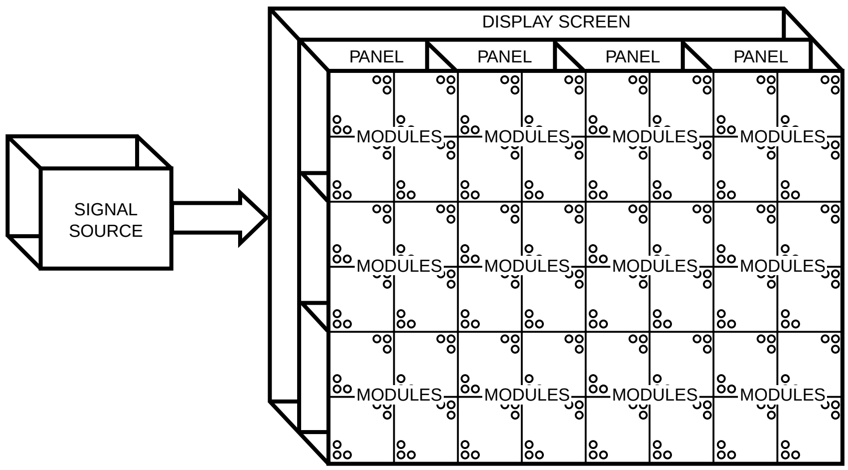


Figure 11. Typical structure of an LED standalone display panel. At the top-level is the display screen controller followed by the panel controller which governs the operation of the LED module.

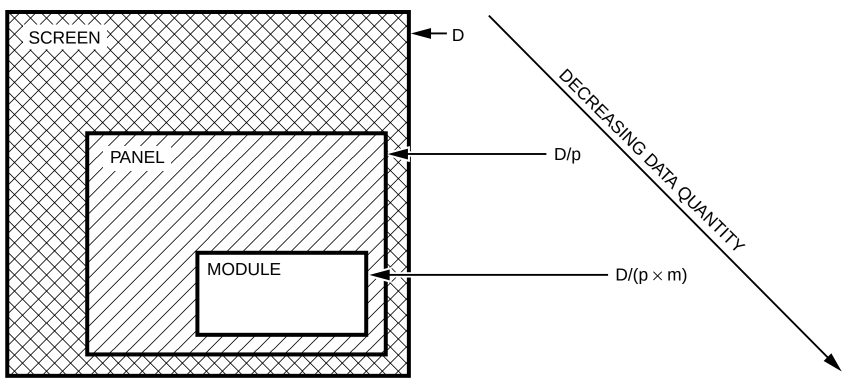


Figure 12. Hierarchy of a display screen - a screen consists of several panels while a panel consists of several modules. D is the amount of data per screen. e.g. in a 320x240 pixel screen with 24-bit color depth, D = 320x240x24 = 1.84 million bits. p = number of panels per screen and m = number of modules per panel.

Splitting the screen into several components allows for easier management of data. A typical configuration is a 32x16 LED module. Each panel might consist of 2x3 modules resulting in a 64x48 LED panel. Lastly, the screen can contain 5x5 panels producing a 320x240 LED screen.

