TI Designs
16-Ch Status LED Driver for PLC Modules

TI Designs

TI Designs provide the foundation that you need including methodology, testing, and design files to quickly evaluate and customize the system. TI Designs help *you* accelerate your time to market.

Design Resources

TIDA-00560	Tool Folder Containing Design Files
TLC5928	Product Folder
SN74LVC2G86	Product Folder
SN74LVC1G332	Product Folder
BeagleBone Black Wiki	BeagleBone Resource Folder

TI E2E™ Community

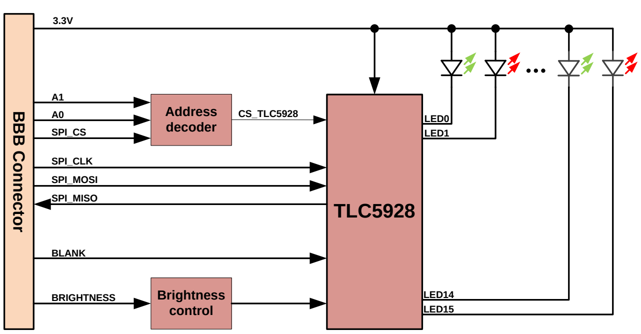
ASK Our E2E Experts
WEBENCH® Calculator Tools

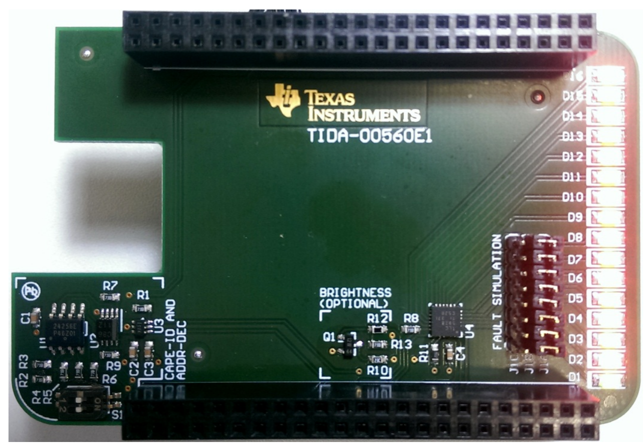
Design Features

- 16-Channel Shift Register With Wide Range Constant Current Sinks (2 mA to 35 mA)
- Two-Level Brightness Setting for Light-Emitting Diode (LED) to Compensate for Ambient Light
- Diagnostic Features: LED Open and LED Shorted to GND
- Standardized SPI Connects Seamlessly to Standard Processors
- Number of Channels Increasable by Daisy-Chaining Multiple LED Driver Devices
- BeagleBone Cape Support for Easy Evaluation

Featured Applications

- Programmable Logic Controller (PLC)
- PLC Analog Input and Output modules
- PLC Digital Input and Output modules





An IMPORTANT NOTICE at the end of this TI reference design addresses authorized use, intellectual property matters and other important disclaimers and information.

1 Key System Specifications

PARAMETER	SPECIFICATION	DETAILS
LED channels	Number of LED channels: 16	See TLC5928 data sheet (SBVS120)
V_{CC}	Supply voltage range: 3.0 V to 5.5 V	See Section 5.1.1
I_{LED}	Brightness signal low or high: 2 mA or 5 mA	See Section 5.1.3
I_{CC}	I_{CC} (LEDs off), 3.3 V, no data transfer: 0.535 mA I_{CC} (LEDs on, 2 mA), 3.3 V, no data transfer: 31.89 mA I_{CC} (LEDs on, 5 mA), 3.3 V, no data transfer: 79.9 mA	See Section 8.3.4
Luminous intensity	Red: 20 mcd at 2 mA, 50 mcd at 5 mA Green: 4 mcd at 2 mA, 10 mcd at 5 mA	See Section 4.1.3 See Section 5.1.3
HW brightness control	Changing forward current in two steps	See Section 5.1.3
LED fault simulation	LEDs D1-D8	See Section 5.1.4
System control	Any host processor using serial peripheral interface (SPI) or general purpose input and output (GPIO)	See Section 7
Board integration	Compatible with BeagleBone Cape specification	See Section 5.2
Operating conditions	−40°C to +85°C	See TLC5928 data sheet (SBVS120)

2 System Description

This TIDA-00560 TI design demonstrates a way to control up to 16 LEDs independently as required in systems with multiple channels, such as analog or digital PLC I/O modules. The very small board space of the core components and the standard interface of the LED driver allow seamless integration into small form factor modules.

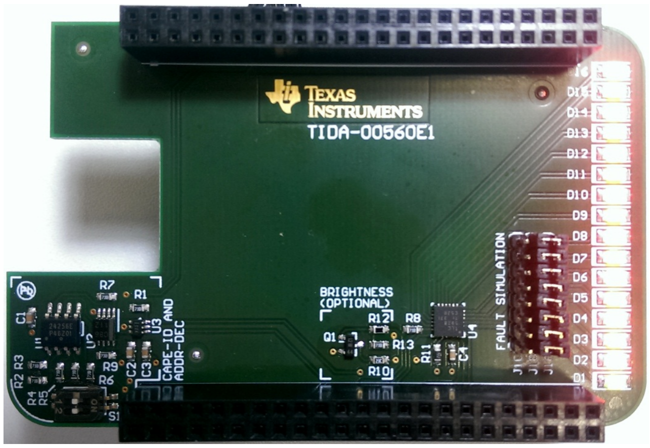


Figure 1. TIDA-00560 Board Image

2.1 *TLC5928*

The main part of this TI design, the TLC5928 [1], is a 16-channel, constant-current LED driver with fault detection. The current is selectable with a resistor and can be in the range of 2 mA to 35 mA. The channel-to-channel accuracy is typical at 1%. The daisy-chain feature allows cascading of multiple TLC5928 devices in a chain. The device-to-device accuracy is also typical at 1%. While the digital part works in the range of 3.0 V to 5.5 V, the current-sink input voltages can be as high 17 V supporting multiple LEDs per current sink in series. A dedicated BLANK pin allows the sudden shut off of all LEDs without SPI communication.

3 Block Diagram

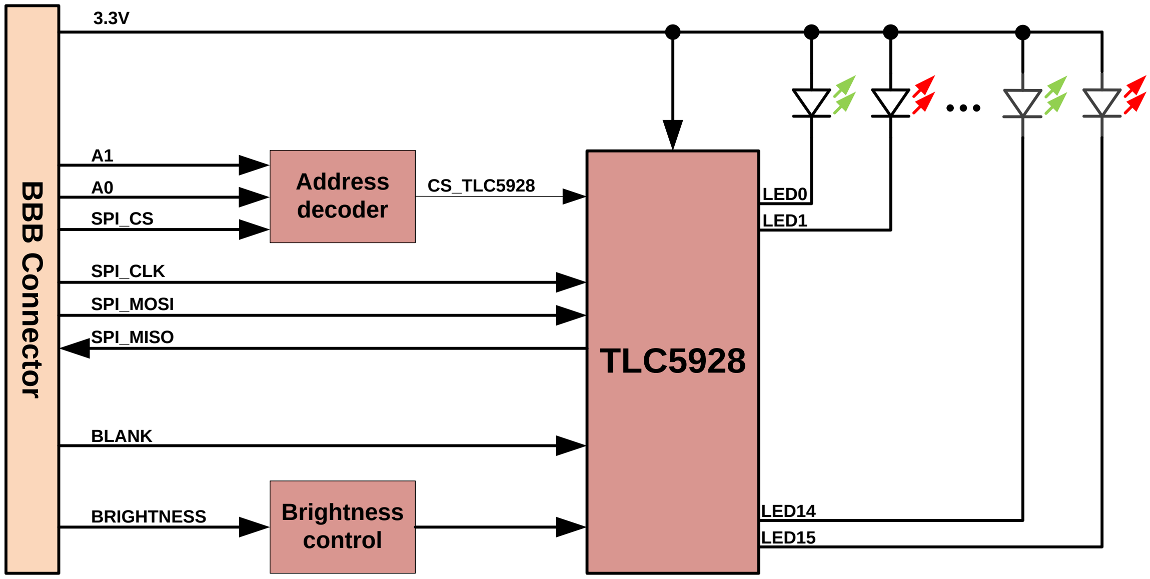


Figure 2. TIDA-00560 Block Diagram

3.1 Highlighted Products

The TI design for a 16-channel LED driver features the following device:

- TLC5928
  - 16 channels, constant-current sink output
  - 35-mA capability (constant-current sink)
  - 10-ns high-speed constant-current switching transient time
  - LED power-supply voltage (up to 17 V)
  - V_{CC} = 3.0 V to 5.5 V
  - Constant-current accuracy:
    - Channel-to-channel = ±1%
    - Device-to-device = ±1%
  - 35-MHz SPI data transfer rate
  - Readable error information:
    - LED open detection (LOD)
    - LED short to GND
    - Pre-thermal warning
    - Operating temperature: –40°C to +85°C

Figure 3 shows the block diagram for the TLC5928 device.

