

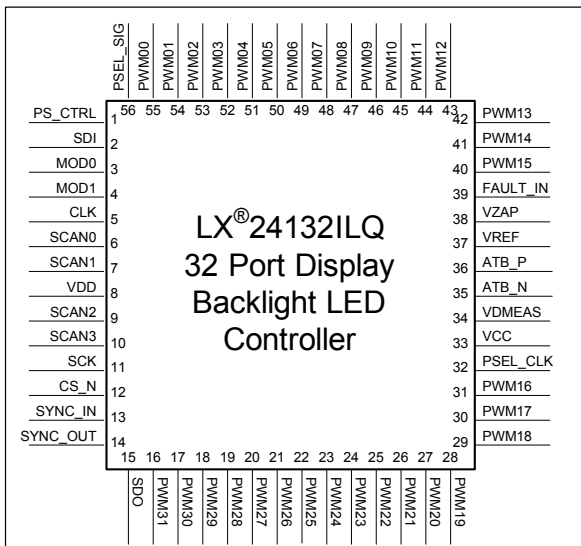


Introduction

LX24132ILQ - 32 Port Display Backlight LED Controller

DESCRIPTION	KEY FEATURES
<p>The LX[®]24132ILQ is a 32 Port Display Backlight LED Controller. The device is part of a chipset where each LX[®]24132ILQ is capable of controlling four LX[®]23108ILQ 8 Port LED Drivers. Each LX[®]23108ILQ 8 Port LED Driver contains eight FETs driving and controlling up to 4 x 8 LED channels, where each channel is capable of driving a current of up to 200 mA.</p> <p>The LX24132ILQ's internal power supply control circuitry adjusts the voltage level of an external LED power source. This is done by regulating the LED supply voltage to the optimum level, thus minimizing the system power loss. At the same time, accurate current regulation for each of the 32 LED strings is maintained.</p> <p>The LX[®]24132ILQ 32 Port Display Backlight LED Controller is configured through the SPI interface which speeds up communication and reduces the number of control signals between the LX[®]24132ILQ and the Host system (FPGA, Video Processor, CPU). The LX[®]24132ILQ supports a daisy chain connection in cases where more than 32 different LED Controllers (different PWM/Duty cycle) are required. PWM output signals can be synchronized with an external Video Processor V-Sync signal.</p> <p>Each one of the 32 PWM channels can be enabled/disabled by external one out of four SCAN0-3 input pins.</p> <p>The LX[®]24132ILQ 32 Port Display Backlight LED Controller can detect three types of system faults on each of its 32 channels (over-temperature, open LEDs and short LEDs). After detecting the faults the unit takes the required measures to protect the system.</p>	<ul style="list-style-type: none"> ◆ White or RGB LEDs Backlight controller for large size display panels ◆ Up to 32 LED strings with ±1.5% precision current matching ◆ Wide dimming ratio with PWM and LED current amplitude control ◆ 12-bit PWM duty-cycle resolution and 8-bit resolution for LED current setting ◆ LED power supply voltage control ◆ Automatic minimum FET Vd report ◆ SPI communication interface ◆ Open string, short LED and over-temperature protection for each individual channel ◆ On-chip thermal monitoring ◆ Synchronization to external input pulse (Sync-In) ◆ Enable/Disable PWM output by four input pins (SCAN0-3) ◆ Internal power supply trimming DAC
<p>IMPORTANT: For the most current data, consult Microsemi's website: http://www.microsemi.com</p>	

Pin Configuration



PACKAGE ORDER INFO	
T_A (°C)	Plastic MLPQ 8 x 8 mm QFN 56 pin
	RoHS Compliant/Pb free
-40° to +85° C	LX24132ILQ
<p>Note: Available in Tape & Reel. Add the letters "TR" to the part number. (LX24132ILQ-TR)</p>	

LX24132ILQ - 32 Port Display Backlight LED Controller

Typical Application

- LCD Display Back-lighting
- LED Signage
- LED Displays

Figure 1 illustrates a typical application where a System Controller communicates with a LX[®]24132ILQ 32 Port Display Backlight LED Controller through an SPI bus. The LX[®]24132ILQ is connected to four LX[®]23108LILQ arrays supporting a total of 32 LED strings. Each LX[®]23108LILQ transmits current very accurately and can drive eight LED strings. The output currents' value is set by the LX[®]24132ILQ which produces a precise and stable voltage V_{REF}.

All four LX[®]23108LILQs are cascaded, allowing the LX[®]24132ILQ LED Controller to sequentially select all 32 channels for monitoring, search for V_{DMIN} and optimally adjust the power supply voltage through PDM control. Channel dimming is individually adjusted by 32 PWM output pins; 8 inputs per single LX[®]23108LILQ. Currents are scaled by the external sense resistors.

Figure 2 illustrates multiple connections of several LX[®]24132ILQs supporting 32x4 LED strings.

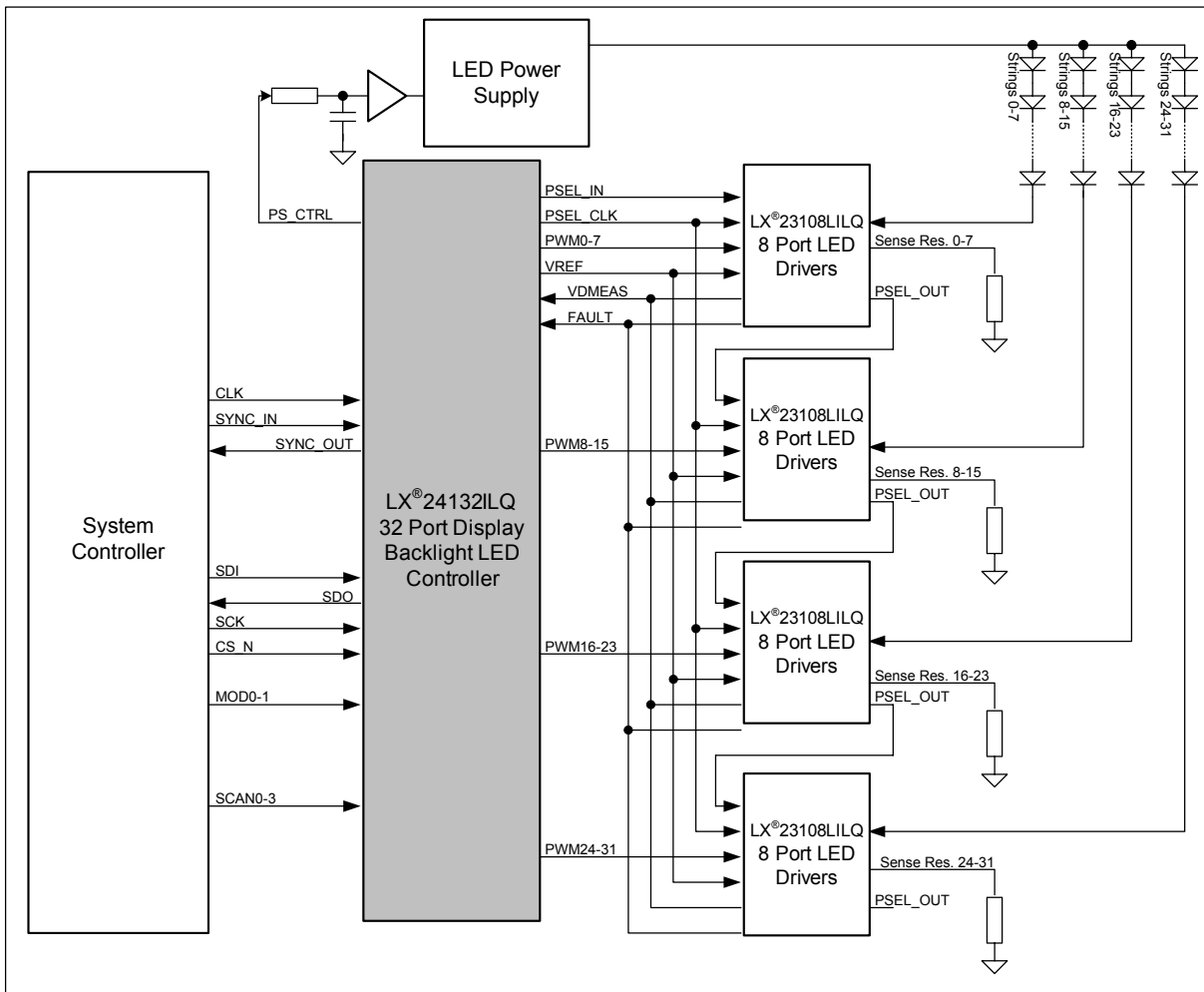


Figure 1: Low Voltage LX[®]24132ILQ and LX[®]23108LILQ - Typical Application

LX24132ILQ - 32 Port Display Backlight LED Controller

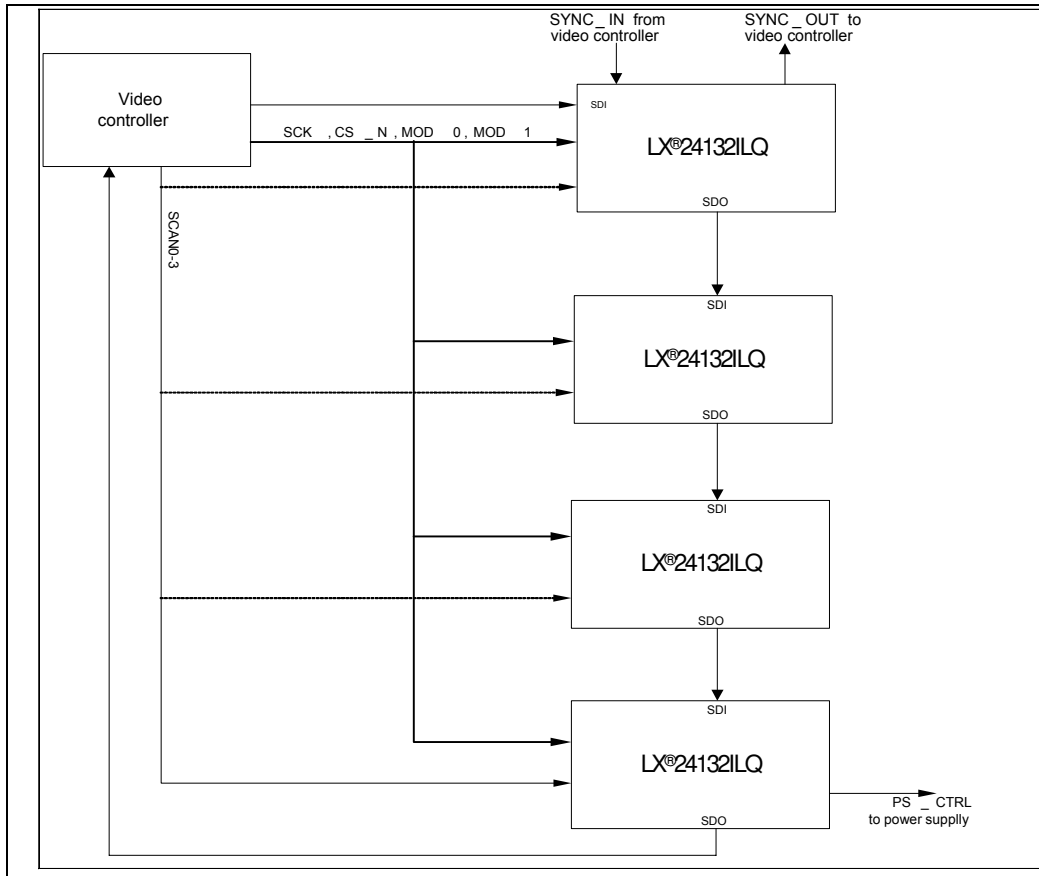


Figure 2: Typical Application with Four LED Controllers



LX24132ILQ - 32 Port Display Backlight LED Controller

APPLICABLE DOCUMENTS

- LX[®]23108ILQ 8 Port LED Drivers Datasheet, Catalogue Number 06-0074-058
- LX[®]24132ILQ /LX[®]23108ILQ AN-182, Designing a Low Current LED BackLight Driver System, Catalogue Number 06-0077-080

ABSOLUTE MAXIMUM RATINGS

Supply Input Voltage (VCC, VDD)	-0.5 to 4.5 VDC
All Other Pins	-0.5 VDC to VCC+0.3 up to 4.5 VDC
Operating Ambient Temperature Range	-40° to +85° C
Maximum Operating Junction Temperature	150° C
ESD Protection at All I/O pins	± 2 KV HBM
Storage Temperature Range	-65° to +150° C
Package Peak Temperature for Solder Reflow (40 seconds maximum exposure)	+260° C (+0, -5° C)

Notes: Exceeding these ratings could result in damage to the device. All voltages are with respect to Ground.

THERMAL DATA (POWER CONSUMPTION)

21°C/W, according to JESD51-7.
thermal resistance-junction to ambient
Junction Temperature Calculation: $T_J = T_A + (P_D \times \theta_{jA})$.



LX24132ILQ - 32 Port Display Backlight LED Controller

Electrical Characteristics

Unless otherwise specified, the following specifications apply to the operating ambient temperature and the following test conditions: $V_{DD} = V_{CC} = 3.3$ VDC. Performance is guaranteed for $0^{\circ}C \leq T_J \leq +110^{\circ}C$.