The LED Module

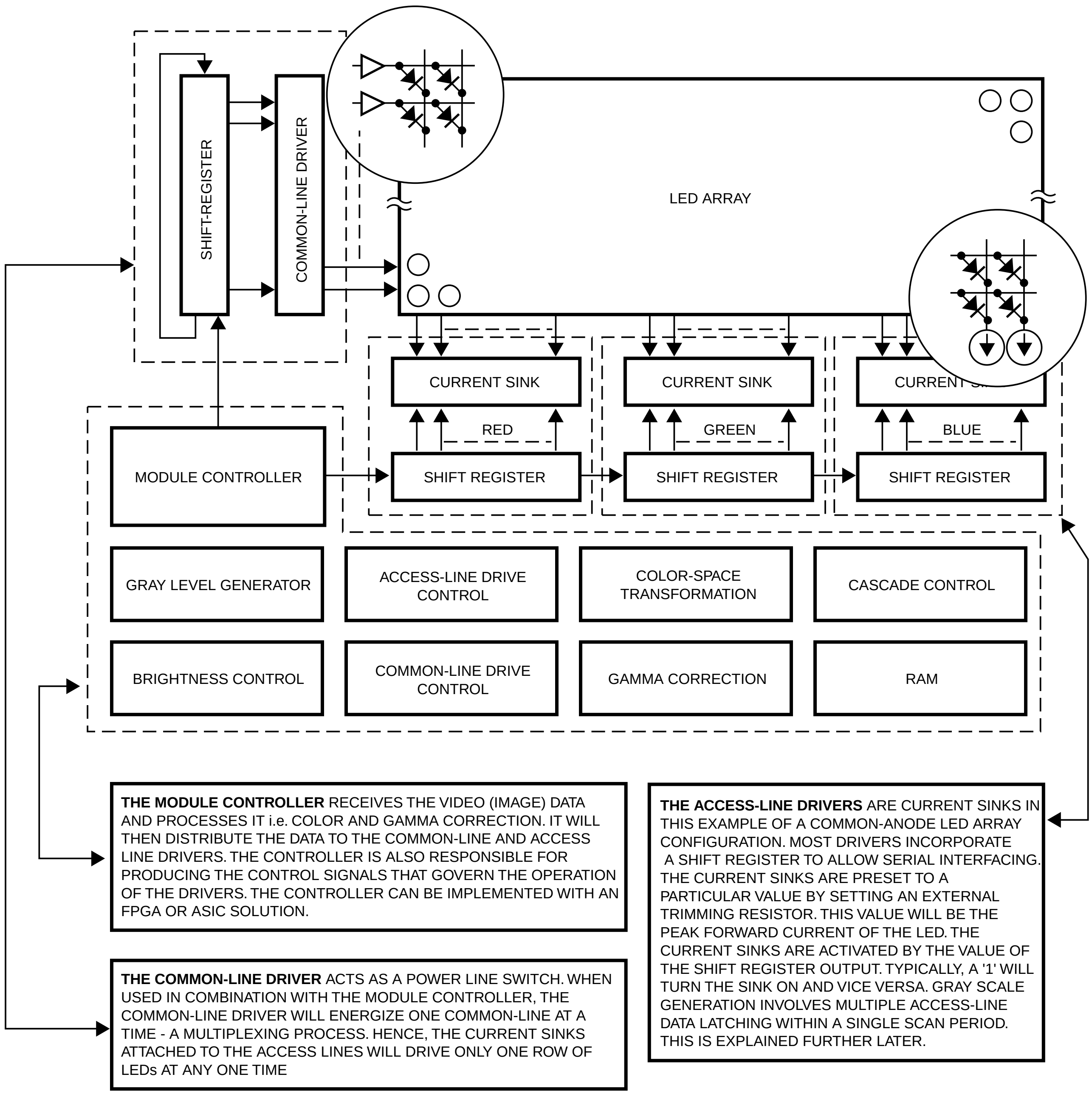


Figure 13. Internal structure of a module controller.

The module controller manages and processes the video data for the module. It has several key functions, which are shown in Figure 13 on the previous page. Each is described below:

1. **Common-line drive control** - manages the multiplexing action of the module. It controls which common-line is to be energized and synchronizes with the access-line drive control to ensure that the correct row data is fed into the access-line drivers.

2. **Access-line drive control** - determines which LED in the currently energized common-line row is to be turned ON. The TB62706 (Toshiba), for example, has 16 current-sink outputs and a shift register for serial interface. A "1" will turn on the LED and vice-versa.

3. **Gray scale control** - gray scales are the brightness levels each LED in a pixel can be controlled to. A typical full-color system can have up to 256 gray scales. That means that each LED's brightness can be tuned from minimum to maximum brightness in 256 steps. If each pixel has three LEDs - red, green and blue - the number of color combinations is 256x256x256 = 16.7 million.

   Gray scales are achieved by dividing each scan period into 256 time slots (more slots are needed for a larger number of gray levels). Thus, the access-line drivers are latched with data 256 times in a single scan period.

4. **Brightness control** - brightness control is different from gray scales. Brightness control refers to the control of the display's overall luminance value, not an individual LED. Manipulating the length of the scan period can control the overall luminance or brightness.

5. **Cascade control** - generates the control signals used to interface to the other modules.

6. **RAM** - stores the incoming video data. Usually, a double-buffering method because the video data received still needs to be processed. So, while one buffer is used to drive the display, the second buffer contains freshly received data to be operated on. These two processes can happen in parallel.

7. **Gamma correction** - corrects the non-linear transfer function of the LED screen. Put another way, the signal transfer between the electrical and optical components of the display system is non-linear. This leads to expansion of the bright region and compression of the dim region. NTSC and PAL video signals are gamma-corrected prior to transmission to eliminate the non-linear effect of the display. Hence, the display must take this into account to obtain a linear signal transfer function. This topic will be further discussed in the next section.

8. **Color-space transformation** - is necessary with rich-color tiles because its color space is limited compared to full-color tiles and color televisions. For this reason, rich-color tiles cannot be used for display of full-color video. It can be used in limited color video like cartoons. Again, this topic will be further discussed in the next section.