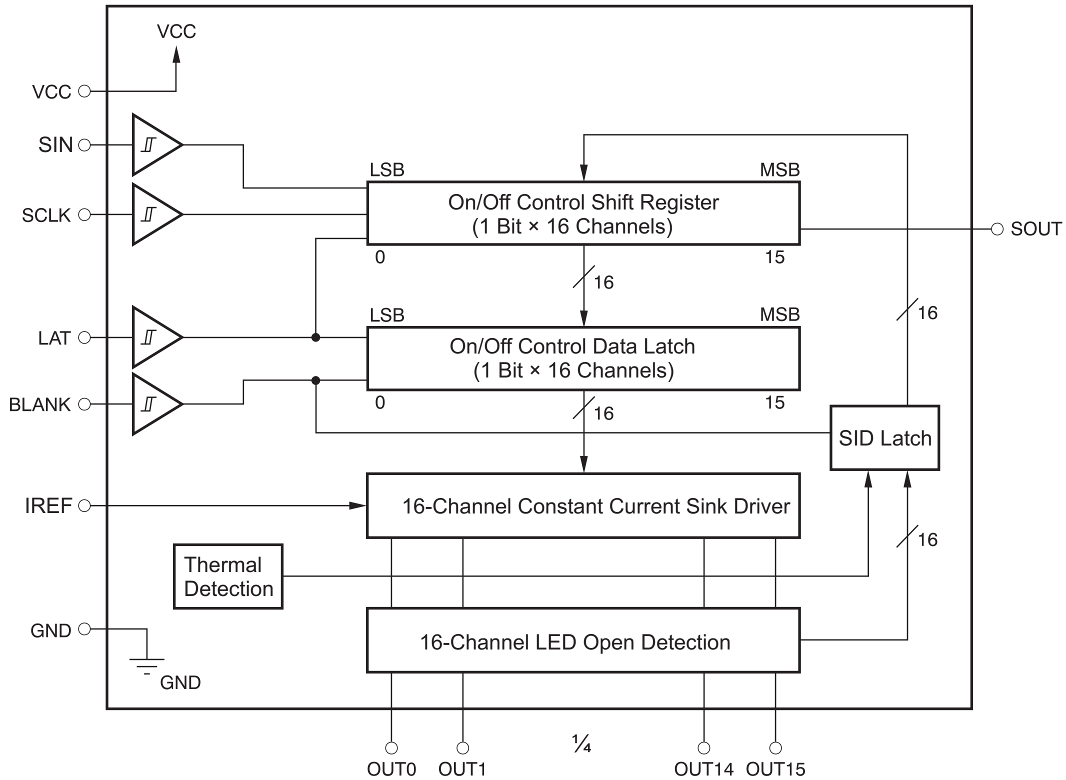


Figure 3. TLC5928 Block Diagram

For more information see the respective product folder at: www.ti.com/product/tlc5928.

4 System Design Theory

4.1 Methods of Driving LEDs

Designers currently use different methods to drive an LED. The system environment and performance requirements dictate which method to use. For example, if an LED just indicates that a voltage is available (actual brightness is secondary), the LED simply connects to the observed voltage with a resistor in series. However, this approach is insufficient where brightness control is a requirement. To better describe each approach, the following paragraph provides a brief description of the LED characteristics.

Every LED (and every other diode) has a U-I characteristic similar to what Figure 4 shows. As an example, Figure 4 shows the curve of a Wurth Elektronik LED (part number 150120VS75000) as used in this TI design.

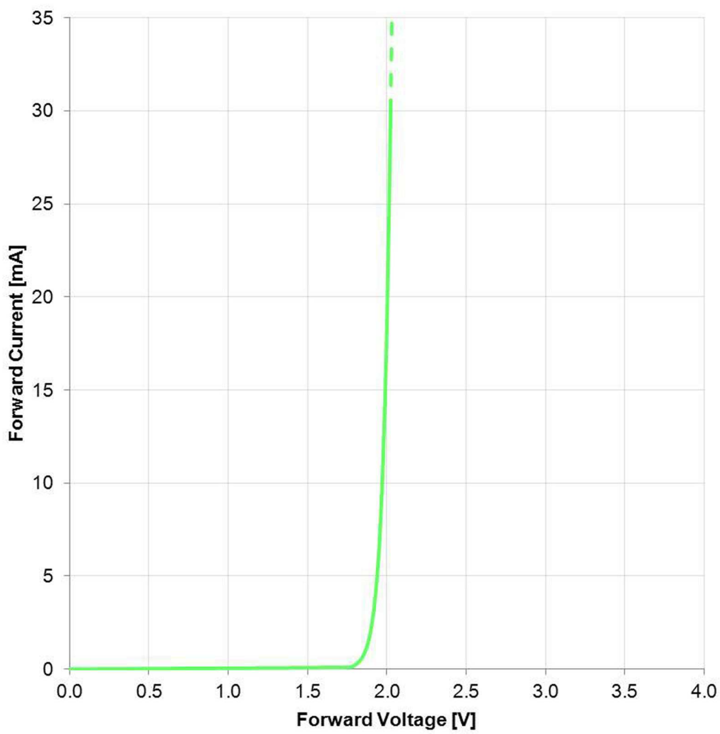


Figure 4. Forward U-I Curve of LED

Given the steepness of the curve in the area of interest (here: 1.9 V to 2.0 V), the user can discern that small changes in the forward voltage result in a larger change of the forward current. Rather than trying to set the voltage, a more practical way is to inject the current and let the voltage follow. Setting the forward current also enables better control of the brightness because the forward current and the luminous intensity are almost linear dependent (see Figure 5).

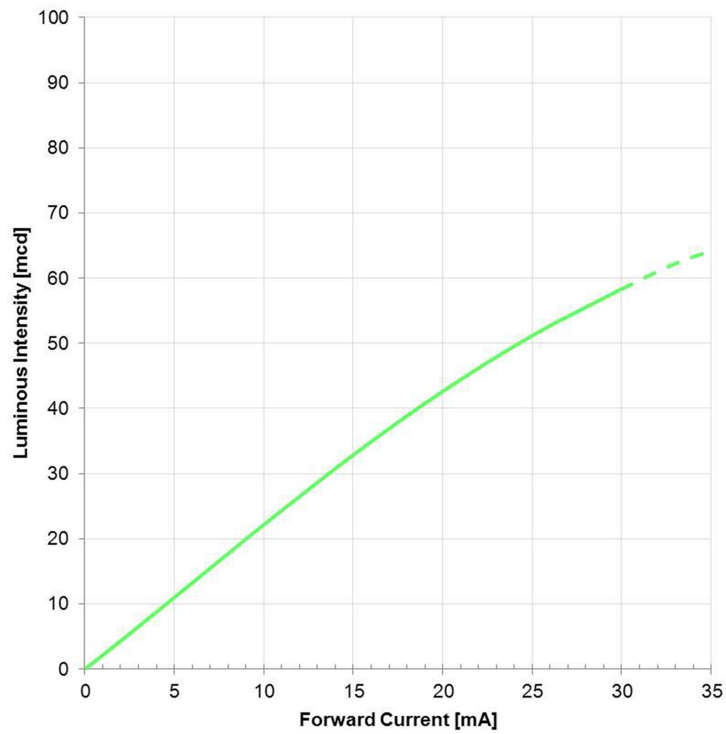


Figure 5. Forward Current — Luminous Intensity Curve

4.1.1 Driving Single LED Using GPIO

The simplest way to turn an LED on or off is to use a general-purpose input or output (GPIO) pin (see Figure 6).

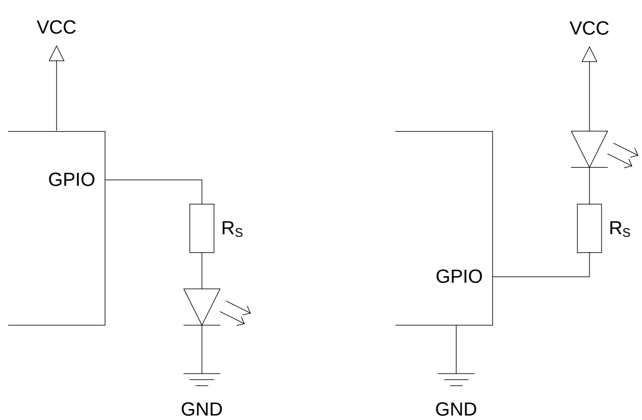


Figure 6. Driving LED With GPIO

The circuit just requires a resistor in series to limit the current. Equation 1 calculates the value of the resistor:

$$R_S = \frac{V_{GPIO} - V_{LED}}{I_{LED}} \quad (1)$$

However, R_S cannot be calculated precisely due to reasons such as:

- Production-related deviation in V_{LED} (at a given I_{LED})
- Current-dependent V_{GPIO}

In addition to these reasons, the calculated value of R_S might not be available. Selecting the next available value also adds an error.

The standard GPIOs of microcontrollers and microprocessors are not designed to drive higher currents, in that the R_{DSON} of their high-side and low-side MOSFETs are not optimized. As more current flows through the MOSFET, the voltage drop over its source-drain increases, which results in a voltage shift at the GPIO pin. The datasheet of the device usually specifies this characteristic. As examples, Figure 7 (sinking current) and Figure 8 (sourcing current) show the plots for the MSP430G2553 low-power microcontroller [2].