PARAMETER	SYMBOL	TEST CONDITIONS / COMMENT	LX [®] 24132ILQ LED CONTROLLER			UNITS
			MIN.	TYP.	MAX.	
POWER SUPPLY						
Input voltage	V_{DD}, V_{CC}	*Note 1	3.0	3.3	3.6	VDC
Operating current	I_{DD}	From VDD power supply		3	5	mA
	I_{CC}	From VCC power supply		3.5	5	mA
DIGITAL INTERFACE						
Input logic high threshold	V_{IH}		2.2			VDC
Input logic low threshold	V_{IL}				0.8	VDC
Output high voltage	V_{OH}	$I_{OH} = -1$ mA	2.4			VDC
Output low voltage	V_{OL}	$I_{OH} = 1$ mA			0.4	VDC
SCK clock frequency	Fsck				32	MHz
SDI set up time to SCK	Tsu_sdi		5			nS
SDI hold time to SCK	Thld_sdi		5			nS
SDO prop delay from SCK	Tprop_sdo	20 pF capacitive load			12	nS
CS_N set up time to SCK	Tsu_cs		5			nS
CS_N hold time to SCK	Thld_cs		5			nS
Time between adjacent SPI transactions	Tadj_cs			37		Clock s
Time from deassertion of CS_N to 1st SCK rising edge	Tdel_sck		4			Clock s
MOD[1:0] set up time to CS	Tsu_mod		0			nS
MOD[1:0] hold time to CS	Thld_mod		0			nS
CLK frequency	Fclk	Duty cycle 40% - 60%	4		8.192	MHz
SYNC_IN pulse width	sync_in_pulse		3			Clock s
SYNC_OUT pulse width	sync_out_pulse	$T_{SYNC_OUT [CLK]} = 320 * (PWM_FRDIV + 1)$		320		Clock s
LED CURRENT CONTROL						
DAC resolution	Vref_err			8		bits
V_{REF} output accuracy error		1.25 VDC < $V_{REF} \leq 2.5$ VDC			± 1.43	%
		0.5 VDC < $V_{REF} < 1.25$ VDC			± 2	%
	Vref	0.234 VDC < $V_{REF} < 0.5$ VDC			± 5.26	%
V_{REF} output range				2.5		VDC
Note 1: During power up VDD should not precede VCC.						



LX24132ILQ - 32 Port Display Backlight LED Controller

PARAMETER	SYMBOL	TEST CONDITIONS / COMMENT	LX [®] 24132ILQ LED CONTROLLER			UNITS
			MIN.	TYP.	MAX.	
POR CELL						
POR threshold		High threshold	2.5	2.7	2.9	VDC
POR hysteresis			0.2	0.25	0.3	VDC
POR time delay		Power on reset activation pulse to all blocks.		0.6	1.2	mS
ADC						
Resolution				7		bits
LSB				15.6		mV
Offset error				1		LSB
Reading error		1 VDC ≤ Drain Voltage ≤ 2 VDC			± 3	LSB
Full scale ADC input				2		V
PWS CONTROL						
Duty-cycle resolution		At all system clock frequencies (fPWM)		12		bits
PWM frequency range		$f_{PWM}[kHz] = \frac{f_{CLK}[kHz]}{(4096 \times (PWM_FR DIV + 1))}$ Refer to Table 1 for more details.	50		2K	Hz
SYSTEM PARAMETERS - PROTECTIONS						
Open LED detection		Open LED detection is digitally controlled and configurable by user (See <i>Application Note AN-182, Catalogue Number 06-0077-080</i> . Per Channel detection is traced when LED voltage is lower than the set "Open Drain Voltage Threshold".				
Short LED detection		At T _{AMB} = 25° C and V _{CC} = 5 VDC Short LED detection for the LX [®] 23108LILQ FET Array is digitally controlled. Per channel detection is traced when LED voltage is higher than the "Short Drain Voltage Threshold".	8	8.8	10	VDC
Over-temperature detection		The LX [®] 24132ILQ LED Controller has a reading mechanism which receives per channel over-temperature indications from the LX [®] 23108LILQ LED Driver.	150	180	210	° C

www.Microsemi.com

LX24132ILQ



LX24132ILQ - 32 Port Display Backlight LED Controller

Functional Pin Description

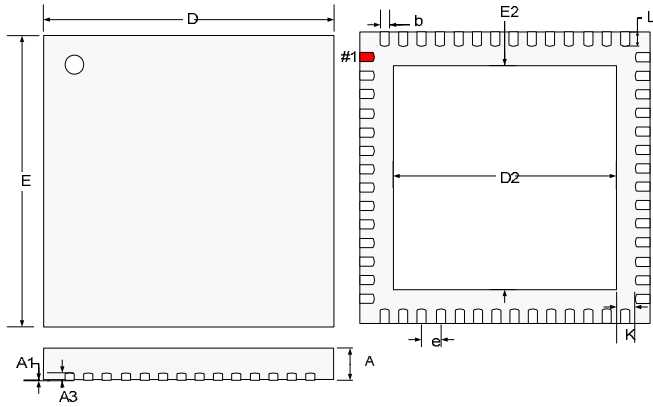
NAME	PIN #	DESCRIPTION
AGND	EPAD	Exposed PAD - Analog ground. A proper ground plane should be deployed around this pin wherever possible.
V _{CC}	33	Supply voltage for the internal analog circuit. A low ESR bypass capacitor (not less than 1 uF) should be placed as close as possible to this pin, using low impedance traces to AGND.
ATB_N ATB_P	35, 36	Production test output pins; do not connect.
SDI	2	Serial data input pin. The data from the SPI Host to this pin is clocked into the input shift register at the rising edge of the SCK clock. The MSB is inputted first.
SDO	15	Serial data output pin. The data from the input shift register is shifted out from this pin at the falling edge of the SCK clock. The MSB is outputted first. The SDO pin can be connected to the SDI pin of another device to form a cascaded SPI chain. Data from the LED controller can also be transmitted to the System Controller from this pin. If not used, leave this pin disconnected.
SCK	11	Clock signal for the SPI operation. A single clock shifts the data by one bit. It is only active when shifting data. This happens when CS_N is at 'low' level.
CS_N	12	Chip Select - SPI control signal input. The data shift starts when the CS falls from 'high' to 'low' and the data shifted into the input register is latched into the buffer registers at the rising edge of this signal.
SYNC_IN	13	Synchronization input. A new PWM cycle is started when a rising edge is received from this pin, regardless of the on-going PWM cycle's status. The PWM counter is reset to zero and the PWM comparator is reloaded if data is available. If not used, connect this pin to ground. Note: Minimum SYNC_IN positive pulse width should be less than two CLK pulses width.
SYNC_OUT	14	Synchronization output utilized for multi-chip applications. It synchronizes several LX [®] 24132ILQ 32 Port Display Backlight LED Controller devices. It can also be utilized to synchronize the LCD frame timing through the Video Controller.
CLK	5	System clock - PWM control clock. This signal clocks the counting of the internal PWM counter and is also used for internal logic operations.
V _{DD}	8	Connects to the Core Logic and I/O supply rail. A ceramic 1 uF decoupling capacitor or greater should be connected from this pin to the DGND.
PWM0-31	55 to 40 31 to 16	PWM Gate control output signals used to command up to 4 x 8-channel LX23018LILQ LED Driver ICs
VDMEAS	34	MOSFET drain voltage sensing input. The external MOSFET drain voltage is sensed via this pin and used to control the external power supply, maintaining the optimum voltage for the LED strings.
FAULT_IN	39	Fault input signal coming from the LED Driver IC. It is asserted 'low' when fault event is detected at one of the LED strings.
PS_CTRL	1	Power supply control signal output. This Pulse Density Modulation signal (PDM) is used to interface to the power supply, adjusting the DC voltage of the LED strings to the optimum level.
MOD0	3	An input signal; used for production only. Should be connected to GND
MOD1	4	An input signal; selects between Configuration (VDD) mode and Operation (GND) mode.
SCAN0-3	6, 7, 9, 10	Input logic signals utilized for scanning zone selection.
VZAP	38	Zapping input for IC production trimming; must be tied to VCC.
VREF	37	Analog reference output signal used for the 8-channel LED Drivers.
PSEL_SIG	56	Channel selection serial output signal to LX [®] 23108LILQ which is a 32 bits packet, used to select (for monitoring) one of the 32 strings drain voltages and thermal sensors signals.
PSEL_CLK	32	Channel selection serial clock signal. Each single clock shifts the internal register data of the LX [®] 23108LILQ by one bit, allowing the monitoring of the next channel.



LX24132ILQ - 32 Port Display Backlight LED Controller

Package Information

LQ 56-Pin 8x8mm QFN



DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	0.80	1.00	0.031	0.039
A1	0.00	0.05	0	0.002
A3	0.20 REF		0.008 REF	
K	0.20 MIN		0.008 MIN	
e	0.50 BSC		0.02 BSC	
L	0.30	0.50	0.012	0.02
b	0.18	0.30	0.007	0.012
D2	6.00	6.25	0.236	0.246
E2	6.00	6.25	0.236	0.246
D	8.00 BSC		0.315 BSC	
E	8.00 BSC		0.315 BSC	

Note:

1. Dimensions do not include protrusions; these shall not exceed 0.155mm (.006") on any side. Lead dimension shall not include solder coverage.

LX24132ILQ - 32 Port Display Backlight LED Controller

LX24132ILQ Principle of Operation

The LX[®]24132ILQ 32 Port Display Backlight LED Controller is designed to drive up to 32 LED strings and control external power supplies to regulate the LED current. The interface with the Hosting system (Video Processor / FPGA / CPU) is accomplished via a standard SPI bus. LED current and PWM dimming duty-cycle commands are received from the Hosting system in a digital format and executed by the internal circuitry to obtain the desired BackLight control. In addition, the device provides a power supply control signal (PS_CTRL) used to control the external power supply and trim it to the optimum level. This minimizes system power dissipation, while maintaining accurate current regulation for each of the 32 LED strings.

Block Diagram

Figure 3 shows the LX[®]24132ILQ LED 32 Port Display Backlight LED Controller block diagram, describing its main functions. The SPI block implements the communication with the external Controller. The power supply control circuitry adjusts the LED strings' main voltage by utilizing the PDM method (Pulse Density Modulation). The scanning control circuitry includes the required logic needed to perform the zone selection.

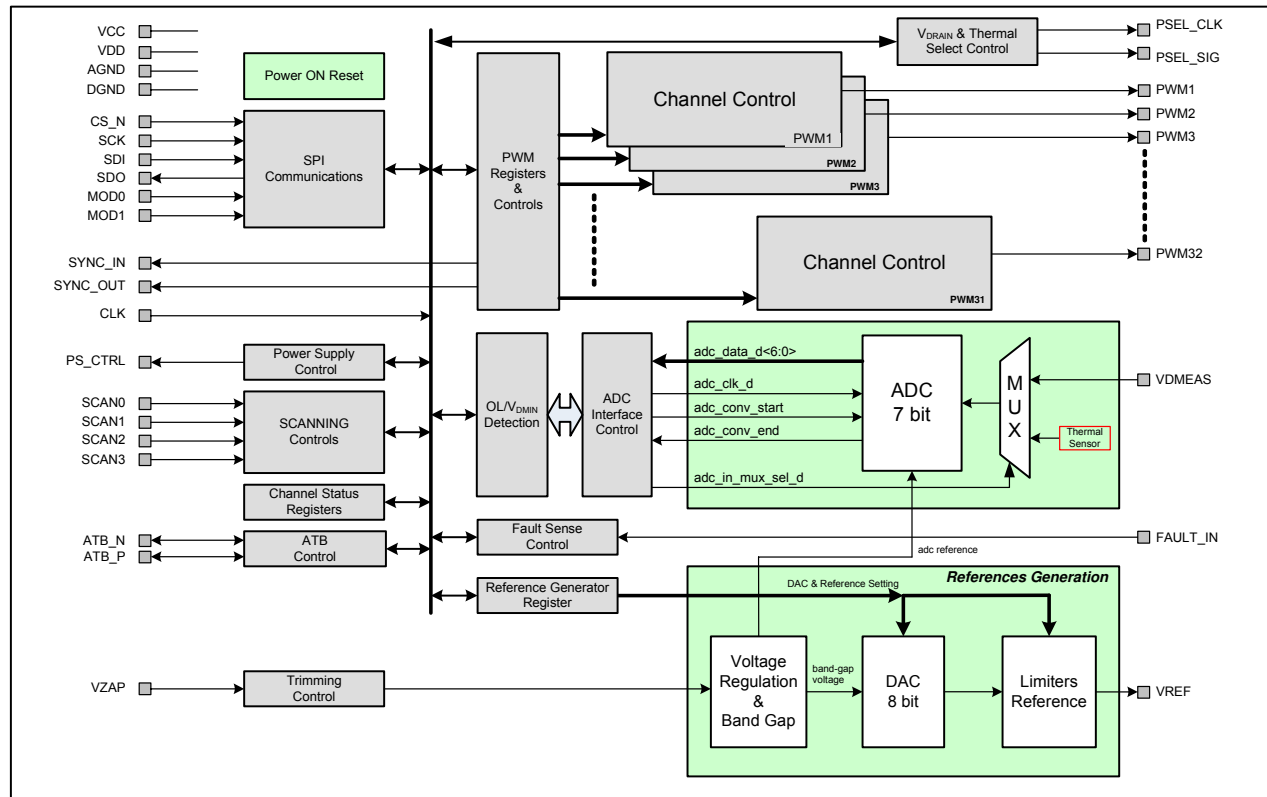


Figure 3: LX[®]24132ILQ: Detailed Block Diagram



LX24132ILQ - 32 Port Display Backlight LED Controller

The Channel Status Registers are used to enable/disable channels and indicate their status (thermal event, under-load, etc.). The ATB Control block is used for testing purposes. The PWM registers and control signals establish the PWM parameters common to all channels. Each Channel Control block generates PWM command signals individually. The OL/VDMIN Detection block controls the search for open strings and a valid VDMIN. An ADC Interface Control block selects (for monitoring) the required channel voltages and the internal thermal sensor. The Fault Sense Control circuitry detects thermal and load short-circuit events at the associated LX[®]23108LILQ devices.

The Reference Generator register in conjunction with the Reference Generation block produce an accurate output voltage used to set the string currents. The VDRAIN & Thermal Select Control instruct the LX[®]23108LILQ as which channel is to be monitored. The Trimming Control is used for production purposes.

Slave SPI Interface

The SPI interface consists of four signal lines: SDI, SDO, SCK, and CS_N (see Figure 4).

- **SDI**: Serial data input to the LX[®]24132ILQ
- **SDO**: Serial data output from the LX[®]24132ILQ
- **SCK**: Serial data clock input
- **CS_N**: Serial data chip select

The SDI signal is sampled at the rising edge of the SCK, while the SDO is driven at the falling edge of the SCK, where the MSB is shifted out first.