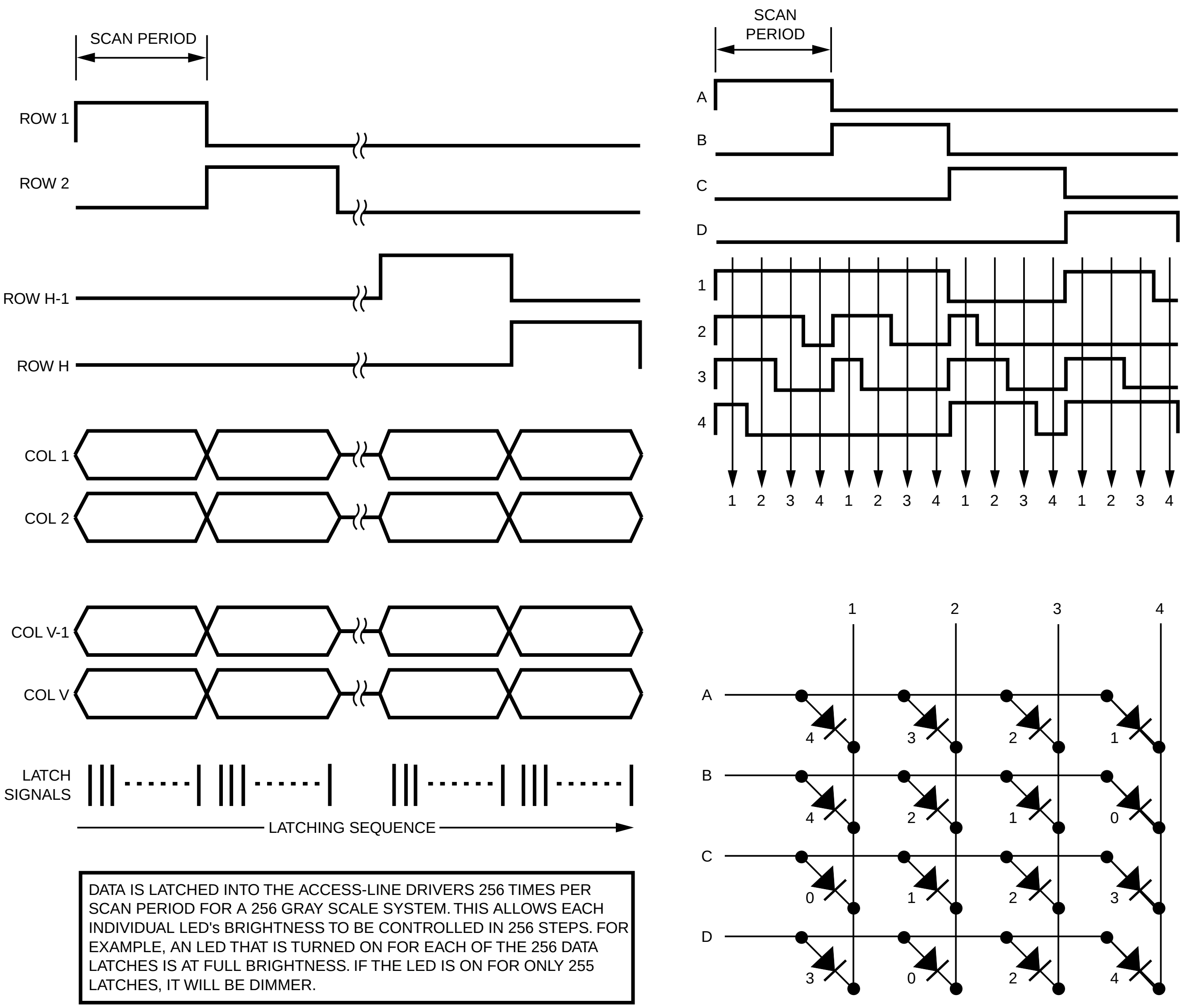


Figure 14. Implementing gray scale control. H = number of rows in a module and V = number of columns in a module. The arrows (in the top-right figure) indicate data latches into the access-line drivers - 4 per scan period in this example. The bottom-right figure shows the effect - 4 is brightest, 1 is dimmest and 0 is off.

The two figures on the right are taken from the earlier section. It shows the timing sequence used to produce a 4 gray scale system. We can observe from the top right figure that data is latched into the access-line drivers 4 times in a single scan period. That allows us to control the pulse width applied to each individual LED. Four gray scales are produced. Although the example is of a simple 4x4 LED matrix, the same concept applies for larger array of LEDs. In addition, if more gray scales are required, we simply need to increase the number of data latches to the access-line drivers within a single scan period.