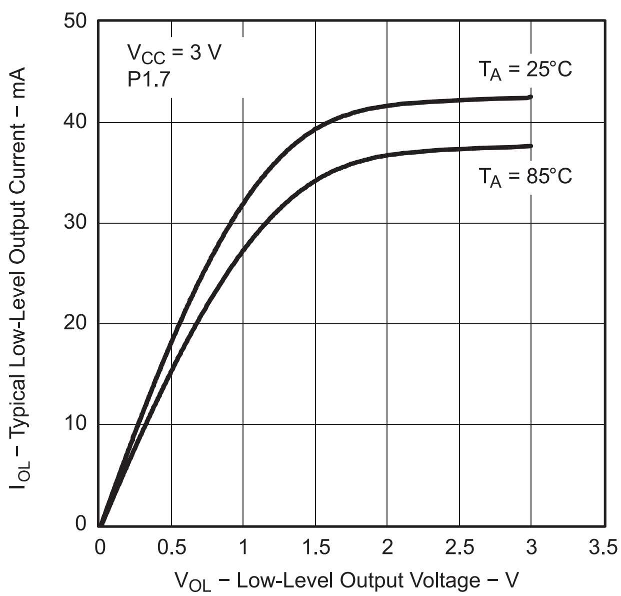


Figure 7. Voltage Shift During Current Sink

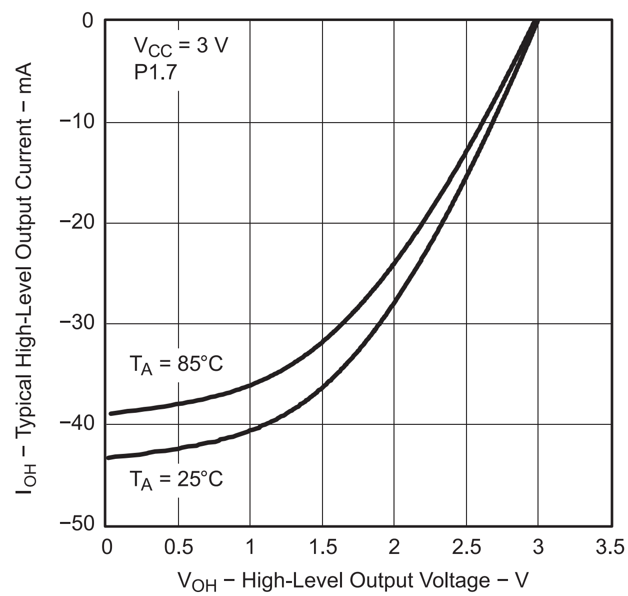


Figure 8. Voltage Shift During Current Source

The attentive reader may notice that the voltage drops are not similar for a certain current. Taking the 30-mA point from the 25°C curve: the ground shift caused by the low-side MOSFET is around 0.9 V (see Figure 7), while the voltage drop induced by the high-side MOSFET for the same current is around 3.0 V − 1.9 V = 1.1 V (see Figure 8). This difference in voltage drops means that the right-side circuit in Figure 6 performs slightly better than the left-side option. If a user desires better control over the voltage drop or requires driving a higher current, TI recommends using an external MOSFET.

4.1.2 Driving Multiple LEDs With GPIOs

The previous solution that Section 4.1.1 outlines requires one GPIO per LED channel. If the number of LEDs that must be driven are more than the number of GPIOs available, the user can implement a GPIO expander. Expanders usually have a standard serial interface as an input and multiple GPIOs as outputs. A commonly used serial interface for this type of device is the I^2C bus. Because the I^2C bus is shareable among multiple devices (memory, sensor, and so forth), adding GPIOs does not require an additional pin. A good example of this is the PCF8574 device [3].

Another way to add GPIOs is to use a standard logic device, such as the SN74HC595 [4] with an SPI-compatible serial port. The values for the 8-bit output are serially clocked into the shift register and transferred to the storage register with a separate signal to avoid an unintended GPIO state change during the time of new data arrival.

4.1.3 Driving LED With Integrated Current Source

The more efficient way to drive an LED is to control it using a current source. The forward voltage is the result of the current being injected; therefore, the current source must be able to provide the expected forward voltage. By supporting higher voltages, multiple LEDs can be connected in series while maintaining the same brightness because the same current flows through this string of LEDs.

A designer can simply create current sources with a few components, but the bill of materials (BOM) and real estate requirements increase with multiple channels. This TIDA-00560 TI design shows the simplicity of a multi-channel current-driven LED design using the TLC5928 device.

The current limit for the sources is set by an external resistor connected to the TLC5928 device. The formula to set the current limit can be calculated using the following Equation 2:

$$R_{IREF} = \frac{V_{IREF}}{I_{LED}} \times 42 \quad (2)$$

With a constant V_{IREF} of 1.2 V, the R-I curve can be calculated. Figure 9 shows the plot over the entire supported LED current range of 2 mA to 35 mA.

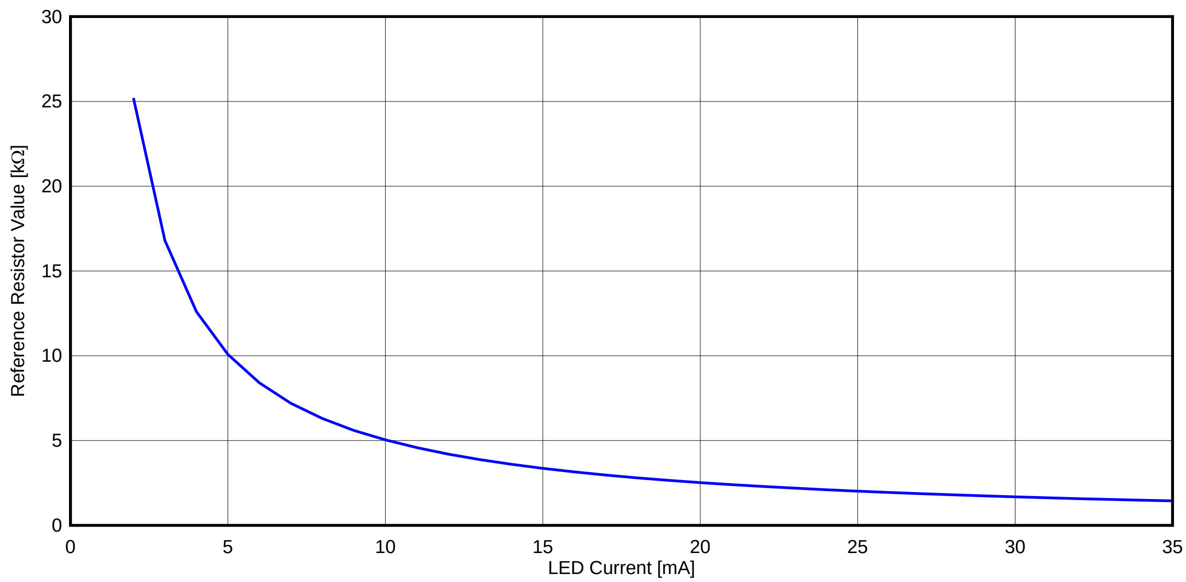


Figure 9. LED Current Versus Resistor Value

4.1.4 Board Space

Another important aspect of the TIDA-00560 design is the real estate required by the solution. While this design is mechanically compatible with the specifications listed for the BeagleBone Cape, the actual LED driver solution can be much smaller. The TLC5928 device comes in a 4 mm × 4 mm QFN-24 package and the absolute minimum of required additional components are the current-limiting reference resistor and the bypass cap. Because no additional resistors for the LEDs are required, this design is the most physically compact with the shortest BOM among all of the previously mentioned options.

5 Getting Started Hardware

5.1 Running the TI Design

Although the TIDA-00560 TI design is designed to run on a BeagleBone Black Cape, connecting to any other processor with SPI or without SPI (emulation required through GPIO) is possible. Table 1 shows the assignments of headers J8 and J9.

Table 1. Pin Assignments

PIN ON BOARD	SIGNAL	DESCRIPTION	MANDATORY FOR OPERATION
J8.11	CAPE_A1	Cape address decoding	No
J8.12	CAPE_A0	Cape address decoding	No
J9.19	I2C_SCL	Cape identification	No
J9.20	I2C_SDA	Cape identification	No
J9.25	BRIGHTNESS	Brightness adjust	No
J9.27	BLANK	Blank signal	Yes
J9.28	SPI_CS	SPI chip select	Yes
J9.29	SPI_MISO	Data from TLC5928	Only for diagnostics
J9.30	SPI_MOSI	Data to TLC5928	Yes
J9.31	SPI_SCLK	SPI shift clock	Yes

The signals CAPE_A0 and CAPE_A1 are pulled-high by resistors R7 and R9 (see Figure 14). For proper operation of the address decoder logic, the user must set both S1 switches to “off” if the CAPE_A0 and CAPE_A1 signals are not driven. This assures that the I^2C address (set by S1) matches the address of CAPE_A0/CAPE_A1 and the SPI chip select (CS) signal is passed to the TLC5928.

The optional I^2C bus is required in conjunction with the specification designated by the BeagleBone Cape; Section 5.2 describes this in further detail. This feature is not required for proper operation, though.

The optional signal BRIGHTNESS selects between the default LED current of 2 mA (BRIGHTNESS set low) or 5 mA (BRIGHTNESS set high). If this signal is not driven, an LED current of only 2 mA is supported.

The diagnostic features of the TLC5928 device (*LED open*, *LED short to ground*, and *thermal*) is optional. If not required, the signal SPI_MISO can be omitted.

The TIDA-00560 design requires a three-wire SPI and the BLANK pin at the minimum for proper operation.