SPI data transmission/reception should be executed, starting from bit 415 to bit 0 (Operation mode), or starting from bit 223 to bit 0 (Configuration mode) Figure 5 shows the SPI timing diagram for a 4-driver configuration. The SPI data of the fourth LX[®]24132ILQ is transmitted first, following the SPI data of the third, second and then the first LX[®]24132ILQ.

SPI Noise Immunity

The LX[®]24132ILQ SPI interface includes special noise immunity protection mechanisms, which improve the electrical SPI interface immunity against high frequencies noise, short spikes and

glitches in a noisy PCB environment. These mechanisms include:

- Filter short glitches and spikes of SPI CS_N signal
- If SPI CS_N chip select signal is Set Low for less than 2 x System Clocks, this pulse is ignored, the transaction is canceled and the internal registers remains unchanged.

Ignore the SPI transaction if the number of SPI SCK clocks is not module 8 (8, 16, 24, 32, etc.); in this case the internal registers remains unchanged.

SPI Timing

Figure 4 describes LX[®]24132ILQ SPI timing (refer to Electrical Characteristics for timing values).

In addition to various SPI timing values, the first SCK SPI rising edge should occur Tdel_sck (four system clocks) after CS_N SPI chip select negation. Minimum time between the end of SPI transaction to the start of the next SPI transaction should be Tadj_cs (37 system clocks). Refer to Electrical Characteristics, page 5 for more details.

LX24132ILQ - 32 Port Display Backlight LED Controller

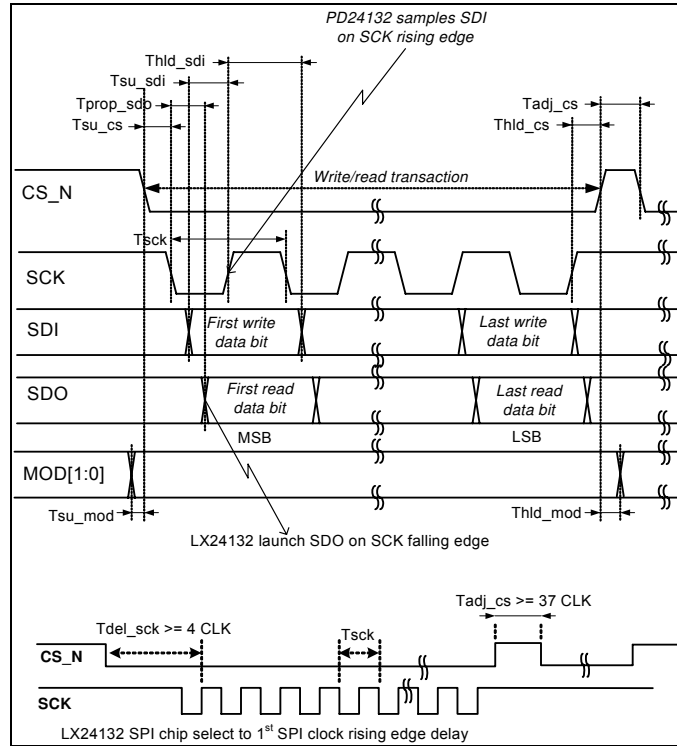


Figure 4: SPI Signals Diagrams

LX24132ILQ - 32 Port Display Backlight LED Controller

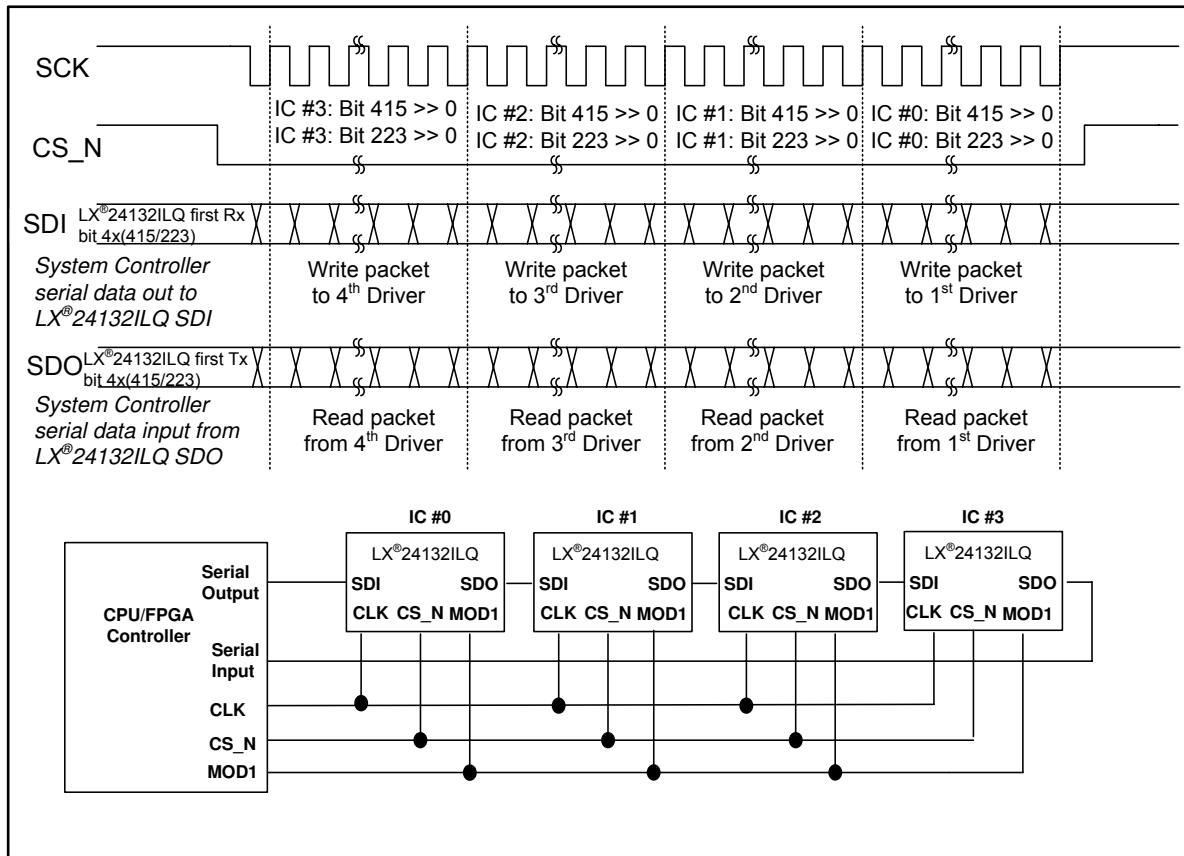


Figure 5: SPI Signals Diagram for a Four LED Controllers Configuration

Communication Modes

Chip configuration and monitoring is accomplished by sending and receiving a series of bits over the SPI.

There are two communication modes:

- Operation mode (MOD1 input pin = Low)
- Configuration mode (MOD1 input pin = High)

COMMUNICATION MODE	MOD1 INPUT PIN
Operation Mode	GND
Configuration Mode	VDD

Operation mode: Bit pattern involves reading and writing registers accessed periodically, providing LED PWM pulse width and more.

Configuration mode: Bit pattern involves reading and writing registers that are usually configured only once, for example PWM_FRDIV (crystal frequency divider) register etc.

Figure 6 and Figure 7 illustrate the read/write registers involved in each SPI bit stream for operation and configuration mode.

For multiple chip applications, the SPI communication should be cascaded (see Figure 5).



LX24132ILQ - 32 Port Display Backlight LED Controller

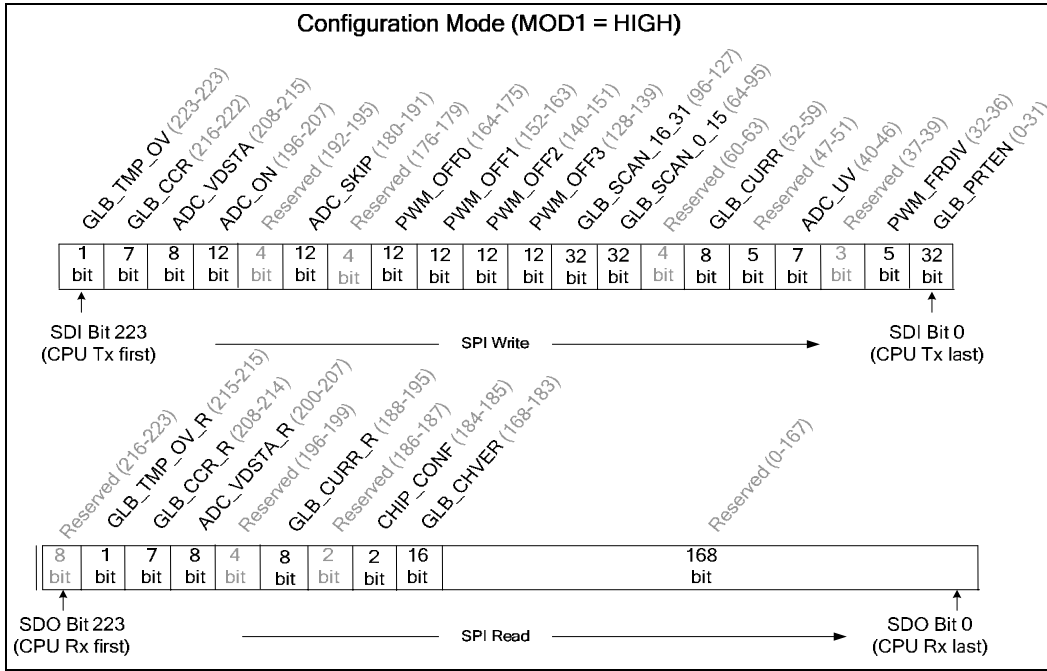


Figure 6: Configuration Mode SPI Bit Stream

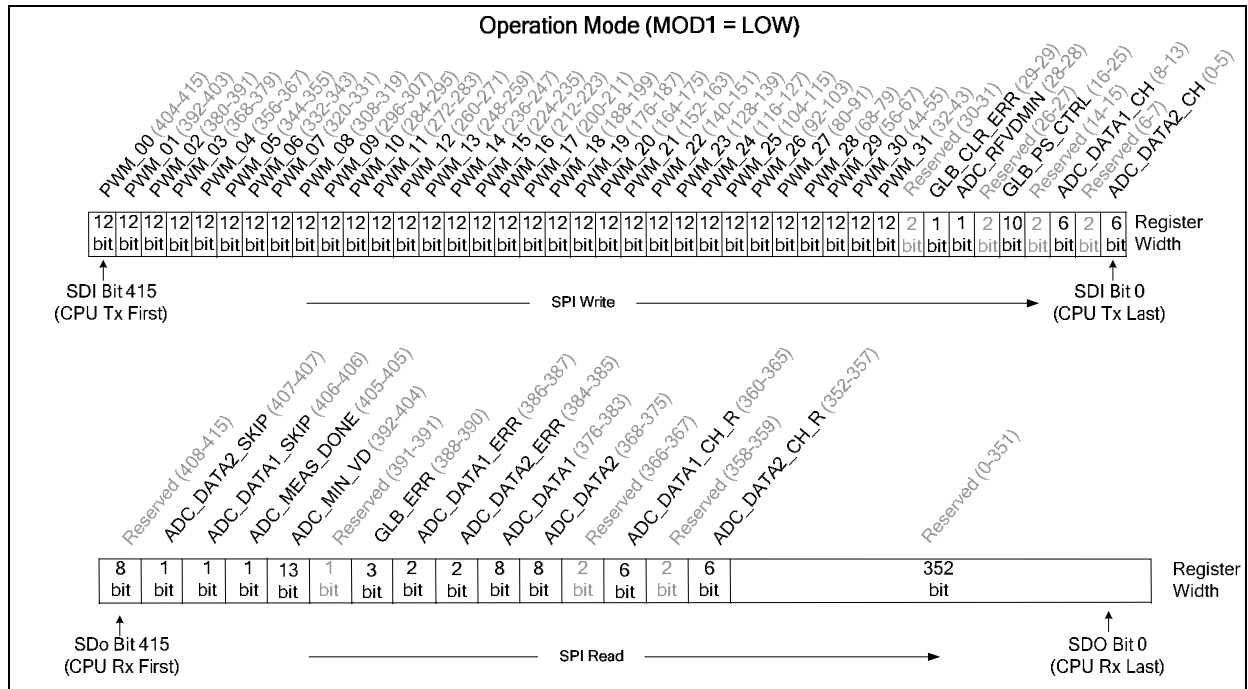


Figure 7: Operation Mode SPI bit Stream

PWM Control Input Pin: SYNC_IN

The SYNC_IN input signal determines if the PWM output will be synchronized with the SYNC_IN input clock.

Clearing GLB_CCR register/ bit1 resets the internal 12 bit PWM counter (clocked via the CLK pin divided by PWM_FRDIV+1 register) to zero on each SYNC_IN rising clock.

LX24132ILQ - 32 Port Display Backlight LED Controller

The counter continues to increment up to '4095' and restarts from '0' again, unless another SYNC_IN pulse resets the counter before reaching '4095'.

PWM Control Input Pins: SCAN0-3

SCAN0-3 input signals (Figure 8), together with the GLB_CCR bit0 register, control the PWM output pins.

SCAN0-3 input pins can be used to enable/disable PWM output signals.

Lowering the SCAN0-3 input enforces PWM[N] output pins to 'Low' (no current through the LEDs). Restoring the SCAN0-3 input to 'High' re-enables the PWM[N] output generation.

Note that only PWM channels linked by the GLB_SCAN0_15 and the GLB_SCAN16_31 to this specific SCAN[N] input pin will be affected.

The GLB_CCR bit0 determines the way that the SCAN[N] input affects the PWM[N] outputs. Setting Bit0 to '0' synchronizes the SCAN[N] input with the internal 12 bit PWM counter, preventing partial PWM output pulses.