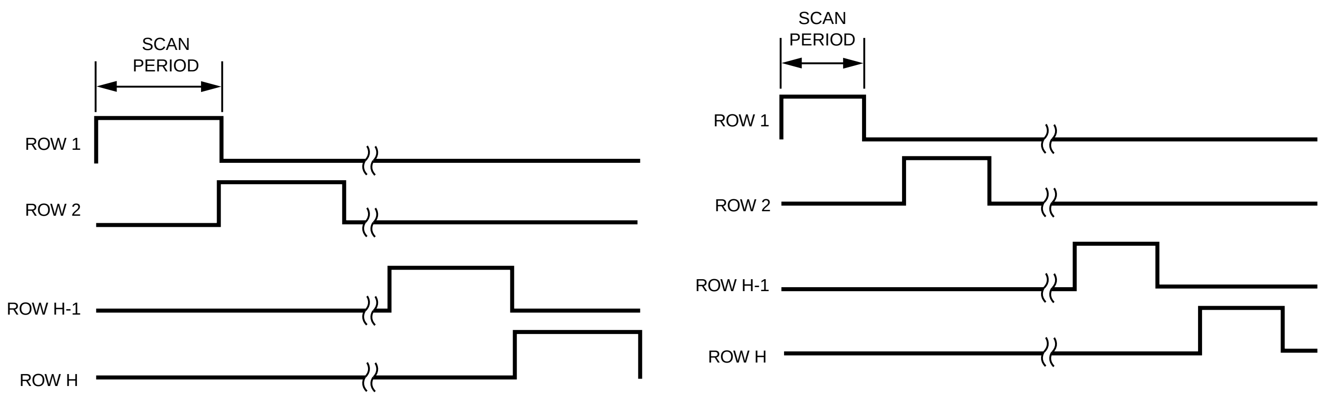
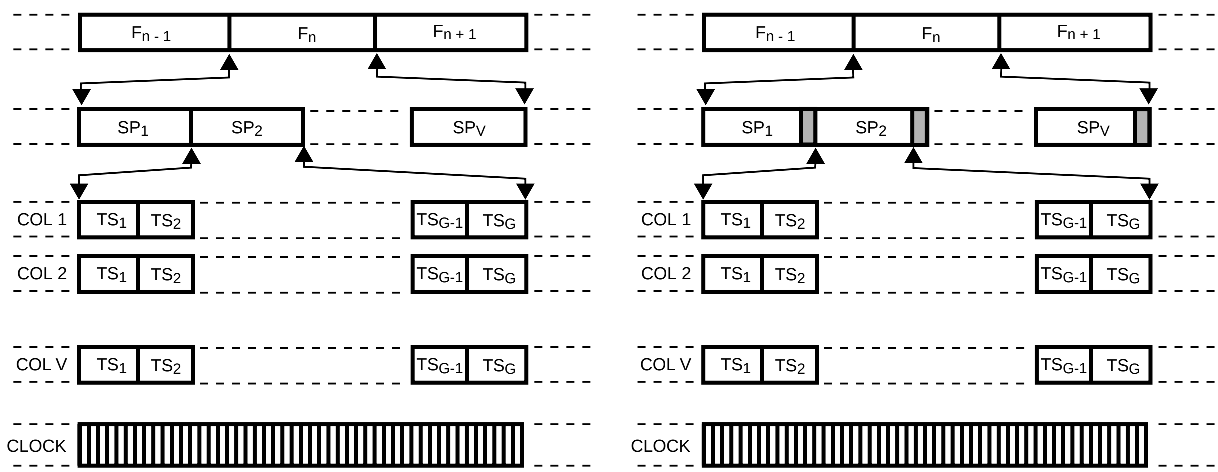


Figure 15. Brightness control by manipulating the length of a scan period. The overall luminance of the display will be brighter with longer scan periods (figure on the left).



F - Frame
n - Frame index
SP - Scan Period
H - number of rows in a module
V - number of columns in a module
G - number of gray scales (equivalent to number of time slots)

Figure 16. The concept of Pulse Width Modulation (PWM) for light output control. Manipulating the width of the Scan Period (SP) affects the overall brightness of the display screen (figure on the right). Increasing the number of data latches (time slots) per scan period increases the number of gray scales.

All the data latches (time slots) need to be contained in a single scan period. If the scan period is shortened to reduce the overall brightness of the screen (see Figure 16), the module controller must ensure that all the data latches occur within that shortened scan period.

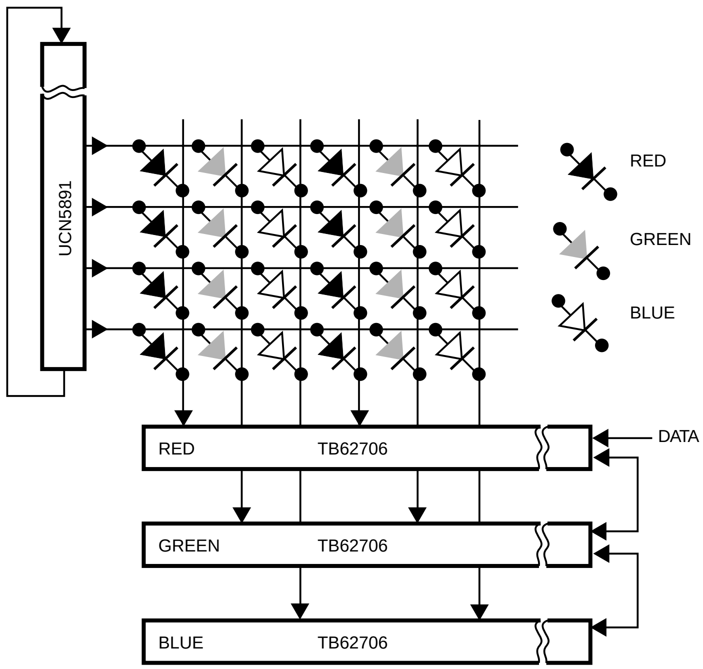


Figure 17. Interfacing a three-color LED array to the drivers. The UCN5891 incorporates a shift register. By connecting the serial output to the serial input and loading the register with "0"s, we can scan the common lines by shifting a single "1" through the register. The TB62706 has 16 current-sink outputs (only two outputs are shown above). We can either dedicate one data line per driver or cascade the three drivers in series (shown above). Cascading will reduce the number of data lines to one but requires a higher data rate.