5.1.1 Voltage Range

The TLC5928 requires two different voltages:

- V_{CC}: 3.0 V to 5.5 V
- V_O: up to 17 V

In a digital environment, from where the TLC5928 is driven, a common voltage for V_{CC} is 3.3 V. Setting V_O to also run from this voltage is also effective to save the second voltage rail; therefore, check for the minimum V_O for this design. To obtain the supported voltage range, the user must know that each current sink in the TLC5928 requires a certain voltage drop over the OUTx pin to properly regulate the sink current (see Figure 10).

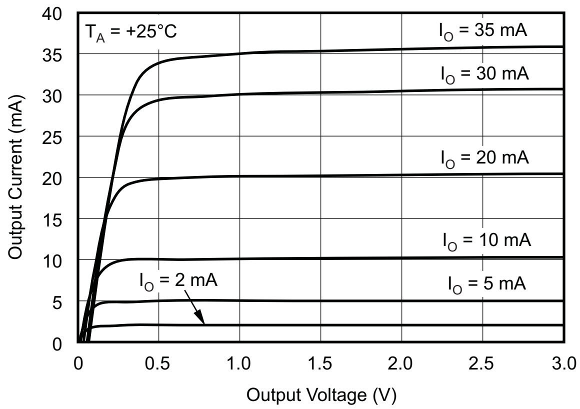


Figure 10. Required Voltage Drop for Stable Current Regulation

Figure 10 clearly shows that the amount of voltage drop is dependent on the target sink current. This TI design is optimized for 2-mA and 5-mA sink currents. According to Figure 10, the regulation is stable at around the 0.5-V voltage drop. By adding the working point voltage drop (V_{LED}) of the LED (conservatively 2.0 V—see Figure 4), a $V_O \geq 2.5$ V is required. The comparatively lower forward voltage for colored LEDs (compared to blue or white LEDs) as used in PLC I/O modules is an advantage here. As a result, the design works in the full range of the digital supply of the TLC5928 (V_{CC}), including the target of 3.3 V.

The difference between the minimum V_O (V_{O_MIN}) and the applied V_O must be consumed by the current sinks, increasing the power dissipation. The closer the applied V_O gets to V_{O_MIN} the better. Just generating a new voltage rail for V_O may not be economical, but if a voltage rail closer to V_{O_MIN} is available, the user should consider using this closer voltage rail to minimize the power consumption of the circuit.

5.1.2 Power Dissipation

As with every thoroughly designed circuit, the user must calculate the power dissipation, followed by any appropriate actions, if required. The used QFN-24 package is capable of dissipating 1615 mW at 85°C. The maximum power of the TLC5928 device with the given parameter is calculated in Equation 3:

$$P_{D\_MAX} = 16 \times \left(V_{OUT} \times I_{LED}\right) + V_{CC} \times I_{CC\_MAX(est)}$$
$$P_{D\_MAX} = 16 \times \left(\left(3.3\text{ V} - 2.0\text{ V}\right) \times 5\text{ mA}\right) + 3.3\text{ V} \times 5\text{ mA}$$
$$P_{D\_MAX} = 120.5\text{ mW} \tag{3}$$

A P_{D_MAX} of 120.5 mW for this design is far below the limit of 1615 mW. This number applies with the use of a soldered power pad. TI highly recommends to always solder the power pad for proper heat dissipation and mechanical stability reasons.

5.1.3 Brightness Control

The current limit for the output channels is set with a single resistor from pin IREF to GND. This TIDA-00560 TI design features an optional two-step brightness approach. This approach is useful in conjunction with an ambient light sensor such as the HDC1000 [5] to compensate for ambient light changes. Figure 11 shows the circuit.

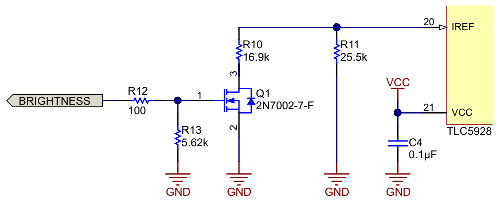


Figure 11. Brightness Circuit

In default configuration, the digital signal BRIGHTNESS is low and Q1 is not conducting. The IREF pin of the TLC5928 device is connected to GND by R11 only. The value of 25.5 kΩ sets the current limit for each channel to about 2 mA. If BRIGHTNESS is set high, Q1 is conducting and puts R10 parallel to R11, which results in a resistance of about 10.16 kΩ registered by the TLC5928 device. In this case the current limit rises to about 5 mA.

5.1.4 Simulating Fault Detection

In addition to switching the LEDs, each current sink can be observed for an *LED open* or *short to GND* event. The TLC5928 device internally senses the voltage drop across each OUTx pin to GND. If the nominal voltage is less than 300 mV, the devices assumes a fault and reports it to the processor over the MISO line during SPI communication. Both fault conditions can be tested with this TI design (see Figure 12).

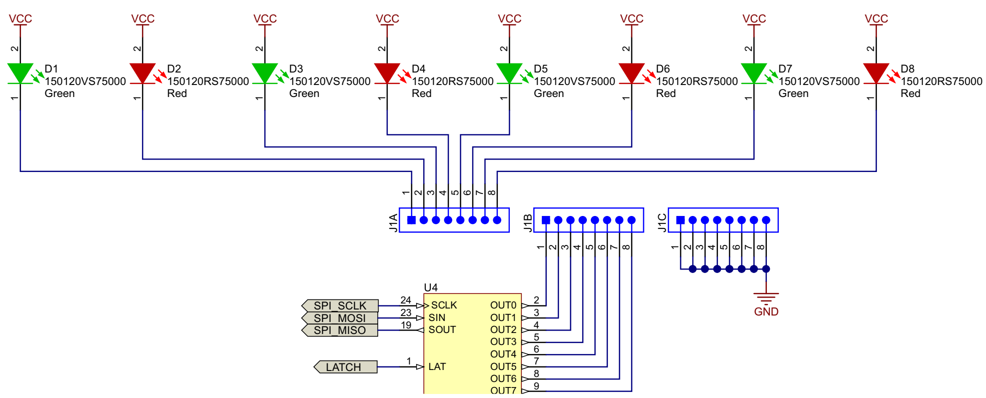


Figure 12. Fault Simulation

All of the sixteen channels support the *LED open* and *short to GND* features, although only the first eight channels (OUT0 through OUT7) utilize these features in this TIDA-00560 TI design. The behavior is understood for OUT0 (D1) but also applies to the remaining seven channels; the default jumper setting is a shorted J1A.1 and J1B.1, which connects the cathode of D1 to the OUT0 pin of the TLC5829 device. If no jumper is inserted, the *LED open fault* is simulated. Finally, if J1B.1 and J1C.1 is shorted, the OUT0 pin is connected to GND, simulating the *short to GND* fault. Additionally, the TLC5928 device can also report an *over temperature* condition, setting all bits of the 16-bit status word to “1”.

5.2 BeagleBone Cape Concept

This TIDA-00560 TI design is compatible with the BeagleBone Cape concept. The design allows a seamless connection to the BeagleBone and BeagleBone Black development platforms for easy evaluation. The implemented handshake mechanism allows for an automatic adaption of the BeagleBone GPIOs using device tree overlay files. The mandatory hardware for each cape is an EEPROM that stores all of the relevant information of the cape. The EEPROM is connected to the serial interface I^{2}C of the AM3359 device [6] used with the BeagleBone platform. During Linux startup, the AM3359 device scans the address range 0x54 to 0x57 for any cape board extensions (as up to four capes can be supported at a time). The address is selected by switch S1. Table 2 shows the mapping.

Table 2. S1 to I^{2}C Address Mapping

I^{2}C ADDRESS	S1.1	S1.2
0x54	On	On
0x55	Off	On
0x56	On	Off
0x57	Off	On

Figure 13 shows the circuit for the cape identification. This EEPROM is not protected here, which means it can be written to at any time.

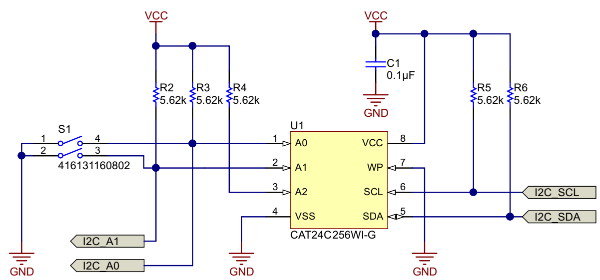


Figure 13. Cape Identification

Now imagine a scenario where four boards of this TIDA-00560 TI design are stacked. All TLC5928 devices would connect to the same SPI port and the AM3359 would not be able to distinguish between these devices. To solve this issue, this design implements an additional address decoding logic for the SPI CS signal. The SPI CS signal of the SPI is routed to the TLC5829 as the signal LATCH, but only if the signal CAPE_A0 (J8.12) matches the signal I20_A0 (set by S1.1) and the signal CAPE_A1 (J8.11) matches the signal I20_A1 (set by S1.2). If the addresses do not match, signals SPI_SCLK and SPI_MOSI still reach the TLC5928 device, but they have no impact because the SPI CS stays high. More importantly, the address logic assures that only one TLC5928 drives the SPI_MOSI at a time.

Driving more than one output may destroy the output buffer of a TLC5829 DOUT pin (connected to SPI_MOSI), and in the worst-case scenario, this damage may be irreversible. See Figure 14 for the implementation in hardware.