However, setting Bit0 to '1' causes SCAN[N] input to perform binary AND function with the PWM[N] output signal, which consequently may generate partial PWM output pulses (see Figure 8).

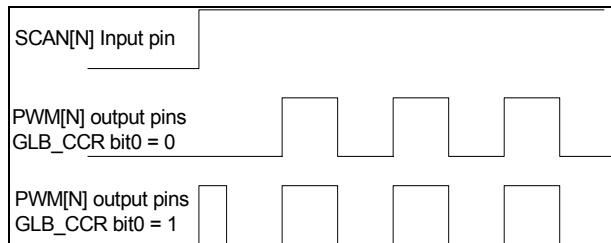


Figure 8: SCAN Input Operation

PWM Output Pins PWM0-PWM31

PWM output generation (see Figure 9) is based on:

- Internal 12-bit PWM counter clocked via the CLK pin divided by PWM_FRDIV+1 register,
- PWM_00-PWM_31 registers controlling the duty cycle of each PWM[N] channel
- Four PWM_OFF0 – PWM_OFF3 registers controlling the PWM off timing (stop LED current).

12 bit internal PWM counter counts from 0-4095, repeatedly, unless SYNC_IN input pulse resets the counter to zero on each SYNC_IN rising edge.

PWM_[N] registers modification is performed by sending new SPI Operation mode bit pattern with various PWM values. Since PWM_[N] is double buffered, new SPI Operation mode PWM_[N] values

are written to the first buffer on the SPI CS rising edge and are loaded into the second buffer on the PWM_OFF timing of each string. The new PWM value is effective on the next PWM cycle.

GLB_SCAN_0_15 and GLB_SCAN_16_31 registers link each PWM[N] channel (two bit per PWM channel) to one out of four PWM_OFF[N] registers. Whenever an internal 12 bit PWM counter value matches any of the PWM_OFF0, PWM_OFF1, PWM_OFF2 or PWM_OFF3 registers, all the PWM[N] channels which were linked by the GLB_SCAN_0_15 and GLB_SCAN_16_31 registers to this specific PWM_OFF[N] register are turned off (stop LED current).

PWM00_PWM31 registers control PWM[N] channel duty cycle. Figure 9 shows an example where PWM0 output pin is configured to use PWM_OFF0 (GLB_SCAN_0_15 bit 0&1 = 0).

The PWM_OFF0 register is set to 3000 and PWM_00 register duty cycle is set to 500. Whenever the 12 bit internal counter reaches 2500 (3000 - 500), PWM0 output pin toggles to High and turns to Low whenever the 12 bit internal PWM counter reaches 500 ticks higher (3000).

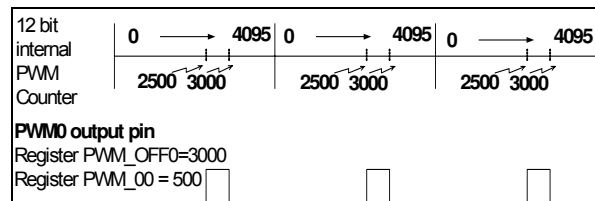


Figure 9: PWM operation

Power Supply Control

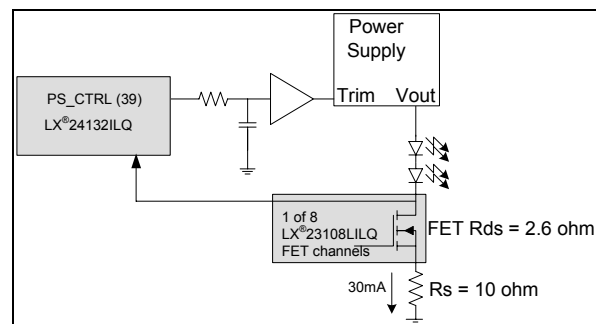


Figure 10: Simplified Power Supply Control Loop

Setting the ADC_RFVDMIN register to '1' causes the LX[®]24132ILQ 32 Port Display Backlight LED Controller to search for the channel with the lowest FET Vdrain ADC reading. When the searching process ends the LX[®]24132ILQ sets bit 0 in the ADC_MEAS_DONE register. The CPU/FPGA/Video Controller manages the LX[®]24132ILQ that reads the ADC_MIN_VD register (reports channel number and



LX24132ILQ - 32 Port Display Backlight LED Controller

lowest FET Vdrain ADC reading). As shown in Figure 10, the current is 30 mA. Minimum FET Vdrain should be at least:

- 30 mA x (10 Ohm + 2.6 Ohm) = 0.378 Volt
- 10 Ohm = FET external sense resistor
- 2.6 Ohm = LX[®]23108LILQ FET rds resistance

The ADC_MIN_VD register reading can be converted to voltage by:

$$\text{Min FET Vdrain[V]} = \frac{2 \text{ V}}{128} \times \text{ADC_MIN_VD}[0-6]$$

The Controller increases the power supply voltage when the minimum FET Vdrain is too low and decreases it in cases where minimum FET Vdrain is too high.

Power supply trimming is implemented by writing to the GLB_PS_CTRL register (10 bit DAC) which uses a PDM system to generate an equally distributed number of pulses proportional to the GLB_PS_CTRL value. Each pulse's duration equals to a single CLK clock length.

After low pass filtering, the PS_CTRL output DC component is used to trim the LED power supply (see AN-182 for more details).

ADC_UV register: Prevents very low ADC readings from being part of the minimum FET Vdrain search algorithm (reported by the ADC_MIN_VD register).

Any FET Vdrain ADC reading lower than ADC_UV register's value is not compared with the ADC_MIN_VD register. By setting ADC_UV to typical value such as '3':

$$[2000 \text{ mV}/128] \times 3 = 47 \text{ mV}$$

This value enforces the ADC_MIN_VD register to reject any ADC reading below 47 mV. This protection mechanism is required when the disconnected channel is enabled, or one of the LEDs had failed and was disconnected.

Internal A/D Operation

The internal ADC of the LX[®]24132ILQ 32 Port Display Backlight LED Controller measures the voltage on the VDMEAS input pin, which connects to the LX[®]23108ILQ 8 Port LED Drivers VDMEAS output pin. The LX[®]24132ILQ determines which FET vd voltage of the LX[®]23108ILQ is multiplexed to the VDMEAS input pin for FET vdrain measurement. ADC_SKIP, ADC_VDSTA and ADC_ON registers control the internal ADC operation (see Figure 11).

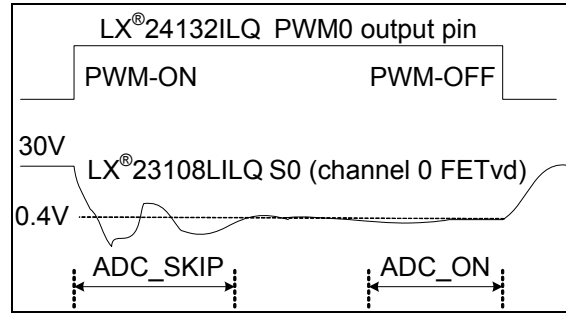


Figure 11: ADC Operation

The LX[®]24132ILQ contains two additional ADCs used by the system controller which manages the LX[®]24132ILQ. The ADCs can be configured to measure any of the 32 PWM channels FET Vdrain voltages, report under-voltage, over-voltage and skip an individual PWM channel.

ADC_SKIP register: Determines the time in which no FET Vdrain measurements are taken (starting from PWM-ON). This is done to prevent incorrect FET Vdrain measurements during the period of time where FET Vdrain is unstable. Typical ADC_SKIP value should be around 100 μsec (see Figure 11).

$$\text{ADC_SKIP} = \frac{\text{CLK}[\text{KHz}] \times \text{t}_{\text{adc_skip}}[\text{uSec}]}{1000 \times (\text{PWM_FRDIV}+1)}$$

ADC_ON register: Controls the allocated time for the A/D system to measure FET Vdrain of the various channels. If only some channels were measured, unmeasured channels are measured on the next PWM cycle (see Figure 11).

A typical ADC channel measurement takes around 6 μsec, assuming that the ADC_VDSTA register is set to 5 μsec and CLK input pin is connected to an 8 MHz clock source.

ADC measurements are conducted by the state machine from channel #0 to channel #31. In cases where a channel is temporarily turned off (due to scan input pin, or phase shift), ADC measurements are not performed although there is enough time for such measurements.

The measurements continue only after the channel will be turn on.

$$\text{ADC_ON} = \frac{\text{CLK}[\text{KHz}] \times \text{t}_{\text{adc_on}}[\text{uSec}]}{1000 \times (\text{PWM_FRDIV}+1)}$$

The LX[®]24132ILQ 32 Port Display Backlight LED Controller can measure PWM as low as ADC_SKIP+ADC_ON. For example, assuming ADC_SKIP = 150 μsec, ADC_ON = 50 μsec and



LX24132ILQ - 32 Port Display Backlight LED Controller

PWM frequency = 120 Hz. The minimum duty cycle which can be measured is:

$$\left[\frac{150 e^{-6} + 50 e^{-6}}{\frac{1}{120}} \right] \times 100\% = 2.4\%$$

Lower PWM duty cycle can be generated by the LX[®]24132ILQ. However, the internal ADC will not be able to read FET Vdrain voltage for channels with lower PWM duty cycle.

Note: The LX[®]24132ILQ 32 Port Display Backlight LED Controller sets skip_error bit2 in the GLB_ERR register in cases where it was not able to measure FET Vdrain, due to low PWM duty cycle.

Single ADC conversion takes around 6 μsec (assuming that the ADC_VDSTA register was set to 5 μsec). It takes 192 μsec (6 μsec x 32) to complete all 32 channels' FET Vdrain reading. Setting ADC_ON to 50 μSec requires four PWM cycles (8.3 mSec x 4 = 33.2 mSec for 120 Hz PWM cycle) to complete all 32 channels FET Vdrain reading.

Increasing the ADC_ON decreases the time required by the LX[®]24132ILQ to sample all 32 FET Vdrain channels, but increases the minimum FET Vdrain readable PWM duty cycle.

ADC_VDSTA register controls the analog mux delay during LX[®]23108LILQ channel change, before stable FET Vdrain measurement can be performed. LX[®]23108LILQ analog mux delay should be set to 5 μsec.

$$ADC_VDSTA = \frac{CLK[KHz] \times 5 \mu Sec}{1000}$$

An additional 1 μSec is required for ADC conversion; therefore it takes around 6 μSec to measure a single FET Vdrain channel.

Current Setting

Current setting is done by GLB_CURR register.

$$GLB_CURR = 0.7168 \times I[mA] \times Rs[ohm]$$

Rs = sense resistor connected to the LX[®]23108ILQ/S[N] pin.

Fault Detection

Fault events are reported by the GLB_ERR register. Three types of errors and events are reported by bits 0 - 2.

Bit0: under_voltage: Will be set whenever one or more out of 32 channels FET Vdrain voltage value is/are below the ADC_UV register's value. A typical

scenario is where a LED is disconnected, or the LX[®]24132ILQ's unconnected channel is enabled by the GLB_PRTEN register.

Note: Setting bit0 in the GLB_CLR_ERR register to '1' clears this error bit.

Bit1: over_temp_over_volt: Will be set whenever FET Vdrain exceeds 7.5 VDC - 11.5 VDC. A typical scenario is whenever a LED becomes shorted, or LED power supply was trimmed to too high voltage.

Note: Setting bit0 in the GLB_CLR_ERR register's to '1' clears this bit.

Bit2: skip: Will be set in cases where at least one out of the 32 channels is not measured by the internal ADC due to low PWM duty cycle.

Note: This bit clears itself whenever the LX[®]24132ILQ measures all 32 channels with no need to skip channels with a very low duty cycle.



LX24132ILQ - 32 Port Display Backlight LED Controller

Table 1: Configuration Mode Write Registers Description (For complete SPI bits pattern, see Figure 6)

CONFIGURATION MODE WRITE REGISTERS				
REGISTER NAME	DESCRIPTION	SPI - BIT POSITION	LENGTH [BITS]	RESET VALUE
GLB_PRTEN	<p>Bit0 - 31: Enable/Disable PWM channels pulse generation on PWM[N] output pins.</p> <p>Bit[N] 0: PWM[N] output pin is disabled Bit[N] 1: PWM[N] output pin is enabled</p> <p>N = 0 - 31 Note: Unused channels should be disabled Field is write only</p>	0 - 31	32	0x0000
PWM_FRDIV	<p>Divide system clock (pin 5) which drives the internal 12 bit PWM counter by: PWM_FRDIV+1.</p> <p>0: Divide by 1 31: Divide by 32</p> $\text{PWM Frequency} = \frac{\text{CLK}}{(4096 \times (\text{PWM_FRDIV} + 1))}$ <p>Example: For a system clock of 7.86432 MHz and PWM_FRDIV set to 15 (divide by 16), PWM frequency is 120 Hz. Field is write only</p>	32 - 36	5	0x0F
	Reserved	37 - 39	3	
ADC_UV	<p>Any LX[®]23108ILQ FET Vdrain ADC measurement during PWM-ON below ADC_UV will be ignored by internal logic responsible to find PWM 0-31 string with the lowest FET Vdrain ADC reading.</p> <p>0: 0 Volt 127: 2 Volt</p> <p>Typical ADC_UV value should be 3 (47 mVolt)</p> <p>Note: If the FET Vdrain measurement in one channel is below the ADC_UV, GLB_ERR register bit0 is set to '1' Field is read-write</p>	40 - 46	7	0x07
	Reserved	47 - 51	5	
GLB_CURR	<p>Bit0-7 control LED current in all PWM channels.</p> $\text{GLB_CURR} = 0.7168 \times I[\text{mA}] \times R_s[\text{ohm}]$ <p>$R_s = \text{LX}^{\text{®}}23108\text{ILQ}$ sense resistor connected to pin S[0-7] Field is read-write</p>	52 - 59	8	0x80
	Reserved	60 - 63	4	