Figure 17 is just an example. There are many other driver ICs that can be used (see Appendix A: Choosing a Driver). The common-line driver shown in the example has a built-in shift register. There are also parallel interface drivers available in the market.

Intelligent drivers

The previous section demonstrated the concept of using constant-current latches at the low side of the LED array. Although more expensive, intelligent drivers reduce the complexity of the module controller considerably. As shown below in Figure 18, intelligent drivers will manage the generation of gray scales. The module controller, however, still needs to manage the common-line driving. Brightness control is also offloaded to the intelligent drivers. Intelligent drivers are highly recommended for video applications.

Detailed explanation of these drivers is beyond the scope of this application note. Please refer to the appropriate driver data sheet. (E.g. TLC5911 from Texas Instruments. See Appendix B.)

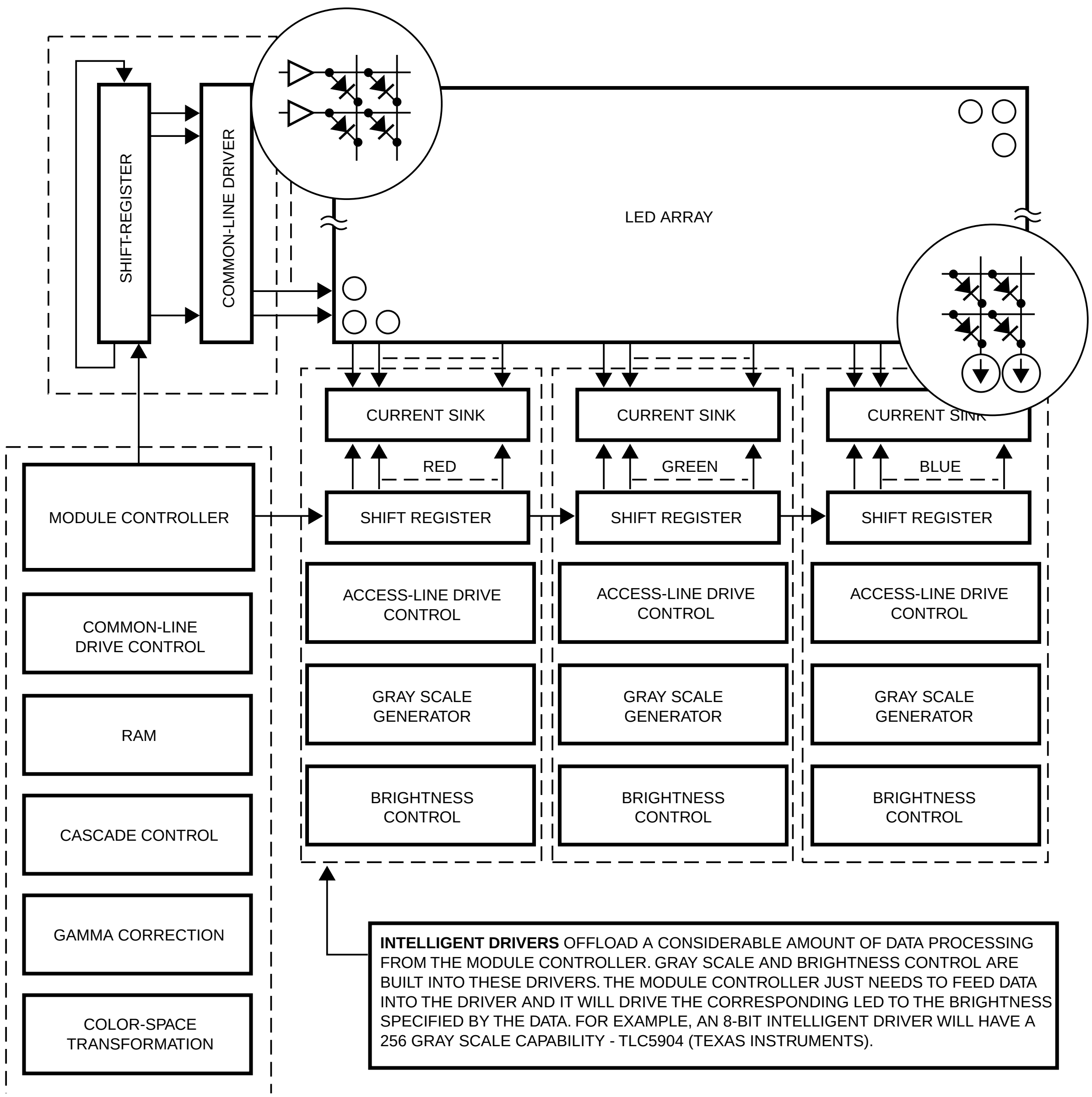


Figure 18. Using intelligent drivers.