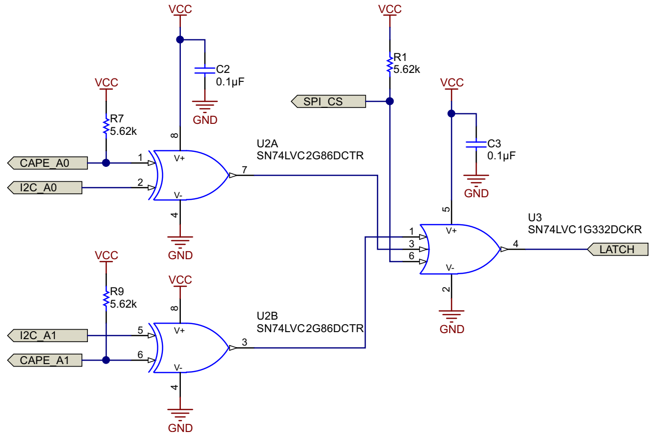


Figure 14. Address Decoder Logic

The full BeagleBone Cape specification is available in the *BeagleBone Black System Reference Manual* [7].

6 Getting Started Firmware

6.1 Software Flow

A simple program for the MSP430G2553 device has been written to test the design. The program mainly consists of an emulated bidirectional 16-bit SPI as required by the TLC5928 device. All signals are generated using GPIOs to maintain maximum flexibility in pin assignment, timing, and portability. See the following Figure 15 for the software flow diagram.

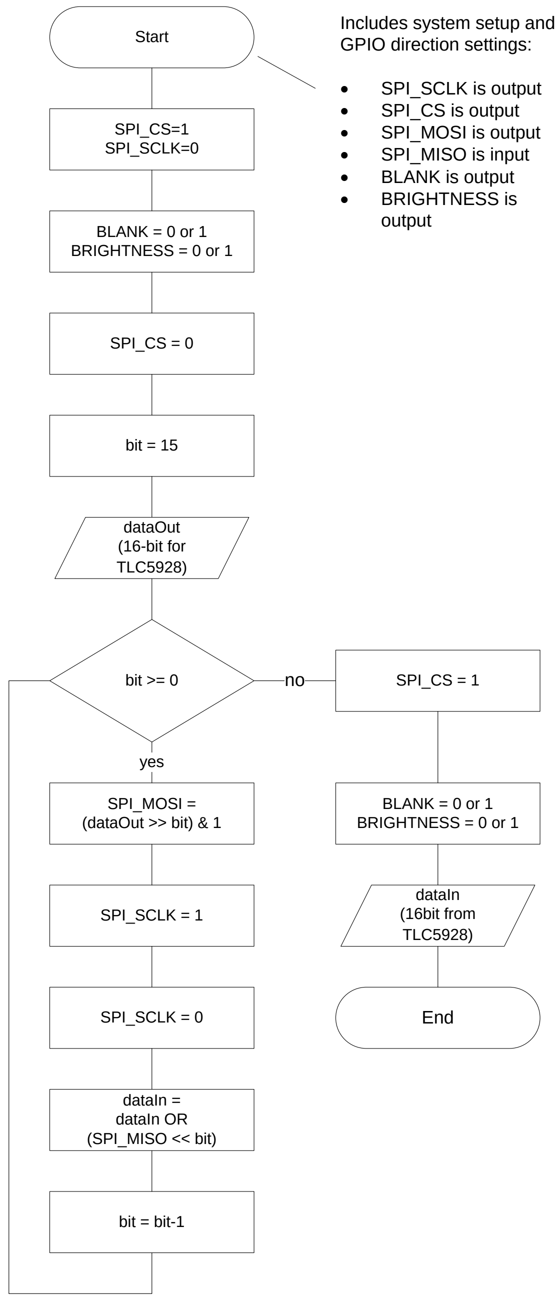


Figure 15. Software Flow (Test Program)

7 Test Setup

Figure 16 shows the test setup.

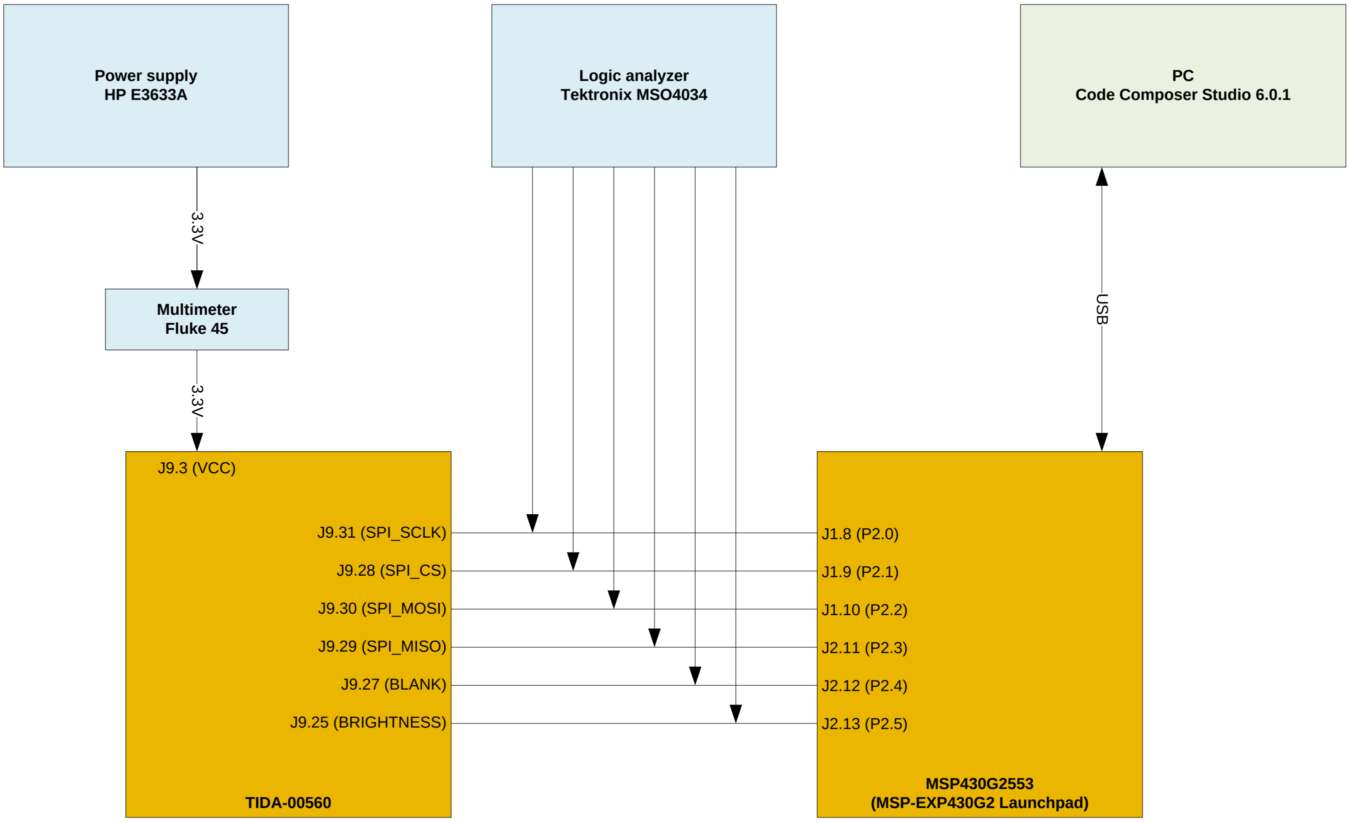


Figure 16. Test Setup

8 Test Data

8.1 Normal Operation

Figure 17 shows a standard data feed from the host to the TLC5928 device. This 16-bit standard SPI is supported by most processors. Data bits are sampled and driven at the SCLK rising edge. SCLK idles low if no transfer is performed. The data feed supports data rates up to 35 MHz.