LX24132ILQ - 32 Port Display Backlight LED Controller

www.Microsemi.com

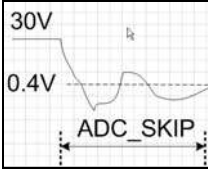
LX24132ILQ

CONFIGURATION MODE WRITE REGISTERS																																								
REGISTER NAME	DESCRIPTION	SPI - BIT POSITION	LENGTH [BITS]	RESET VALUE																																				
GLB_SCAN_0_15	<p>Each two bits associate their channel to one of the SCAN0-SCAN3 input pins and one of PWM_OFF0 - PWM_OFF3 registers. Setting SCANx input signal to Low, forces all PWM channels assigned to same input SCANx pin to turn off. Restoring SCANx input signal to High restores all assigned channels to normal operation.</p> <p>Example: Setting GLB_SCAN_0_15 register bit0 = High, Bit1 = High, assign SCAN3 input signal to control PWM channel 0, as well as assigning PWM_OFF3 register to control the turn off timing of PWM channel 0 (used for PWM phase shift). Field is read-write</p> <table border="1"> <thead> <tr> <th>Bit 2*N+1 (N = channel)</th> <th>Bit 2*N (N = channel)</th> <th>SCANx Input pin</th> <th>PWM_OFFx Register</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>SCAN0</td> <td>PWM_OFF0</td> </tr> <tr> <td>0</td> <td>1</td> <td>SCAN1</td> <td>PWM_OFF1</td> </tr> <tr> <td>1</td> <td>0</td> <td>SCAN2</td> <td>PWM_OFF2</td> </tr> <tr> <td>1</td> <td>1</td> <td>SCAN3</td> <td>PWM_OFF3</td> </tr> </tbody> </table> <p>Note: View GLB_CCR register related to the scanning mode and PWM synchronization.</p> <table border="1"> <tbody> <tr> <td>Bit0-1: PWM Channel [0]</td> <td>Bit16-17: PWM Channel [8]</td> </tr> <tr> <td>Bit2-3: PWM Channel [1]</td> <td>Bit18-19: PWM Channel [9]</td> </tr> <tr> <td>Bit4-5: PWM Channel [2]</td> <td>Bit20-21: PWM Channel [10]</td> </tr> <tr> <td>Bit6-7: PWM Channel [3]</td> <td>Bit22-23: PWM Channel [11]</td> </tr> <tr> <td>Bit8-9: PWM Channel [4]</td> <td>Bit24-25: PWM Channel [12]</td> </tr> <tr> <td>Bit10-11: PWM Channel [5]</td> <td>Bit26-27: PWM Channel [13]</td> </tr> <tr> <td>Bit12-13: PWM Channel [6]</td> <td>Bit28-29: PWM Channel [14]</td> </tr> <tr> <td>Bit14-15: PWM Channel [7]</td> <td>Bit30-31: PWM Channel [15]</td> </tr> </tbody> </table>	Bit 2*N+1 (N = channel)	Bit 2*N (N = channel)	SCANx Input pin	PWM_OFFx Register	0	0	SCAN0	PWM_OFF0	0	1	SCAN1	PWM_OFF1	1	0	SCAN2	PWM_OFF2	1	1	SCAN3	PWM_OFF3	Bit0-1: PWM Channel [0]	Bit16-17: PWM Channel [8]	Bit2-3: PWM Channel [1]	Bit18-19: PWM Channel [9]	Bit4-5: PWM Channel [2]	Bit20-21: PWM Channel [10]	Bit6-7: PWM Channel [3]	Bit22-23: PWM Channel [11]	Bit8-9: PWM Channel [4]	Bit24-25: PWM Channel [12]	Bit10-11: PWM Channel [5]	Bit26-27: PWM Channel [13]	Bit12-13: PWM Channel [6]	Bit28-29: PWM Channel [14]	Bit14-15: PWM Channel [7]	Bit30-31: PWM Channel [15]	64 - 95	32	Not defined
Bit 2*N+1 (N = channel)	Bit 2*N (N = channel)	SCANx Input pin	PWM_OFFx Register																																					
0	0	SCAN0	PWM_OFF0																																					
0	1	SCAN1	PWM_OFF1																																					
1	0	SCAN2	PWM_OFF2																																					
1	1	SCAN3	PWM_OFF3																																					
Bit0-1: PWM Channel [0]	Bit16-17: PWM Channel [8]																																							
Bit2-3: PWM Channel [1]	Bit18-19: PWM Channel [9]																																							
Bit4-5: PWM Channel [2]	Bit20-21: PWM Channel [10]																																							
Bit6-7: PWM Channel [3]	Bit22-23: PWM Channel [11]																																							
Bit8-9: PWM Channel [4]	Bit24-25: PWM Channel [12]																																							
Bit10-11: PWM Channel [5]	Bit26-27: PWM Channel [13]																																							
Bit12-13: PWM Channel [6]	Bit28-29: PWM Channel [14]																																							
Bit14-15: PWM Channel [7]	Bit30-31: PWM Channel [15]																																							
GLB_SCAN_16_31	<p>Same as GLB_SCAN_0_15 for PWM channels 16-31. Field is read-write</p> <table border="1"> <tbody> <tr> <td>Bit0-1: PWM[16]</td> <td>Bit16-17: PWM[24]</td> </tr> <tr> <td>Bit2-3: PWM[17]</td> <td>Bit18-19: PWM[25]</td> </tr> <tr> <td>Bit4-5: PWM[18]</td> <td>Bit20-21: PWM[26]</td> </tr> <tr> <td>Bit6-7: PWM[19]</td> <td>Bit22-23: PWM[27]</td> </tr> <tr> <td>Bit8-9: PWM[20]</td> <td>Bit24-25: PWM[28]</td> </tr> <tr> <td>Bit10-11: PWM[21]</td> <td>Bit26-27: PWM[29]</td> </tr> <tr> <td>Bit12-13: PWM[22]</td> <td>Bit28-29: PWM[30]</td> </tr> <tr> <td>Bit14-15: PWM[23]</td> <td>Bit30-31: PWM[31]</td> </tr> </tbody> </table>	Bit0-1: PWM[16]	Bit16-17: PWM[24]	Bit2-3: PWM[17]	Bit18-19: PWM[25]	Bit4-5: PWM[18]	Bit20-21: PWM[26]	Bit6-7: PWM[19]	Bit22-23: PWM[27]	Bit8-9: PWM[20]	Bit24-25: PWM[28]	Bit10-11: PWM[21]	Bit26-27: PWM[29]	Bit12-13: PWM[22]	Bit28-29: PWM[30]	Bit14-15: PWM[23]	Bit30-31: PWM[31]	96 - 127	32	Not defined																				
Bit0-1: PWM[16]	Bit16-17: PWM[24]																																							
Bit2-3: PWM[17]	Bit18-19: PWM[25]																																							
Bit4-5: PWM[18]	Bit20-21: PWM[26]																																							
Bit6-7: PWM[19]	Bit22-23: PWM[27]																																							
Bit8-9: PWM[20]	Bit24-25: PWM[28]																																							
Bit10-11: PWM[21]	Bit26-27: PWM[29]																																							
Bit12-13: PWM[22]	Bit28-29: PWM[30]																																							
Bit14-15: PWM[23]	Bit30-31: PWM[31]																																							



LX24132ILQ - 32 Port Display Backlight LED Controller

www.Microsemi.com

CONFIGURATION MODE WRITE REGISTERS												
REGISTER NAME	DESCRIPTION	SPI - BIT POSITION	LENGTH [BITS]	RESET VALUE								
PWM_OFF3	<p>All PWM channels which were assigned by the GLB_SCAN_0_15 and GLB_SCAN_16_31 register to PWM_OFF3 will be turned off each time 0-4095 internal cyclic counter reaches the PWM_OFF3 register value; see Figure 9.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Phase Shift</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0%</td> </tr> <tr> <td>2048</td> <td>50%</td> </tr> <tr> <td>4095</td> <td>99.99%</td> </tr> </tbody> </table> <p>Note that the 0-4095 internal counter increments at a rate of sys clock/ (PWM_FRDIV+1) Use this register to perform PWM phase shift; see Figure 9. Field is write only</p>	Value	Phase Shift	0	0%	2048	50%	4095	99.99%	128 - 139	12	Not defined
Value	Phase Shift											
0	0%											
2048	50%											
4095	99.99%											
PWM_OFF2	Same as PWM_OFF3 for all PWM channels which were assigned to use PWM_OFF2. See Figure 9.	140 - 151	12	Not defined								
PWM_OFF1	Same as PWM_OFF3 for all PWM channels which were assigned to use PWM_OFF1. See Figure 9.	152 - 163	12	Not defined								
PWM_OFF0	Same as PWM_OFF3 for all PWM channels which were assigned to use PWM_OFF0. See Figure 9.	164 - 175	12	Not defined								
	Reserved	176 - 179	4									
ADC_SKIP	<p>Prevents internal ADC from measuring FET Vdrain as long as FET Vdrain voltage is unstable. Delay is required to ensure stable ADC Vdrain measurement. Typical value should be around 100 usec. For the best recommended value, contact Microsemi's technical support. See Figure 11. Field is write only</p>  <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> $ADC_SKIP = \frac{CLK[KHz] \times t_{adc_skip}[\mu Sec]}{1000 \times (PWM_FRDIV+1)}$ </div>	180 - 191	12	0x011								
	Reserved	192 - 195	4									

LX24132ILQ



LX24132ILQ - 32 Port Display Backlight LED Controller

www.Microsemi.com

CONFIGURATION MODE WRITE REGISTERS				
REGISTER NAME	DESCRIPTION	SPI - BIT POSITION	LENGTH [BITS]	RESET VALUE
ADC_ON	<p>Allocated time for internal ADC to sample the 32 PWM channels. In cases where ADC cannot measure all 32 channels during a single PWM pulse, it continues channel measurement on the next PWM cycle.</p> <p>ADC_ON time is measured from the end of the PWM pulse. See Figure 11. For the best recommended value, contact Microsemi's technical support.</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> $ADC_ON = \frac{CLK[KHz] \times t_{adc_on}[uSec]}{1000 \times (PWM_FRDIV+1)}$ </div> <p>Note: Minimum ADC_ON is 6µsec Field is write only</p>	196 - 207	12	0x003
ADC_VDSTA	<p>The time required for the LX[®]23108LILQ internal mux to stabilize before the LX[®]24132ILQ 32 Port Display Backlight LED Controller can measure the FET Vdrain voltage. Stabilization time = 5 µsec.</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> $ADC_VDSTA = \frac{CLK[KHz] \times 5uSec}{1000}$ </div> <p>Example: LX[®]24132ILQ CLK frequency = 7864 KHz. ADC_VDSTA = (7864 x 5)/1000 = 39 Field is read-write</p>	208 - 215	8	0x28

LX24132ILQ



LX24132ILQ - 32 Port Display Backlight LED Controller

CONFIGURATION MODE WRITE REGISTERS				
REGISTER NAME	DESCRIPTION	SPI - BIT	LENGTH	RESET VALUE
		POSITION	[BITS]	
GLB_CCR	<p>Bit 0: Scanning_Mode: Synchronization control of PWM outputs to SCAN0-SCAN3 input pins and PWM cycle.</p> <p>0: PWM output is synchronized with the SCANx input pin and internal PWM cycle, see above figure. 1: PWM output follows SCANx input SIGNAL asynchronously in respect to internal PWM cycle, see above figure. This bit is read-write.</p> <p>Bit 1: sync_in_dis: Disables PWM output synchronization with SYNC_IN input pin.</p> <p>0: PWM[N] output pins are synchronized with SYNC_IN signal. 12 bit internal PWM counter will be reset on each SYNC_IN rising edge pulse. 1: SYNC_IN input pin has no effect on PWM outputs. This bit is read-write.</p> <p>Bit 2: sleep_mode: Switch off analog and partial digital sections of the chip for power saving purposes.</p> <p>0: Normal operation 1: Switch off analog and partial digital chip sections. This bit is read-write.</p> <p>Bit 3: sync_out_disable: Enable/Disable pulse at SYNC_OUT output pin each time internal PWM counter reaches "0".</p> <p>0: Enable sync_out. Set Bit4 & bit5 to 00 1: SYNC_OUT output pin = Low. This bit is read-write.</p> <p>Bit 4&5: sync_out_func: For internal use. Set bits 4 and 5 to Low. These bits are write only. Read value always 0.</p> <p>Bit 6: read_mode: 0: For internal use. Set to Low. This bit is write only. Read value always 0.</p>	216 - 222	7	0x00



LX24132ILQ - 32 Port Display Backlight LED Controller

www.Microsemi.com

CONFIGURATION MODE WRITE REGISTERS				
REGISTER NAME	DESCRIPTION	SPI - BIT	LENGTH	RESET VALUE
		POSITION	[BITS]	
GLB_TMP_OV	<p>Enable/Disable LX[®]23108ILQ internal FET Vdrain over-voltage and over-temperature protection.</p> <p>0: Enabled Over Voltage: LX[®]23108ILQ 8 Port LED Drivers turns off PWM channels whenever FET Vdrain voltage exceeds 7.5 VDC -11.5 VDC. To re-enable turned off channels, the user should set bit0 in the GLB_CLR_ERR register. Over Temperature: LX[®]23108ILQ turns off PWM channels whenever FET temperature exceeds 180 +/- 20[c]. To re-enable turned off channels, the user should set bit0 in register GLB_CLR_ERR.</p> <p>1: Disabled Over Voltage: LX[®]23108ILQ FTE Vd over-voltage protection is disabled. Over Temperature: LX[®]23108ILQ turns off PWM channels whenever FET temperature exceeds 180 +/- 20[c]. However, PWM channels are automatically re-enabled whenever FET cools down.</p> <p>Note: Setting bit0 in GLB_CLR_ERR register re-enables PWM channels, regardless of FET temperature. However, if FET temperature is too high, FET channels are turned off again. This field is read-write.</p>	223	1	0x1

Table 2: Configuration Mode Read Registers Description (For complete SPI bits pattern, see Figure 6)