Gamma correction

The transfer function between the electrical and optical components of a display system is non-linear. If this non-linearity is not compensated, high brightness regions are expanded and dim regions are compressed. The figure below shows the characteristic.

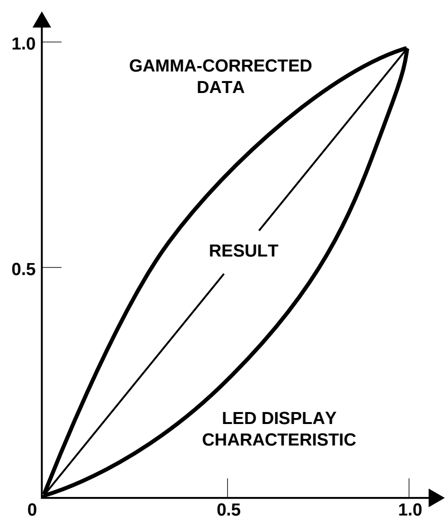


Figure 19. Gamma correction. The LED display's transfer function follows a power law. To compensate for the non-linearity, video data is gamma-corrected. The result is a linear transfer function.

As an example, CRT displays have a gamma of around 0.45. Hence,

$R_{display} = R_{in}^{2.2}$

$G_{display} = G_{in}^{2.2}$

$B_{display} = B_{in}^{2.2}$

where R_{in}, G_{in} and B_{in} are the data input into the display drivers. If we gamma-correct R_{in}, G_{in} and B_{in}:

$R_{in}` = R_{in}^{0.45}$

$G_{in}` = G_{in}^{0.45}$

$B_{in}` = B_{in}^{0.45}$

where $R_{in}`$, $G_{in}`$ and $B_{in}`$ are the incoming video signals from the video source:

$R_{display} = (R_{in}`^{0.45})^{2.2} = R_{in}`$

$G_{display} = (G_{in}`^{0.45})^{2.2} = G_{in}`$

$B_{display} = (B_{in}`^{0.45})^{2.2} = B_{in}`$

A linear relationship will then be established. Gamma-correction is usually implemented using a look-up table, e.g. PROM or DSP. However, digital circuitry has finite resolution. For that reason, the gamma-corrected data should have a higher resolution. Some screen builders use an 8-bit to 9-bit gamma correction while others use 10 bits for increased picture quality. T.I. has an intelligent driver that has gray scales up to 10 bits (TLC5911).

Picture quality is a subjective measurement. Display makers usually tune their screens for acceptable brightness, contrast, and gamma before deployment.

Color Space Transformation

Rich-color displays possess a color space that is significantly smaller than full-color displays. If the incoming video signals assume that the display it is driving has a full-color space, the colors on the rich-color display will appear compressed. Color-correction can be implemented using a look-up table as well. Detailed description on this subject is, however, beyond the scope of this note.

Data distribution

As mentioned earlier, it is best to break down a display system into the module level and the panel level. Each is responsible for data distribution at its own level. The module controller will be responsible for distributing data within its panel and a given panel controller will only receive data relevant for that given panel. The controller should ignore all other data. This simplifies management of data.

There are several methods for distributing data. Only one is discussed here. It is based on a ring topology.

In the model shown in Figure 22, there are several modules attached to a data pipe. The modules are allowed to "see" the data but are not allowed to latch the data into memory. It can only do so if it is in possession of a "token". This token can be in a form of a pulse travelling along a control line. Once any module receives a token, it immediately latches a chunk of video data. The module will pass on the token to the next module after it receives data for each of its pixels.

Panels can use the same approach for data distribution. Other methods include an Ethernet based model and address-line based data bus.

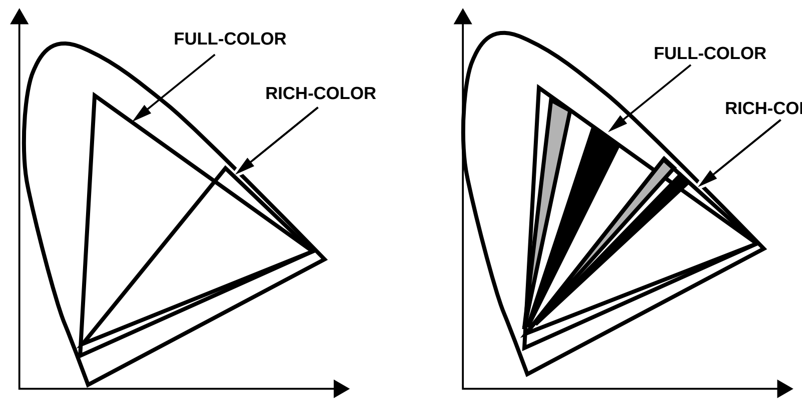


Figure 20. CIE chromaticity diagram showing the color spaces for a full-color display and a rich color display. The two bands of color are compressed on the rich-color space if color-correction is not performed.