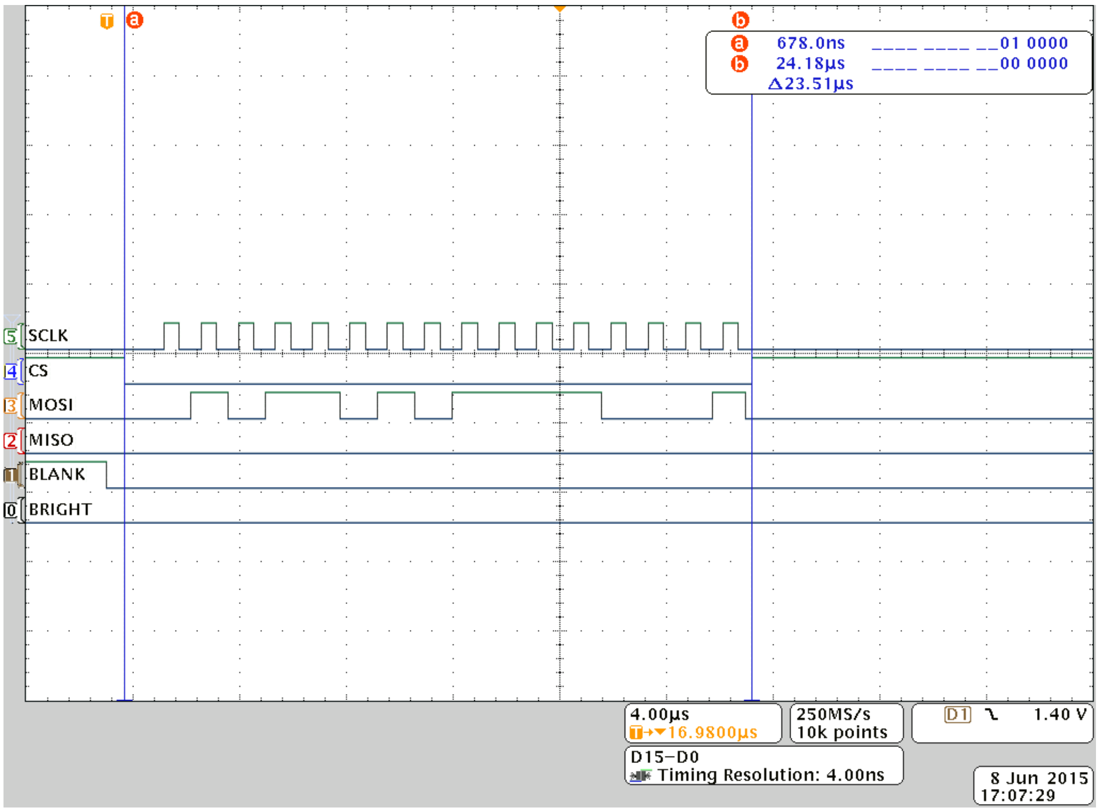


Figure 17. Normal Operation

8.2 LED Open and LED Short to GND

This section explains the *LED open* and *LED short to GND* features. To understand how these conditions were simulated, please refer to Section 5.1.4. Table 3 shows the setup of the test (the connection of the LEDs and whether the LED is driven). The LEDs D9 through D16 not mentioned in Table 3 are turned on.

Table 3. Fault Simulation—Conditions

DIODE	J1 JUMPER	CURRENT SINK	SIMULATED FAULT	RESULT	COMMENT
D1	J1A.1–J1B.1	On	None	No error	Normal operation
D2	J1A.2–J1B.2	On	None	No error	Normal operation
D3	J1A.3–J1B.3	Off	None	No error	Current sink off
D4	No jumper	Off	Open	No error	Current sink off
D5	J1B.5–J1C.5	Off	Short	No error	Current sink off
D6	J1A.6–J1B.6	On	None	No error	Normal operation
D7	No jumper	On	Open	Fault	Open detected
D8	J1B.7–J1C.7	On	Short	Fault	Short detected

As the TLC5928 data sheet describes, a fault is reported only if the following conditions are met: the current sink is enabled, BLANK is low, and a fault event is detected.

Figure 18 shows the jumper settings that Table 3 describes with the LEDs D3, D4, D5, D7, and D8 turned off.

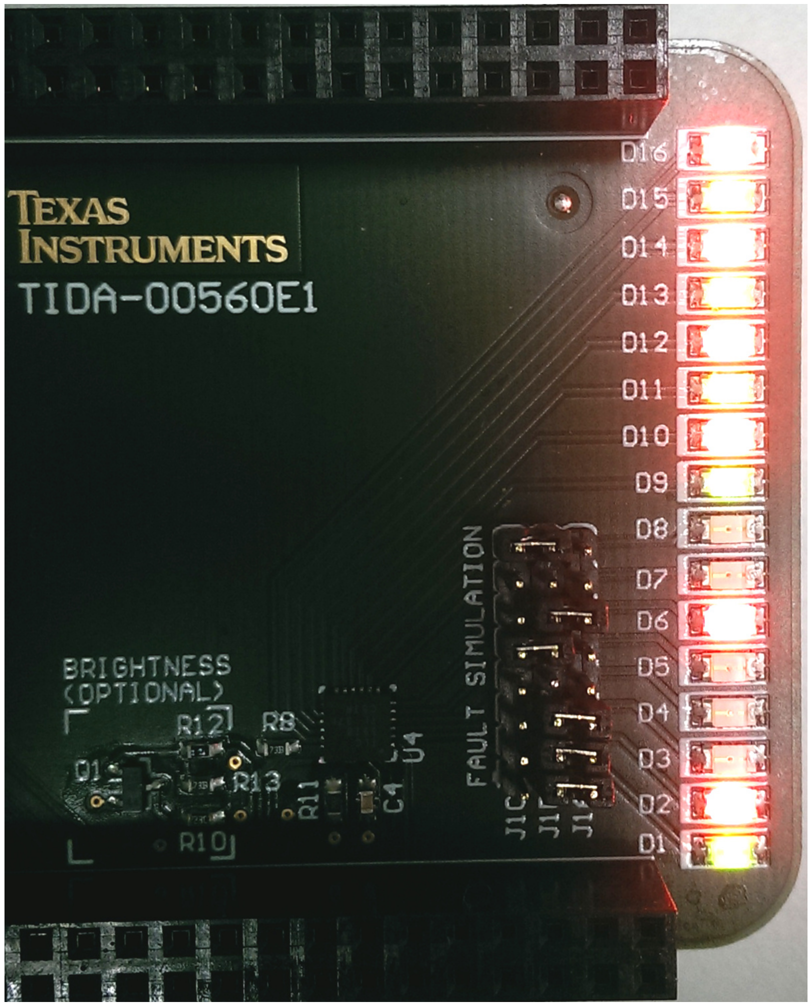


Figure 18. Fault Simulation Evidence

The MISO line reports the condition of each LED through the MOSI line while new data are clocked into TLC5928. Figure 19 describes a fault case using four consecutive transfers; Table 4 describes each transfer in detail.

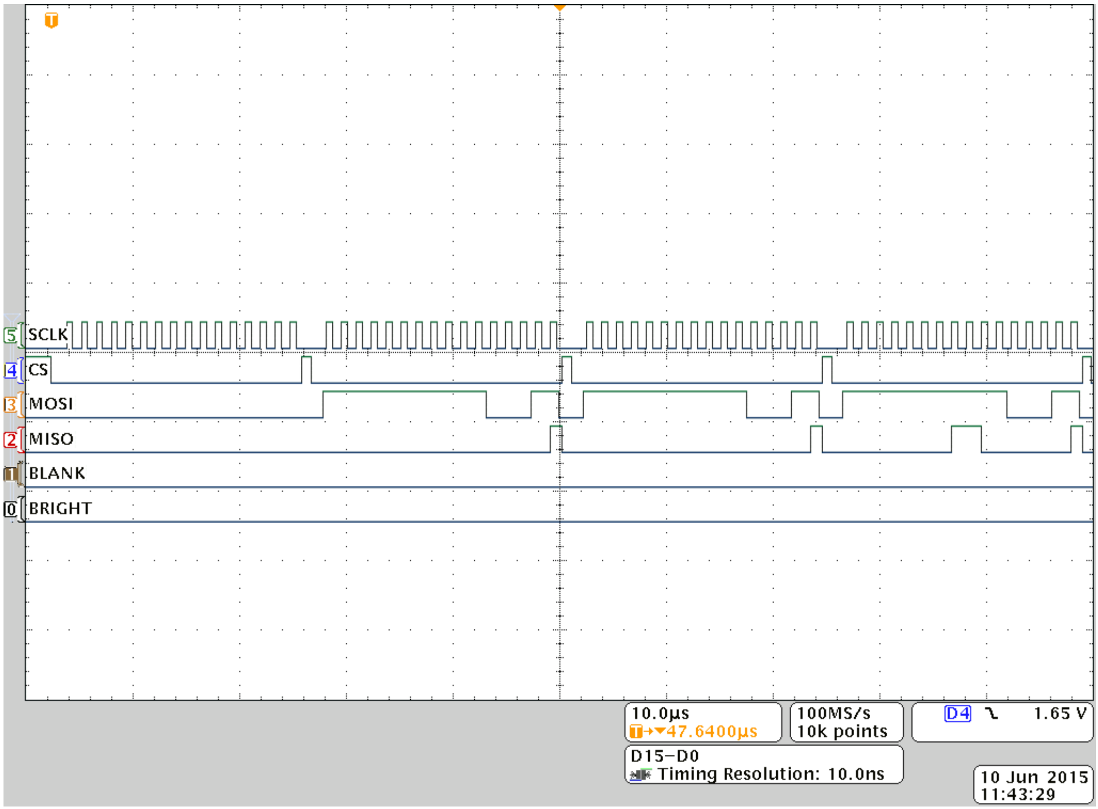


Figure 19. Fault Simulation—Data Transfer

As previously mentioned, the SCLK rising edge drives new data and latches data at the same time. With the last rising SCLK edge of a given transfer, that last status bit (from D1) latches in. After that last status bit is latched in, the TLC5928 device inserts the new value of D16 (latched-in with the first rising SCLK edge of the current transfer) to the MOSI line. This theory is proven by the fact that in transfer 1 this pulse is not present except on all consecutive transfers, where D16 is set high. This behavior has no impact to the host readings because the data changes after the last rising SCLK edge only.

Table 4. Fault Simulation—Data Transfer

TRANSFER	MOSI	MISO	FAULT SENSING
1	LEDs off	No fault reported	No fault
2	D1–D2 on D6–D16 on	No fault reported	No fault (LED data from MOSI updated at rising CS)
3	D1–D2 on D6–D16 on	No fault reported	LED faults detected (faults latched at rising CS)
4	D1–D2 on D6–D16 on	Fault reported	LED faults detected (latched faults transmitted)

Figure 20 shows a zoomed-in view of SCLK cycles 8, 9, and 10 of the 4th transfer to show the dependency of the SCLK and MISO edges. The user can observe that the TLC5928 changes to the next data bit value as soon as the hold time is elapsed after the SCLK rising edge, in that an error is reported on the 9th and 10th SCLK cycle. This series of events corresponds to TLC5928 pins OUT7 and OUT6, which are assigned to LEDs D8 and D7. According to Table 3 and Figure 18, the user can expect these pins to report a fault.