CONFIGURATION MODE READ REGISTERS				
REGISTER NAME	DESCRIPTION	SPI - BIT	LENGTH	RESET VALUE
		POSITION	[BITS]	
	Reserved	0 - 167	168	
GLB_CHVER	Chip version; for internal use	168 - 183	16	0x180
CHIP_CONF	Chip sub version; for internal use	184 - 185	2	0x0
	Reserved	186 - 187	2	
GLB_CURR_R	Reports same value as written to GLB_CURR register (bits 52-59).	188 - 195	8	0x80
	Reserved	196 - 199	4	
ADC_VDSTA_R	Reports same value as written to ADC_VDSTA register (bits 208-215).	200 - 207	8	0x28
GLB_CCR_R	Reports same value as written to GLB_CCR register (bits 216-222).	208 - 214	7	0x0
GLB_TMP_OV_R	Reports same value as written to GLB_TMP_OV register (bit 223).	215	1	0x1
	Reserved	216 - 223	8	

LX24132ILQ



LX24132ILQ - 32 Port Display Backlight LED Controller

Table 3: Operation Mode Write Registers Description (For complete SPI bits pattern, see Figure 6)

OPERATION MODE WRITE REGISTERS				
REGISTER NAME	DESCRIPTION	SPI - BIT POSITION	LENGTH [BITS]	RESET VALUE
ADC_DATA2_CH	0-31: Select which PWM FET Vdrain channel should be measured by ADC2. 32: ADC2 measures the LX [®] 24132ILQ 32 Port Display Backlight LED Controller chip temperature. Refer to ADC_DATA2 register (Operation mode read registers bits 368-375) This field is read-write.	0 - 5	6	0x00
	Reserved	6 - 7	2	
ADC_DATA1_CH	0-31: Selects which PWM FET Vdrain channel should be measured by ADC1. 32: ADC1 measures LX [®] 24132ILQ chip temperature. Refer to ADC_DATA1 register (Operation mode read registers bits 376-383). This field is read-write.	8 - 13	6	0x00
	Reserved	14 - 15	2	
GLB_PS_CTRL	Ten bit D/A used to control PS voltage feeding the LEDs. Register value controls the PDM (Pulsed Density Modulation) signal on the PS_CTRL output pin. External hardware converts the PDM signal to DC voltage ranging from 0 - 3.3 VDC. This field is write only.	16 - 25	10	0x100
	Reserved	26 - 27	2	
ADC_RFVDMIN	Setting bit to 1 sets ADC_MINVD register measured bits 0-6 to 0x7F and clears valid bit 7 (operation mode read bits 392-398, 399). ADC_RFMINVD bit is immediately cleared. Chip starts measuring FET Vdrain for all 32 strings. After all 32 FET Vdrain channels are measured, ADC_MIN_VD register bits 0-6 (adc_measure bits) contains the lowest FET Vdrain ADC reading. ADC_MIN_VD Valid bit 7 is set to '1' and ADC_MEASURE_DONE register bit 0 is set to '1'. Whenever ADC_MEASURE_DONE register equals '1', it means that LX [®] 24132ILQ had found the lowest FET Vdrain out of 32 channels. The user should read the ADC_MIN_VD register's value and trim PS voltage by writing to the GLB_PS_CTRL register. Setting again ADC_RFMINVD bit to 1 will start new minimum FET Vdrain search cycle. This field is write only.	28	1	0x0



LX24132ILQ - 32 Port Display Backlight LED Controller

OPERATION MODE WRITE REGISTERS				
REGISTER NAME	DESCRIPTION	SPI - BIT POSITION	LENGTH [BITS]	RESET VALUE
GLB_CLR_ERR	Setting bit to 1, clears GLB_ERR register error bits (read bits 388-391) and re-enables all channels that were turned off by the LX [®] 23108ILQ 8 Port LED Drivers due to FET Vdrain over-voltage or over-temperature. Note that over temperature/over-voltage bit in the GLB_ERR register cannot be cleared by the GLB_CLR_ERR register in cases where it was set due to over temperature event. This field is cleared on write.	29	1	0x0
	Reserved	30 - 31	2	
PWM_31	Sets PWM duty cycle from 0-99.999% by writing to PWM_31 register values ranging from 0-4095. Note1: New PWM_0 to PWM_31 values are loaded to the PWM_0–PWM_31 registers on the rising edge of CS_N input signal (SPI chip select) and are effective whenever PWM channel is turned off (PWM_OFF timing). This field is read-write.	32 - 43	12	Not defined
PWM_30	See PWM_31	44 - 55	12	Not defined
PWM_29	See PWM_31	56 - 67	12	
PWM_28	See PWM_31	68 - 79	12	
PWM_27	See PWM_31	80 - 91	12	
PWM_26	See PWM_31	92 - 103	12	
PWM_25	See PWM_31	104 - 115	12	
PWM_24	See PWM_31	116 - 127	12	
PWM_23	See PWM_31	128 - 139	12	
PWM_22	See PWM_31	140 - 151	12	
PWM_21	See PWM_31	152 - 163	12	
PWM_20	See PWM_31	164 - 175	12	
PWM_19	See PWM_31	176 - 187	12	
PWM_18	See PWM_31	188 - 199	12	
PWM_17	See PWM_31	200 - 211	12	
PWM_16	See PWM_31	212 - 223	12	
PWM_15	See PWM_31	224 - 235	12	
PWM_14	See PWM_31	236 - 247	12	
PWM_13	See PWM_31	248 - 259	12	
PWM_12	See PWM_31	260 - 271	12	
PWM_11	See PWM_31	272 - 283	12	
PWM_10	See PWM_31	284 - 295	12	
PWM_09	See PWM_31	296 - 307	12	
PWM_08	See PWM_31	308 - 319	12	
PWM_07	See PWM_31	320 - 331	12	
PWM_06	See PWM_31	332 - 343	12	
PWM_05	See PWM_31	344 - 355	12	
PWM_04	See PWM_31	356 - 367	12	

www.Microsemi.com

LX24132ILQ



LX24132ILQ - 32 Port Display Backlight LED Controller

OPERATION MODE WRITE REGISTERS				
REGISTER NAME	DESCRIPTION	SPI - BIT POSITION	LENGTH [BITS]	RESET VALUE
PWM_03	See PWM_31	368 - 379	12	
PWM_02	See PWM_31	380 - 391	12	
PWM_01	See PWM_31	392 - 403	12	
PWM_00	See PWM_31	404 - 415	12	



LX24132ILQ - 32 Port Display Backlight LED Controller

Table 4: Operation Mode Read Registers Description (for a complete SPI bits pattern, see Figure 6)

OPERATION MODE READ REGISTERS				
REGISTER NAME	DESCRIPTION	SPI - BIT	LENGTH	RESET VALUE
		POSITION	[BITS]	
	Reserved	0 - 351	352	
ADC_DATA2_CH_R	Reports which FET Vdrain channel (0-31), or LX [®] 24132ILQ temperature (32) was read by the ADC (same as ADC_DATA2_CH register). This field is read-write.	352 - 357	6	Not defined
	Reserved	358 - 359	2	
ADC_DATA1_CH_R	Reports which FET Vdrain channel (0-31), or LX [®] 24132ILQ temperature (32) was read by the ADC (same as ADC_DATA1_CH register). This field is read-write.	360 - 365	6	Not defined
	Reserved	366 - 367	2	
ADC_DATA2	<p>Bit 0-6: adc_measure for FET Vdrain channel 0-31: ADC_DATA2_CH channel register determines which LX[®]23108ILQ 8 Port LED Drivers FET Vdrain channel is measured by the LX[®]24132ILQ). Result = Bit 0-6. Value of '127' is equivalent to 2 Volt.</p> $\text{FET Vdrain[mV]} = \frac{\text{adc_measure}}{128} \times 2000$ <p>These bits are read only.</p> <p>Bit 0-6: adc_measure for LX[®]24132ILQ temperature (ch 32):</p> $\text{Temperature[c]} = 4.3427 \times \text{ADC_DATA2} - 273$ <p>These bits are read only.</p> <p>Bit 7: valid 0: adc_measure bits are invalid 1: adc_measure bits are valid This bit is cleared on read.</p> <p>Note: Since valid bit is of the "clear on read" type (RC), each SPI transaction clears this bit.</p>	368 - 375	8	Not defined



LX24132ILQ - 32 Port Display Backlight LED Controller

OPERATION MODE READ REGISTERS				
REGISTER NAME	DESCRIPTION	SPI - BIT	LENGTH	RESET VALUE
		POSITION	[BITS]	
ADC_DATA1	<p>Bit 0-6: adc_measure for FET Vdrain channel 0-31: ADC_DATA2_CH channel register determines which LX[®]23108ILQ 8 Port LED Drivers FET Vdrain channel will be measured by the LX[®]24132ILQ). Result = Bit 0-6 These bits are read only.</p> <p>Bit 0-6: adc_measure for LX[®]24132ILQ temperature (ch 32): These bits are read only. Temperature[c] = 4.3427 x ADC_DATA1 - 273</p> <p>Bit 7: valid: 0: adc_measure bits are invalid 1: adc_measure bits are valid. This bit is cleared on read.</p> <p>Note: Since valid bit is of the "clear on read" type (RC), each SPI transaction clears this bit.</p>	376 - 383	8	Not defined
ADC_DATA2_ERR	<p>Bit 0: under_voltage: 0: Normal operation 1: ADC reading for channel number as per ADC_DATA2_CH_R register was below ADC_UV register (configuration mode bit 40 - 46) – LED failure, disconnection failure.</p> <p>Bit 1: over_temp_over_volt: 0: Normal operation 1: FET Vdrain over-voltage (7.5 VDC - 11.5 VDC) or over-temperature event occurred in channel number as per ADC_DATA2_CH_R register. Set bit 0 in GLB_CLR_ERR register to 1 clear both bits. Note that ADC_DATA2_ERR error is valid only in cases where Bit 7: valid in the ADC_DATA2 register is set. This field is read only.</p>	384 - 385	2	0x0



LX24132ILQ - 32 Port Display Backlight LED Controller

OPERATION MODE READ REGISTERS				
REGISTER NAME	DESCRIPTION	SPI - BIT	LENGTH	RESET VALUE
		POSITION	[BITS]	
ADC_DATA1_ERR	<p>Bit 0: under_voltage: 0: Normal operation 1: ADC reading for channel number as per ADC_DATA1_CH_R register was below ADC_UV register (configuration mode bit 40 - 46) – LED failure, disconnection failure.</p> <p>Bit 1: over_temp_over_volt: 0: Normal operation 1: FET Vdrain over voltage (7.5 VDC - 11.5 VDC), or over-temperature event occurred in channel number as per ADC_DATA1_CH_R register. Setting bit 0 in GLB_CLR_ERR register to 1 clear both bits. Note that ADC_DATA1_ERR error is valid only in cases where Bit 7: valid in the ADC_DATA1 register is set. This field is read only.</p>	386 - 387	2	0x0
GLB_ERR	<p>Bit 0: under_voltage: 0: Normal operation 1: Under-voltage error occurred in one of the channels</p> <p>Setting bit 0 in GLB_CLR_ERR register to 1 clear this bit.</p> <p>Bit 1: over_temp_over_volt: 0: Normal operation 1: FET Vdrain over-voltage or over-temperature occurred in one of the channels</p> <p>Setting bit 0 in GLB_CLR_ERR register to 1 clear this bit.</p> <p>Bit 2: skip: 0: Normal operation 1: At least one of the 32 channels was not sampled by the internal ADC, due to low PWM duty cycle.</p> <p>Bit is cleared automatically when PWM duty cycle is increased and all 32 channels can be measured by the internal ADC. This field is read only.</p>	388 - 390	3	0x0
	Reserved	391	1	



LX24132ILQ - 32 Port Display Backlight LED Controller

OPERATION MODE READ REGISTERS				
REGISTER NAME	DESCRIPTION	SPI - BIT	LENGTH	RESET VALUE
		POSITION	[BITS]	
ADC_MIN_VD	<p>Bit 0-6: adc_measure: Setting bit 0 in ADC_RFVDMIN register to '1', sets bit 0-6 to 0x7F (highest value equivalent to 2 Volt). Chip continuously scans channels 0-31. Whenever the ADC reading is lower than bits 0-6 value, bits 0-6 are updated with the new ADC reading (lower value).</p> <p style="text-align: center;">2 V</p> <p>Minimum FET V_{drain}[v] = x adc_measure / 128</p> <p>Note: FET V_{drain} 0-31 channels are continuously scanned. The adc_measure is updated with the lowest FET V_{drain} value.</p> <p>Bit 7: valid (clear on read): 0: adc_measure bits are invalid 1: adc_measure bits are valid</p> <p>Note 1: Since valid bit is of the "clear on read" type (RC), each SPI transaction clears this bit.</p> <p>Note 2: Reading ADC_MIN_VD register before the ADC_MEAS_DONE register is set clears the valid bit.</p> <p>Note 3: In cases where a valid bit is set during the time that ADC_RFVDMIN was set to '1' until ADC_MEAS_DONE become '1' consider adc_measure bits and channel bits as valid, regardless of valid bit7 state.</p> <p>Bit 8 - 12: channel: Channel number (0-31) having the lowest FET V_{drain} reading. This field is read only.</p>	392 - 404	13	0x7F
ADC_MEAS_DONE	<p>Setting bit 0 in ADC_RFVDMIN register clears this bit. This bit is set after all enabled channels (up to 32) FET V_{drain} are measured by the internal ADC.</p> <p>Note: Channels with low PWM duty cycle that cannot be measured are counted as if they were measured. This field is read only.</p>	405	1	0x0



LX24132ILQ - 32 Port Display Backlight LED Controller

OPERATION MODE READ REGISTERS				
REGISTER NAME	DESCRIPTION	SPI - BIT	LENGTH	RESET VALUE
		POSITION	[BITS]	
ADC_DATA1_SKIP	0: Normal operation 1: PWM duty cycle for channel reported by ADC_DATA1_CH_R register is too low and can't be measured. This field is read only.	406	1	0x0
ADC_DATA2_SKIP	0: Normal operation 1: PWM duty cycle for channel reported by register. ADC_DATA1_CH_R is too low, and can not be measured. This field is read only.	407	1	0x0
	Reserved	408 - 415	8	



LX24132ILQ - 32 Port Display Backlight LED Controller

Configuration Example and Reference Code

The code below provides a configuration example and reference code assisting in accessing the LX[®]24132 registers and performing SPI bit manipulation.