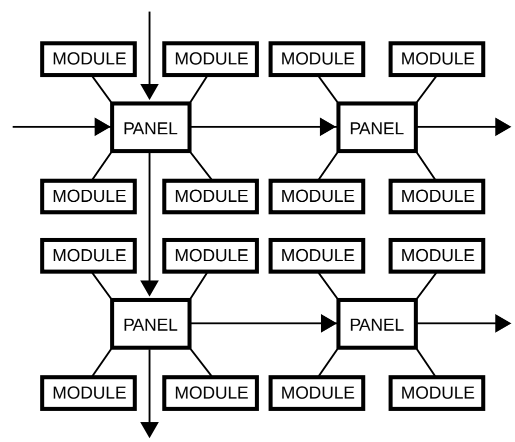


Figure 21. Data distribution. The example above shows a left-to-right, top-to-bottom model.

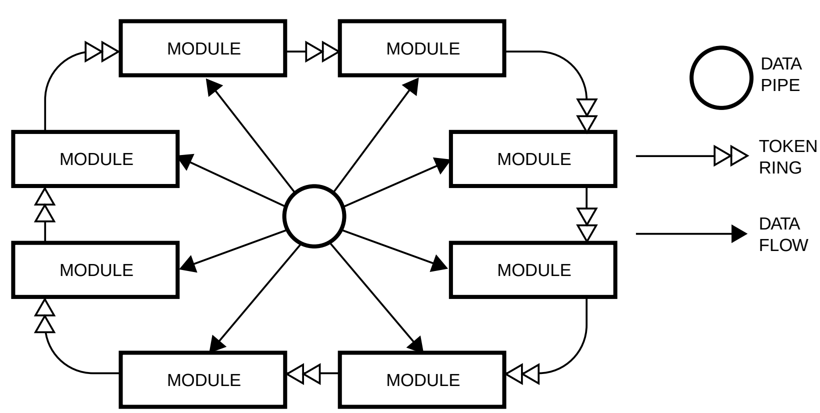


Figure 22. Token-ring based data distribution.

Appendix A: Choosing a Driver

Table 1. Characteristics of Various LED Drivers

Type	Limiting resistor	ASSP [1]	CC[2] latch driver	Intelligent driver
LED brightness consistency	Poor	*	Excellent	Excellent
Individual LED brightness control built-in?	No	*	No	Yes
Outputs	Single	Multiple	Multiple	Multiple
Forward current consistency	Fair	*	Excellent	Excellent
Maximum forward current	Limited by resistor power rating	Moderate (~50 mA)	Moderate (~90 mA)	Moderate (~90 mA)
Direct digital interface	No	Yes	Yes	Yes
Price [3]	$	$$$	$$$	$$$$
Additional comments	Use only when brightness consistency is not critical.	Designed for standard LED packages e.g. 7-segment display and 5x7 dot matrix.	Data usually stored in internal shift register (SIPO [4]). Current control via external resistor.	Brightness of individual LEDs can be digitally controlled (as fine as 1024 brightness levels)

Notes:
* Specifications vary manufacturer-to-manufacturer.
[1] Application Specific Standard Product
[2] Constant-current
[3] $ - cents/LED array, $$ - ~$0.50/output and $$$$ - ~$5.00/IC
[4] Serial-in, Parallel-out

Table 2. Choosing the Right Driver for the Application

Application	Recommended driver type	Comments
Backlight array (e.g. keypad lighting)	Limiting resistor	Use only when brightness consistency is not critical and low-cost is required. Otherwise CC latch driver is recommended.
Standard LED displays	ASSP	E.g. Seven-segment, 5x7 dot matrix and bar graph arrays.
Discrete lamp arrays	CC latch driver	IC allows microcontroller/processor interface.
Message panel	CC latch driver	Brightness consistency is critical.
Video	Intelligent driver	CC latch driver can be used when cost is an issue or if custom functions are required.

Appendix B: Access-line Drivers

CC driver latches and intelligent drivers are recommended for video applications. ASSPs are used for standard products such as 5x7 tiles and segmented displays.