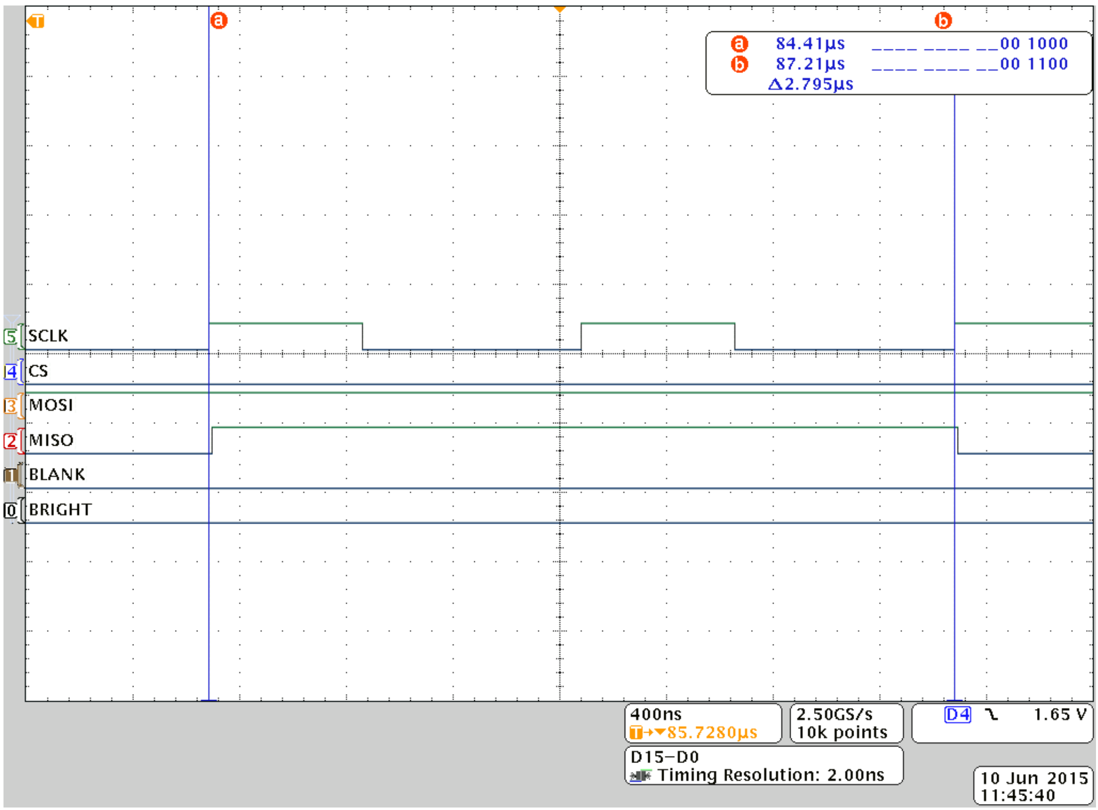


Figure 20. Fault Simulation (Zoomed-In SCLK Cycles 8th–10th)

8.3 Brightness

The user can control the brightness of the LEDs in three ways:

- Globally PWM the constant forward current
- Globally adjusting the constant forward current
- Individually toggling the constant forward current of certain LEDs on or off

8.3.1 Globally PWM the Constant Forward Current

By asserting the BLANK pin (high) of the TLC5928 device, the user can turn off all of the LEDs simultaneously without involving the serial port (to preserve the latched data). When the BLANK = low, all LEDs are controlled by the supplied data over the serial interface and the set forward current. The user can directly control the brightness of the LEDs by the high-to-low ratio by applying a PWM signal (see Figure 21).

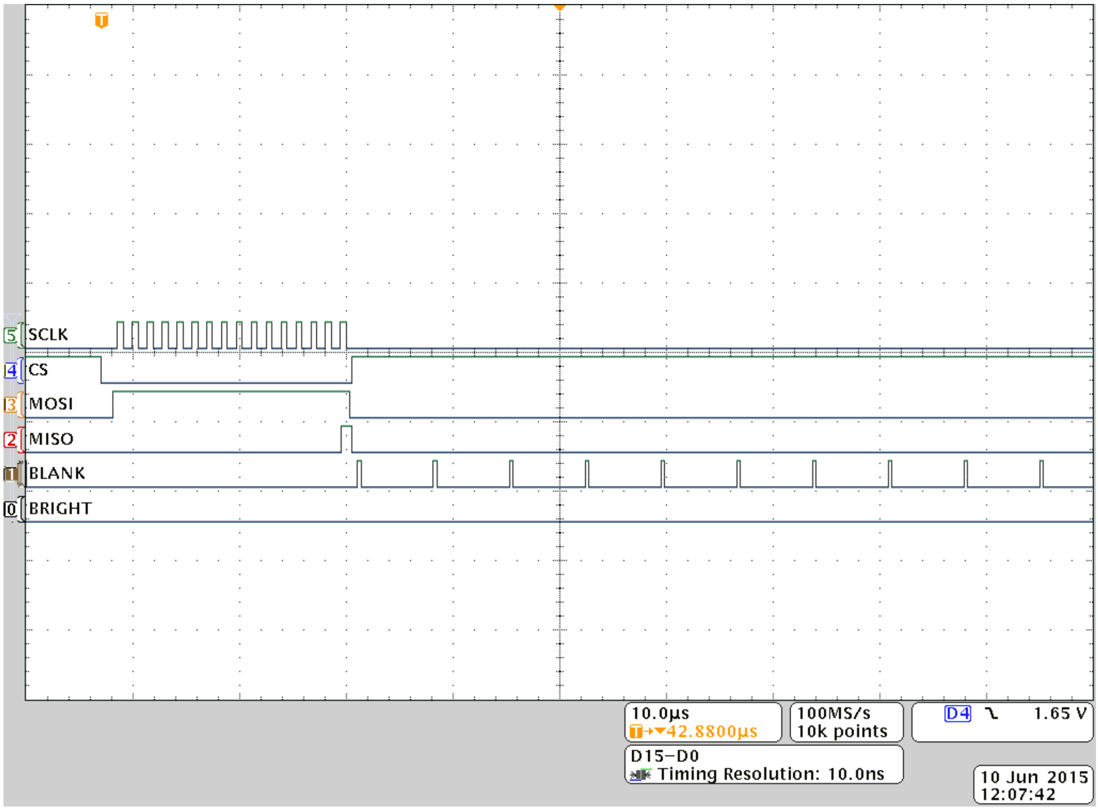


Figure 21. Brightness Control With Blank

8.3.2 Globally Adjusting the Constant Forward Current

Another option to control the brightness is to globally control the forward current of the LEDs. This global control is achievable by connecting a resistor from the TLC5928 pin 20 (IREF) to GND. The value of the resistor sets the current. This TIDA-00560 TI design supports a dynamic change of this resistor value by switching a second resistor in parallel, which in effect, supports two levels of brightness. See Section 5.1.3 for more information.