PARAMETER	VALUE	SOFTWARE #DEFINE<VAL> SEE MAIN.H
Synchronize PWM to external 120 Hz V-Sync clock	yes	
Number of LX [®] 24132ILQ IC's	4	MAX_NUM_LX24132_IC
LX [®] 24132ILQ system clock	7.864 MHz	LX24132_SYS_CLK_KHZ
System clock divider	15+1	LX24132_SYS_CLK_DIV
LED current	30 mA	LED_ImA
LX [®] 23108ILQ sense resistor	10 Ohm	FET_R_SENSE_OHM
LX [®] 23108ILQ internal FET RDS-On	2.6 Ohm	FET_RDS_ON_ONH
No ADC measurement from PWM-ON	150 μsec	ADC_SKIP_USEC
Time before PWM-OFF in which ADC can measure	100 μsec	ADC_ON_USEC

The reference code applies to 120 Hz PWM which can be synchronized with an external 120 Hz V-SYNC pulse.

$$\frac{7.864\text{MHz}}{(15+1) \times 4096} = 120\text{Hz}$$

Function main():

The Configuration function sets various LX[®]24132ILQ 32 Port Display Backlight LED Controller registers to update the bits to be sent over the SPI. Note that writing to those registers modifies only the RAM array bits and nothing is actually sent to the LX[®]24132ILQ.

```
LX24132_SetDef_Cfg_ModeBits_AllIC(LED_ImA);
```

```
LX24132_SetDef_Opr_ModeBits_AllIC();
```

Note: ADC_RFVDMIN register bit0 is configured as '1'. This simplifies the searching process of the FET Vdrain minimum in the following software loop.

After RAM bit array is properly set, configure the LX[®]24132ILQ by sending configured bits to the LX[®]24132ILQ over the SPI.

```
LX24132_SPI_TxRx(eCOMM_MODE_CFG);
```

```
LX24132_SPI_TxRx(eCOMM_MODE_OPR);
```

Upon completion of the configuration process, the LEDs power supply must be monitored to set the power supply voltage to optimal value. A power supply whose voltage is too high causes the LX[®]23108ILQLs to overheat.

A power supply whose voltage is too low causes incorrect LED current (30 mA in our example). Note that PWM values of the various strings can be modified before sending the operation mode SPI bit pattern which is synchronized on the next V-SYNC pulse. The reference code example doesn't modify PWM values, so only LEDs power supply trimming will be handled.

Since the `LX24132_SetDef_Opr_ModeBits_AllIC();` function configured register ADC_RFVDMIN bit0 is set to '1', there is no need to set it again (sent as '1' every 100 mSec).

Calling the `LX24132_SPI_TxRx(eCOMM_MODE_OPR);` every 100 mSec causes Tx/Rx exchange.

SPI_Master_TxRx() function is called to send/receive LX24132 operation-mode bit pattern.

Upon the end of transmission, the function checks the ADC_MEAS_DONE register's value. Bit0 of this register is set to '1' whenever the LX[®]24132ILQ ends scanning of all 32 PWM channels (channels with too low PWM duty cycle are skipped. The same applies for channels with FET Vdrain ADC reading below ADC_UV). If ADC_MEAS_DONE is '1' and ADC_MIN_VD register bit7 (valid) is '1', `Adj_LED_PowerSupply()` function is called.

This function compares the FET Vdrain minimum voltage with the desired FET Vdrain voltage and trimming power supply in order to reduce the voltage difference to a minimum.

Note: Functions that start with USER_xyz() should be filled in by the user, based on the real hardware being used by the user.



LX24132ILQ - 32 Port Display Backlight LED Controller

```
#include "main.h"

/*****
* main()
*
* Parameters:
*
* Return:
*****/
void main(void)
{
    // Set LX24132 Register (no SPI transaction)
    lx24132_SetDef_Cfg_ModeBits_AllIC(LED_ImA);
    lx24132_SetDef_Opr_ModeBits_AllIC();

    LX24132_SPI_TxRx(eCOMM_MODE_CFG); // Init: Tx/Rx LX24132 configuration reg over
SPI
    LX24132_SPI_TxRx(eCOMM_MODE_OPR); // Init: Tx/Rx LX24132 operation reg over SPI

    while(1)
    {
        USER_Wait_mSec(100);

        // Since we set ADC_RFVDMIN bit 0 to '1' inside function
lx24132_SetDef_Opr_ModeBits_AllIC(),
        // New FET Vd min measurement will be done for each eCOMM_MODE_OPR
transaction. The function which
        // sends the SPI bit pattern check if ADC_MIN_VD valid bit was set and there
is valid Min FET Vd
        // ADC reading. Based on this reading it will call Adj_LED_PowerSupply() in
order to trim
        // power supply to optimal voltage
        LX24132_SPI_TxRx(eCOMM_MODE_OPR);
    }
}

/*****
* lx24132_SetDef_Cfg_ModeBits_AllIC()
*
* This function fills LX24132 CFG_MODE SPI patterns which are the same
* for all LX24132 IC's
*
* Parameters:
*
* Return:
*****/
void lx24132_SetDef_Cfg_ModeBits_AllIC( U16 I_ma)
{
    U32 GlbCurr;
    U8 IC;

    GlbCurr = (U32)( (0.7168 * (float)I_ma * FET_R_SENSE_OHM) + 0.5);

    if (GlbCurr > 255)
        GlbCurr = 255;

    // Build CFG_MODE bit pattern for all LX24132 IC's
    for (IC=0 ; IC < MAX_NUM_LX24132_IC ; IC++)
```




LX24132ILQ - 32 Port Display Backlight LED Controller

```
{
    LX24132_SetReg(IC, eREG_GLB_PRTEN, 0xFFFFFFFF ); // Enable all 32 channels
    LX24132_SetReg(IC, eREG_GLB_CURR , GlbCurr ); // Set LED current
    LX24132_SetReg(IC, eREG_PWM_FRDIV, LX24132_SYS_CLK_DIV ); // div by (15+1)
    LX24132_SetReg(IC, eREG_ADC_UV , 3 ); // (3/128)x2000 = 47mVolt
    LX24132_SetReg(IC, eREG_GLB_SCAN_0_15 , 0 ); // Link all 32 PWM_OFF
timing to PWM_OFF0 reg
    LX24132_SetReg(IC, eREG_GLB_SCAN_16_31 , 0 );
    LX24132_SetReg(IC, eREG_PWM_OFF3 , 4095); // Turn off PWM when
internal counter reach 4095
    LX24132_SetReg(IC, eREG_PWM_OFF2 , 4095); // Not in use
    LX24132_SetReg(IC, eREG_PWM_OFF1 , 4095); // Not in use
    LX24132_SetReg(IC, eREG_PWM_OFF0 , 4095); // Not in use
    LX24132_SetReg(IC, eREG_ADC_SKIP , ADC_CONV_FROM_uSEC(ADC_SKIP_USEC)
); // 150uSec
    LX24132_SetReg(IC, eREG_ADC_ON , ADC_CONV_FROM_uSEC(ADC_ON_USEC) );
// 100uSec
    LX24132_SetReg(IC, eREG_ADC_VDSTA , VDSTA_CONV_FROM_uSEC(5) ); //
5uSec
    LX24132_SetReg(IC, eREG_GLB_CCR , 0 ); // PWM SYNC to SYNC_IN
input pin
    LX24132_SetReg(IC, eREG_GLB_TMP_OV , 1 ); // Disable over
voltage/over temp protection
}
}
/*****
* lx24132_SetDef_Opr_ModeBits_AllIC()
*
* This function fills LX24132 OPR_MODE SPI registers which are the same
* for all LX24132 IC's
*
* Parameters:
*
* Return:
*****/
void lx24132_SetDef_Opr_ModeBits_AllIC(void)
{
    U8 IC, Str;

    // Build default PWM_MODE bit pattern
    for (IC=0 ; IC < MAX_NUM_LX24132_IC ; IC++)
    {
        LX24132_SetReg(IC, eREG_ADC1_CH , 0 ); // measure string #0 (init but
not in use)
        LX24132_SetReg(IC, eREG_ADC2_CH , 1 ); // measure string #1 (init but
not in use)
        LX24132_SetReg(IC, eREG_GLB_PS_CTRL , 700); // Set Power Supply trim to 70%
of its max voltage
        LX24132_SetReg(IC, eREG_ADC_RFVDMIN , 1 ); // Find new Vmin

        // Setting GLB_CLR to '1'simplify GLB_UDL, GLB_OVT_OVL management. We get
errors plus they are cleared
        LX24132_SetReg(IC, eREG_GLB_CLR_ERR , 1 ); // Clr GLB_ERR reg bits. Re
enable strings shut down strings

        for (Str=0;Str<32;Str++)
            LX24132_SetReg(IC, (enum eLX24132_REG)(eREG_PWM_00 + Str), 409); // 10%
duty cycle
    }
}
```



LX24132ILQ - 32 Port Display Backlight LED Controller

```
/******  
* LX24132_SetReg()  
*  
* This function updates various bits inside LX24132_Cfg_Tx_Bits[][] and  
* LX24132_Opr_Tx_Bits[][] array according to the register which has to  
* be updated.  
*  
* NOTE: Only bits are updated. Nothing is being sent to Rose.  
*  
* Parameters:  
* U8 IC_Index: Rose IC index  
* enum eLX24132_REG eReg : Register inside Rose (converted to specific number  
* of bits to be updated inside long SPI write pattern  
* U32 Data : Register new data value. Please note: since biggest  
* reg width is 32 bits, this parameter was adjusted to  
* same size  
*  
* Return:  
*****/  
void LX24132_SetReg(U8 IC_Index, enum eLX24132_REG eReg, U32 Data)  
{  
    switch (eReg)  
    {  
        //-- Operation MODE registers ---  
        case eREG_ADC2_CH:  
            Set_Bits( (U16)Data, LX24132_Opr_Tx_Bits[IC_Index], OFST_REG_ADC_DATA2_CH,  
SIZE_REG_ADC_DATA2_CH);  
            break;  
  
        case eREG_ADC1_CH:  
            Set_Bits( (U16)Data, LX24132_Opr_Tx_Bits[IC_Index], OFST_REG_ADC_DATA1_CH,  
SIZE_REG_ADC_DATA1_CH);  
            break;  
  
        case eREG_GLB_PS_CTRL:  
            Set_Bits( (U16)Data, LX24132_Opr_Tx_Bits[IC_Index], OFST_REG_GLB_PS_CTRL,  
SIZE_REG_GLB_PS_CTRL);  
            break;  
  
        case eREG_ADC_RFVDMIN:  
            Set_Bits( (U16)Data, LX24132_Opr_Tx_Bits[IC_Index], OFST_REG_ADC_RFVDMIN,  
SIZE_REG_ADC_RFVDMIN);  
            break;  
        case eREG_GLB_CLR_ERR:  
            Set_Bits( (U16)Data, LX24132_Opr_Tx_Bits[IC_Index], OFST_REG_GLB_CLR_ERR,  
SIZE_REG_GLB_CLR_ERR);  
            break;  
  
        case eREG_PWM_31:  
            Set_Bits( (U16)Data, LX24132_Opr_Tx_Bits[IC_Index], OFST_REG_W_PWM31,  
SIZE_REG_W_PWM31);  
            break;  
  
        case eREG_PWM_30:  
            Set_Bits( (U16)Data, LX24132_Opr_Tx_Bits[IC_Index], OFST_REG_W_PWM30,  
SIZE_REG_W_PWM30);  
            break;  
  
        case eREG_PWM_29:
```

**LX24132ILQ - 32 Port Display Backlight LED Controller**

```
        Set_Bits( (U16)Data, LX24132_Opr_Tx_Bits[IC_Index], OFST_REG_W_PWM29,
SIZE_REG_W_PWM29);
        break;

    case eREG_PWM_28:
        Set_Bits( (U16)Data, LX24132_Opr_Tx_Bits[IC_Index], OFST_REG_W_PWM28,
SIZE_REG_W_PWM28);
        break;

    case eREG_PWM_27:
        Set_Bits( (U16)Data, LX24132_Opr_Tx_Bits[IC_Index], OFST_REG_W_PWM27,
SIZE_REG_W_PWM27);
        break;

    case eREG_PWM_26:
        Set_Bits( (U16)Data, LX24132_Opr_Tx_Bits[IC_Index], OFST_REG_W_PWM26,
SIZE_REG_W_PWM26);
        break;

    case eREG_PWM_25:
        Set_Bits( (U16)Data, LX24132_Opr_Tx_Bits[IC_Index], OFST_REG_W_PWM25,
SIZE_REG_W_PWM25);
        break;

    case eREG_PWM_24:
        Set_Bits( (U16)Data, LX24132_Opr_Tx_Bits[IC_Index], OFST_REG_W_PWM24,
SIZE_REG_W_PWM24);
        break;

    case eREG_PWM_23:
        Set_Bits( (U16)Data, LX24132_Opr_Tx_Bits[IC_Index], OFST_REG_W_PWM23,
SIZE_REG_W_PWM23);
        break;

    case eREG_PWM_22:
        Set_Bits( (U16)Data, LX24132_Opr_Tx_Bits[IC_Index], OFST_REG_W_PWM22,
SIZE_REG_W_PWM22);
        break;

    case eREG_PWM_21:
        Set_Bits( (U16)Data, LX24132_Opr_Tx_Bits[IC_Index], OFST_REG_W_PWM21,
SIZE_REG_W_PWM21);
        break;

    case eREG_PWM_20:
        Set_Bits( (U16)Data, LX24132_Opr_Tx_Bits[IC_Index], OFST_REG_W_PWM20,
SIZE_REG_W_PWM20);
        break;

    case eREG_PWM_19:
        Set_Bits( (U16)Data, LX24132_Opr_Tx_Bits[IC_Index], OFST_REG_W_PWM19,
SIZE_REG_W_PWM19);
        break;

    case eREG_PWM_18:
        Set_Bits( (U16)Data, LX24132_Opr_Tx_Bits[IC_Index], OFST_REG_W_PWM18,
SIZE_REG_W_PWM18);
        break;

    case eREG_PWM_17:
```