Manufacturer	Part number	Type	Functions	Documentation
Toshiba	TB62705	CC latch driver	Constant current sink, up to 90 mA, 8 outputs, serial interface.	http://doc.semicon.toshiba.co.jp/noseek/us/td/06linia.htm
	TB62706	CC latch driver	As above except 16 outputs.	
	TB62717	CC latch driver	As above except 24 outputs	
	TB62715	CC latch driver	High constant sink current, up to 150 mA, 16 outputs, serial interface.	
	TB62708	CC latch driver	Constant current source, up to 90 mA, 16 outputs, serial interface.	
	TB62710	CC latch driver	As above except 8 outputs.	
	TB62709	ASSP	For seven segment displays, controls up to 4 digits, up to 40 mA, serial interface.	
	TB62713	ASSP	For 5x7 dot-matrix displays, up to 50 mA, serial interface.	
	TB62718	Intelligent driver	Constant current sink, up to 90 mA, 8-bit gray scale control, 16 outputs, thermal shutdown, output open detection, parallel interface.	
Allegro Microsystems	A6275	CC latch driver	Drop-in for TB62705.	www.allegromicro.com
	A6276	CC latch driver	Drop-in for TB62706.	
Texas Instruments	TLC5921	CC latch driver	Constant current sink, up to 80 mA, 16 outputs.	www.ti.com
	TLC5920	ASSP	For 16x8 dot matrix displays, 16 current sinks and 8 common line drivers, up to 30 mA per output.	
	TLC5904	Intelligent driver	Constant current sink, up to 80 mA, 8-bit gray scale control, 16 outputs, thermal shutdown, output open detection, serial interface.	
	TLC5905	Intelligent driver	As above except serial data input.	
	TLC5910	Intelligent driver	Constant current sink, up to 80 mA, 10-bit gray scale control, 16 outputs, thermal shutdown, output open detection, parallel interface, 6-bit brightness correction feature.	
	TLC5911	Intelligent driver	As above except 7-bit brightness correction feature.	
Philips	HEF4511B HEF4543	ASSP	For seven segment displays, BCD-to-7 segment decoder driver.	www.philips.com
Maxim	MAX7219 MAX7221	ASSP	For seven segment displays, controls up to 8 digits, serial interface.	www.maxim-ic.com

(continues)

Appendix B: Access-line Drivers (continued)

Manufacturer	Part number	Type	Functions	Documentation
National Semiconductor	MM5486	CC latch driver	Constant current sink, up to 15 mA, 33 outputs, serial interface.	www.national.com
Fairchild Semiconductor	MM74C911 MM74C912 MM74C917	ASSP	Segment display drivers, contains both segment drivers and common line (digit) drivers.	www.fairchildsemi.com
	CD4511BC DM7446A M7447A	ASSP	Seven segment display driver, BCD decoder.	
	DM9368	ASSP	As above with constant source outputs.	
	DM9374	ASSP	As above except with sink outputs.	
Intersil	ICM7217	ASSP	Seven segment display driver, up to 4 digits.	www.intersil.com
	ICM7228	ASSP	Seven segment display driver, up to 8 digits, on-board display RAM.	
	ICM7243	ASSP	14 or 16 segment display driver, on-board ASCII decoder and display RAM.	
Sgs-Thompson	M54HC4511	ASSP	For seven segment displays, BCD-to-7 segment decoder driver.	www.st.com
	M5450	CC latch driver	Constant current sink, up to 15 mA, 34 outputs, serial interface.	
	M5451	CC latch driver	As above except 35 outputs.	
	M5480	ASSP	3 $^1/_2$ digit driver, 23 outputs, constant current sink, serial interface.	
	M5481	ASSP	Seven segment driver, 14 outputs, constant current sink, serial interface.	
	M5482	ASSP	As above except 15 outputs.	
Motorola	MC14499	ASSP	Seven segment driver, controls up to 4 digits, serial interface.	www.motorola.com
	MC14489	ASSP	As above except controls up to 5 digits.	

Appendix C: Common-line Drivers

This appendix covers drivers in IC form. However, discrete transistors work just as well although they require more board space.

Manufacturer	Part number	Type	Functions	Documentation
Toshiba	TD62708	Source driver	8-channels, high-side, 1.8 A, active low chip enable.	http://doc.semicon.toshiba.co.jp/noseek/us/td/06linia.htm
	TD62783	Source driver	8-channels, high-side, 0.5 A, active high.	
	TD62785	Source driver	8-channels, high-side, 0.5 A, active low.	
Allegro Microsystems	UCN5891	Source driver	8-channels, high-side, 0.5 A, built-in shift register for serial interface.	www.allegromicro.com
	UDN2981	Source driver	8-channels, high-side, 0.5 A, active high.	
	UDN2540	Power driver	Quad Darlington transistors, 2.5 A, parallel digital interface.	
	UDN2597	Sink drivers	8-channels, low-side, 0.5 A, active high.	
	UDN2987	Source drivers	8-channels, high-side, 0.5 A, active high, over-current protection.	
Sgs-Thompson	ULN2064	Transistor array	Quad Darlington output, 1.5 A.	www.st.com
	ULN2803	Transistor array	8 Darlington outputs, 500 mA.	

www.semiconductor.agilent.com

Data subject to change.

May 7, 2001
5980-3132EN

Agilent Technologies