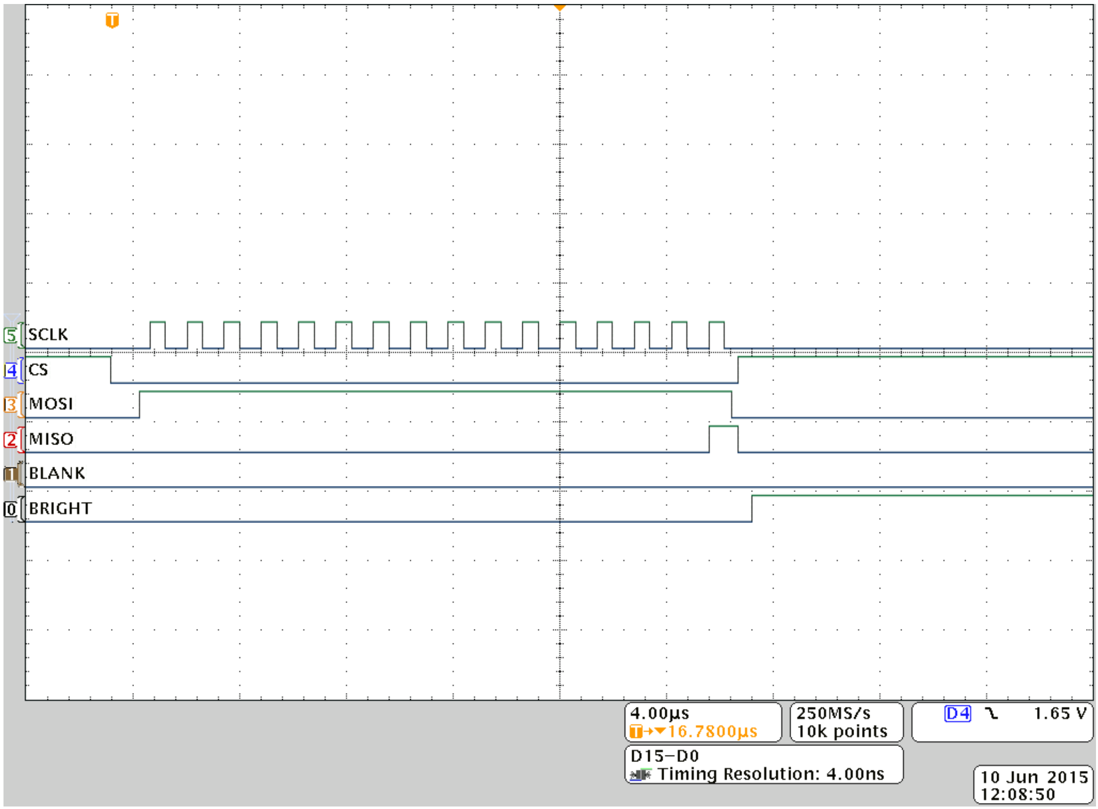


Figure 22. Brightness Control By Changing Forward Current

8.3.3 Individually Toggle Constant Forward Current of Certain LEDs On or Off

This TIDA-00560 TI design uses red- and green-colored LEDs. These colors are also widely used in PLC I/O modules. By nature, brightness with the same forward current differs so much that it is noticeable by the human eye. As an example, with I_F = 5 mA, the Wurth Electronic green LED supplies 10 mcd while the red LED supplies 50 mcd. To compensate for that mcd disparity, the user can set SPI data transfers to trigger with a fixed frequency that turns red LEDs on or off while constantly keeping the green LEDs on.

8.3.4 Current Consumption

The quiescent current for the board (all LEDs off) is expected to be fairly low. According to the data sheet, the TLC5928 device consumes typically 1 mA when the BLANK = high and R_{IREF} = 27 kΩ. The measured value with R_{IREF} = 25.5 kΩ is 0.535 mA. This measured value is a fairly good value given the fact that this number includes the (static) current of the BeagleBone Cape circuitry (EEPROM, logic).

Table 5 shows the current consumption of the entire board with all of the LEDs on with 2 mA or 5 mA for a different V_{CC}.

Table 5. Current Consumption

V_{CC} [V]	I_{LED} [mA]	I_{CC} [mA]	ΔI_{CC} [5 mA, 2 mA]
3 V	2 mA	31.69	–
3.3 V	2 mA	31.89	–
5 V	2 mA	33.12	–
5.5 V	2 mA	35.71	–
3 V	5 mA	78.37	46.68
3.3 V	5 mA	79.9	48.01
5 V	5 mA	81.67	48.55
5.5 V	5 mA	82.19	46.48

The expected current delta between 2 mA and 5 mA for the same V_{CC} is:

$$I_{DELTA\_CALC} = 16 \times 5\ \text{mA} - 16 \times 2\ \text{mA} = 48\ \text{mA} \quad (4)$$

The measured current delta for the four different V_{CC} is in the range of 46.48 mA to 48.55 mA, which is in the expected range.

The current measurement for a single LED (here D1 through OUT0) has been performed for 2 mA and 5 mA. Table 6 shows the results of these current measurements.

Table 6. Current Measurement D1

R_{IREF} [kΩ]	$I_{LED\_EXPECTED}$ [mA]	I_{LED_MEAS} [mA]	e [%]
25.5	1.976	1.938	1.9
10.16	4.959	4.924	0.7

According to the data sheet, the error tends to get larger at lower currents (maximum 1.5% for 2 mA). With the reference resistor value tolerance of 1%, the results are all in the range of expectation.

9 Design Files

9.1 Schematics

To download the schematic, see the design files at TIDA-00560.

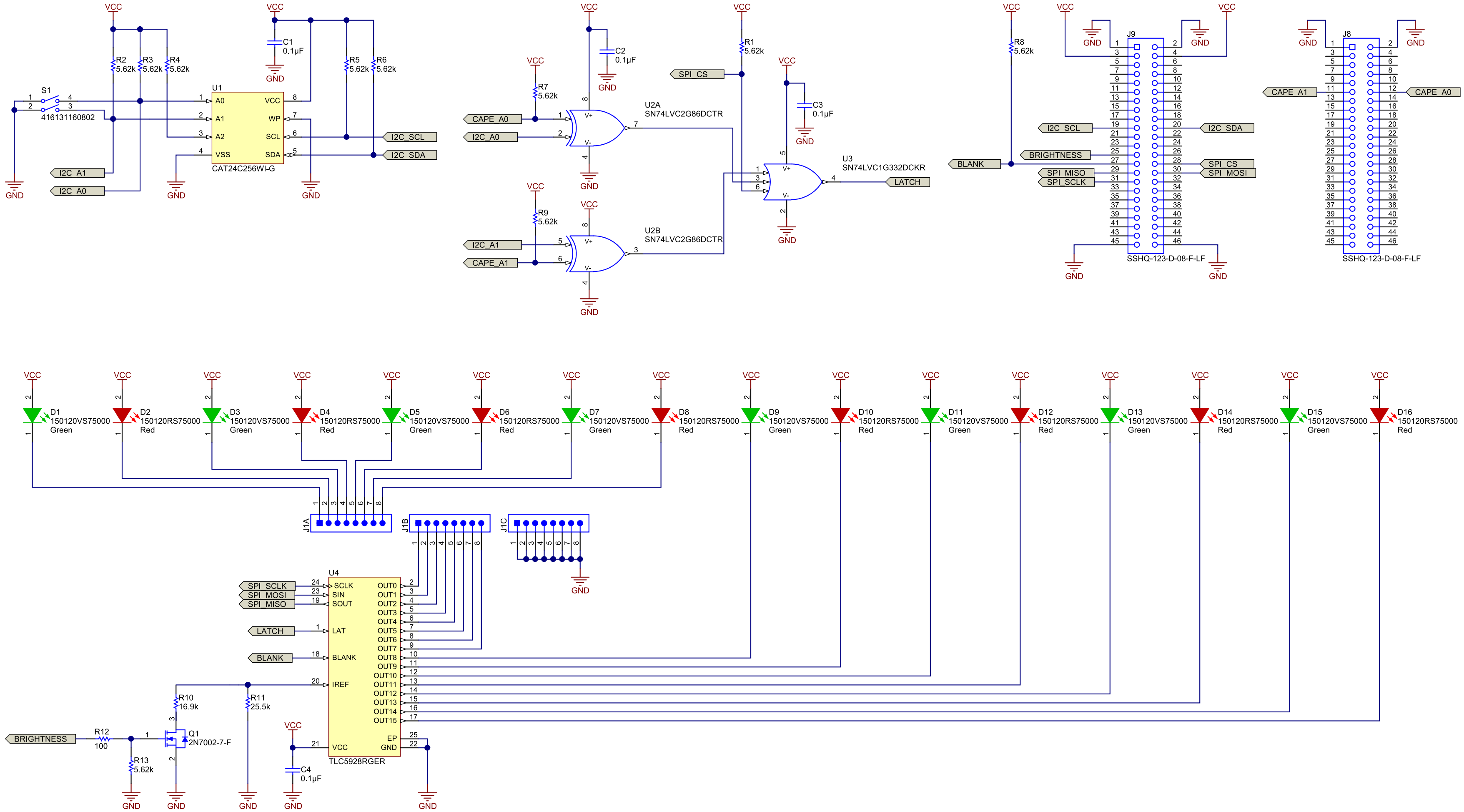


Figure 23. TIDA-00560 Schematic

9.2 Bill of Materials

To download the bill of materials (BOM), see the design files at TIDA-00560.

9.3 PCB Layout Recommendations

9.3.1 Layer Plots

To download the layout prints, see the design files at TIDA-00560.

9.4 Altium Project

To download the Altium project files, see the design files at TIDA-00560.

9.5 Gerber Files

To download the Gerber files, see the design files at TIDA-00560.

9.6 Assembly Drawings

To download the assembly drawings, see the design files at TIDA-00560.

10 Software Files

To download the software files, see the design files at TIDA-00560.

11 References

1. Texas Instruments, *16-Channel, Constant-Current LED Driver with LED Open Detection*, TLC5948 Data Sheet (SBVS120)
2. Texas Instruments, *MIXED SIGNAL MICROCONTROLLER*, MSP430G2553 Data Sheet (SLAS735)
3. Texas Instruments, *PCF8574 Remote 8-Bit I/O Expander for I^2C Bus*, PCF8574 Data Sheet (SCPS068)
4. Texas Instruments, *8-BIT SHIFT REGISTERS WITH 3-STATE OUTPUT REGISTERS*, SN74HC595 Data Sheet (SCLS041)
5. Texas Instruments, *HDC1000 Low Power, High Accuracy Digital Humidity Sensor with Temperature Sensor*, HDC1000 Data Sheet (SNAS643)
6. Texas Instruments, *AM335x Sitara™ Processors*, AM3359 Data Sheet (SPRS717)
7. GitHub.com, *BeagleBone Black System Reference Manual – Revision B*, BeagleBone Black System Reference Manual .pdf (http://bit.ly/1LQWUEI)

12 About the Author

LARS LOTZENBURGER is a Systems Engineer at Texas Instruments where he is responsible for developing reference design solutions for the industrial segment. Lars brings to this role his extensive experience in analog/digital circuit development, PCB design, and embedded programming. Lars earned his Diploma in Electrical Engineering from the University of Applied Science in Mittweida, Saxony, Germany.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265