**LX24132ILQ - 32 Port Display Backlight LED Controller**

```
        Set_Bits( (U16)Data, LX24132_Opr_Tx_Bits[IC_Index], OFST_REG_W_PWM17,
SIZE_REG_W_PWM17);
        break;

    case eREG_PWM_16:
        Set_Bits( (U16)Data, LX24132_Opr_Tx_Bits[IC_Index], OFST_REG_W_PWM16,
SIZE_REG_W_PWM16);
        break;

    case eREG_PWM_15:
        Set_Bits( (U16)Data, LX24132_Opr_Tx_Bits[IC_Index], OFST_REG_W_PWM15,
SIZE_REG_W_PWM15);
        break;

    case eREG_PWM_14:
        Set_Bits( (U16)Data, LX24132_Opr_Tx_Bits[IC_Index], OFST_REG_W_PWM14,
SIZE_REG_W_PWM14);
        break;
    case eREG_PWM_13:
        Set_Bits( (U16)Data, LX24132_Opr_Tx_Bits[IC_Index], OFST_REG_W_PWM13,
SIZE_REG_W_PWM13);
        break;

    case eREG_PWM_12:
        Set_Bits( (U16)Data, LX24132_Opr_Tx_Bits[IC_Index], OFST_REG_W_PWM12,
SIZE_REG_W_PWM12);
        break;

    case eREG_PWM_11:
        Set_Bits( (U16)Data, LX24132_Opr_Tx_Bits[IC_Index], OFST_REG_W_PWM11,
SIZE_REG_W_PWM11);
        break;

    case eREG_PWM_10:
        Set_Bits( (U16)Data, LX24132_Opr_Tx_Bits[IC_Index], OFST_REG_W_PWM10,
SIZE_REG_W_PWM10);
        break;

    case eREG_PWM_09:
        Set_Bits( (U16)Data, LX24132_Opr_Tx_Bits[IC_Index], OFST_REG_W_PWM09,
SIZE_REG_W_PWM09);
        break;

    case eREG_PWM_08:
        Set_Bits( (U16)Data, LX24132_Opr_Tx_Bits[IC_Index], OFST_REG_W_PWM08,
SIZE_REG_W_PWM08);
        break;

    case eREG_PWM_07:
        Set_Bits( (U16)Data, LX24132_Opr_Tx_Bits[IC_Index], OFST_REG_W_PWM07,
SIZE_REG_W_PWM07);
        break;

    case eREG_PWM_06:
        Set_Bits( (U16)Data, LX24132_Opr_Tx_Bits[IC_Index], OFST_REG_W_PWM06,
SIZE_REG_W_PWM06);
        break;

    case eREG_PWM_05:
```

**LX24132ILQ - 32 Port Display Backlight LED Controller**

```
        Set_Bits( (U16)Data, LX24132_Opr_Tx_Bits[IC_Index], OFST_REG_W_PWM05,
SIZE_REG_W_PWM05);
        break;

    case eREG_PWM_04:
        Set_Bits( (U16)Data, LX24132_Opr_Tx_Bits[IC_Index], OFST_REG_W_PWM04,
SIZE_REG_W_PWM04);
        break;

    case eREG_PWM_03:
        Set_Bits( (U16)Data, LX24132_Opr_Tx_Bits[IC_Index], OFST_REG_W_PWM03,
SIZE_REG_W_PWM03);
        break;

    case eREG_PWM_02:
        Set_Bits( (U16)Data, LX24132_Opr_Tx_Bits[IC_Index], OFST_REG_W_PWM02,
SIZE_REG_W_PWM02);
        break;

    case eREG_PWM_01:
        Set_Bits( (U16)Data, LX24132_Opr_Tx_Bits[IC_Index], OFST_REG_W_PWM01,
SIZE_REG_W_PWM01);
        break;

    case eREG_PWM_00:
        Set_Bits( (U16)Data, LX24132_Opr_Tx_Bits[IC_Index], OFST_REG_W_PWM00,
SIZE_REG_W_PWM00);
        break;

    //-- Configuration MODE registers ---
    case eREG_GLB_PRTEN: // 32 bit register so we need to handle it differently
        Set_Bits( (U16)Data      , LX24132_Cfg_Tx_Bits[IC_Index], OFST_REG_GLB_PRTEN
, 16);
        Set_Bits( (U16)(Data>>16), LX24132_Cfg_Tx_Bits[IC_Index],
OFST_REG_GLB_PRTEN+16, 16);
        break;
    case eREG_PWM_FRDIV:
        Set_Bits( (U16)Data, LX24132_Cfg_Tx_Bits[IC_Index], OFST_REG_PWM_FRDIV,
SIZE_REG_PWM_FRDIV);
        break;

    case eREG_ADC_UV:
        Set_Bits( (U16)Data, LX24132_Cfg_Tx_Bits[IC_Index], OFST_REG_ADC_UV,
SIZE_REG_ADC_UV);
        break;

    case eREG_GLB_CURR:
        Set_Bits( (U16)Data, LX24132_Cfg_Tx_Bits[IC_Index], OFST_REG_GLB_CURR,
SIZE_REG_GLB_CURR);
        break;
    case eREG_GLB_SCAN_0_15: // 32 bit register so we need to handle it differently
        Set_Bits( (U16)Data      , LX24132_Cfg_Tx_Bits[IC_Index],
OFST_REG_GLB_SCAN_0_15      , 16);
        Set_Bits( (U16)(Data>>16), LX24132_Cfg_Tx_Bits[IC_Index],
OFST_REG_GLB_SCAN_0_15 + 16, 16);
        break;

    case eREG_GLB_SCAN_16_31: // 32 bit register so we need to handle it differently
```



LX24132ILQ - 32 Port Display Backlight LED Controller

```
        Set_Bits( (U16)Data      , LX24132_Cfg_Tx_Bits[IC_Index],
OFST_REG_GLB_SCAN_16_31      , 16);
        Set_Bits( (U16)(Data>>16), LX24132_Cfg_Tx_Bits[IC_Index],
OFST_REG_GLB_SCAN_16_31 + 16, 16);
        break;

    case eREG_PWM_OFF3:
        Set_Bits( (U16)Data, LX24132_Cfg_Tx_Bits[IC_Index], OFST_REG_PWM_OFF3,
SIZE_REG_PWM_OFF3);
        break;

    case eREG_PWM_OFF2:
        Set_Bits( (U16)Data, LX24132_Cfg_Tx_Bits[IC_Index], OFST_REG_PWM_OFF2,
SIZE_REG_PWM_OFF2);
        break;

    case eREG_PWM_OFF1:
        Set_Bits( (U16)Data, LX24132_Cfg_Tx_Bits[IC_Index], OFST_REG_PWM_OFF1,
SIZE_REG_PWM_OFF1);
        break;

    case eREG_PWM_OFF0:
        Set_Bits( (U16)Data, LX24132_Cfg_Tx_Bits[IC_Index], OFST_REG_PWM_OFF0,
SIZE_REG_PWM_OFF0);
        break;

    case eREG_ADC_SKIP:
        Set_Bits( (U16)Data, LX24132_Cfg_Tx_Bits[IC_Index], OFST_REG_ADC_SKIP,
SIZE_REG_ADC_SKIP);
        break;

    case eREG_ADC_ON:
        Set_Bits( (U16)Data, LX24132_Cfg_Tx_Bits[IC_Index], OFST_REG_ADC_ON,
SIZE_REG_ADC_ON);
        break;

    case eREG_ADC_VDSTA:
        Set_Bits( (U16)Data, LX24132_Cfg_Tx_Bits[IC_Index], OFST_REG_ADC_VDSTA,
SIZE_REG_ADC_VDSTA);
        break;

    case eREG_GLB_CCR:
        Set_Bits( (U16)Data, LX24132_Cfg_Tx_Bits[IC_Index], OFST_REG_GLB_CCR,
SIZE_REG_GLB_CCR);
        break;

    case eREG_GLB_TMP_OV:
        Set_Bits( (U16)Data, LX24132_Cfg_Tx_Bits[IC_Index], OFST_REG_GLB_TMP_OV,
SIZE_REG_GLB_TMP_OV);
        break;
    }
}

/*****
* LX24132_GetReg()
*
* This function reads from various bits inside LX24132_Cfg_Rx_Bits[][] and
* LX24132_Opr_Rx_Bits[][] array according to the register which has to
```

www.Microsemi.com

LX24132ILQ

**LX24132ILQ - 32 Port Display Backlight LED Controller**

```
* be read.
*
* NOTE: Bits are read from RAM array. No Rose access over SPI
*
* Parameters:
*   U8          IC_Index: Rose IC index
*   enum eLX24132_REG eReg    : Register inside Rose (converted to specific number
*                               of bits to be updated inside long SPI write pattern
*
* Return:
*   U16: Register data value. Please note that since biggest reg width is 64 bits,
*       this parameter was adjusted to same size
*****/
U16 LX24132_GetReg(U8 IC_Index, enum eLX24132_REG eReg)
{
    switch (eReg)
    {
        //-- Configuration MODE registers ---
        case eREG_GLB_CHVER:
            return Get_Bits(LX24132_Cfg_Rx_Bits[IC_Index], OFST_REG_GLB_CHVER,
                SIZE_REG_GLB_CHVER);

        case eREG_CHIP_CONF:
            return Get_Bits(LX24132_Cfg_Rx_Bits[IC_Index], OFST_REG_CHIP_CONF,
                SIZE_REG_CHIP_CONF);

        case eREG_GLB_CURR_R:
            return Get_Bits(LX24132_Cfg_Rx_Bits[IC_Index], OFST_REG_GLB_CURR_R,
                SIZE_REG_GLB_CURR_R);

        case eREG_ADC_VDSTA_R:
            return Get_Bits(LX24132_Cfg_Rx_Bits[IC_Index], OFST_REG_ADC_VDSTA_R,
                SIZE_REG_ADC_VDSTA_R);

        case eREG_GLB_CCR_R:
            return Get_Bits(LX24132_Cfg_Rx_Bits[IC_Index], OFST_REG_GLB_CCR_R,
                SIZE_REG_GLB_CCR_R);

        case eREG_GLB_TMP_OV_R:
            return Get_Bits(LX24132_Cfg_Rx_Bits[IC_Index], OFST_REG_GLB_TMP_OV_R,
                SIZE_REG_GLB_TMP_OV_R);

        //-- Operation MODE registers ---
        case eREG_ADC2_CH_R:
            return Get_Bits(LX24132_Opr_Rx_Bits[IC_Index], OFST_REG_ADC2_CH_R,
                SIZE_REG_ADC2_CH_R);

        case eREG_ADC1_CH_R:
            return Get_Bits(LX24132_Opr_Rx_Bits[IC_Index], OFST_REG_ADC1_CH_R,
                SIZE_REG_ADC1_CH_R);

        case eREG_ADC_DATA2:
            return Get_Bits(LX24132_Opr_Rx_Bits[IC_Index], OFST_REG_ADC_DATA2,
                SIZE_REG_ADC_DATA2);

        case eREG_ADC_DATA1:
            return Get_Bits(LX24132_Opr_Rx_Bits[IC_Index], OFST_REG_ADC_DATA1,
                SIZE_REG_ADC_DATA1);
    }
}
```



LX24132ILQ - 32 Port Display Backlight LED Controller

```

    case eREG_ADC_DATA2_ERR:
        return Get_Bits(LX24132_Opr_Rx_Bits[IC_Index], OFST_REG_ADC_DATA2_ERR,
SIZE_REG_ADC_DATA2_ERR);

    case eREG_ADC_DATA1_ERR:
        return Get_Bits(LX24132_Opr_Rx_Bits[IC_Index], OFST_REG_ADC_DATA1_ERR,
SIZE_REG_ADC_DATA1_ERR);

    case eREG_GLB_ERR:
        return Get_Bits(LX24132_Opr_Rx_Bits[IC_Index], OFST_REG_GLB_ERR,
SIZE_REG_GLB_ERR);

    case eREG_ADC_MIN_VD:
        return Get_Bits(LX24132_Opr_Rx_Bits[IC_Index], OFST_REG_ADC_MIN_VD,
SIZE_REG_ADC_MIN_VD);

    case eREG_ADC_MEAS_DONE:
        return Get_Bits(LX24132_Opr_Rx_Bits[IC_Index], OFST_REG_ADC_MEAS_DONE,
SIZE_REG_ADC_MEAS_DONE);

    case eREG_ADC_DATA1_SKIP:
        return Get_Bits(LX24132_Opr_Rx_Bits[IC_Index], OFST_REG_ADC_DATA1_SKIP,
SIZE_REG_ADC_DATA1_SKIP);

    case eREG_ADC_DATA2_SKIP:
        return Get_Bits(LX24132_Opr_Rx_Bits[IC_Index], OFST_REG_ADC_DATA2_SKIP,
SIZE_REG_ADC_DATA2_SKIP);
    }

    return 0;
}

```

```

/*****
* Set_Bits()
*
* This function updates up to 16 bits inside WORD array. Bit offset is
* continues(for example Bit100). Bit location 0-15 apply to pBase[0]
*
* NOTE: limit test was omitted due to performance considerations
*
* Parameters:
* U16 Data : Data to be updated (up to 16 bit at once)
* U16 *pBase : Pointer to start address of U16[] WORD array
* U16 BitOfst: Abs bit location (offset) inside array.
* U8 NumBits: Number of bits to be updated (1-16).
*
* Return:
*****/
void Set_Bits(U16 Data, VU16 *pBase, U16 BitOfst, U8 NumBits)
{
    U32 Long;
    U16 Word_Ofst = BitOfst >> 4; // div offset by 16
to find on which WORD to start
    U16 Bit_Ofst = BitOfst - (Word_Ofst<<4); // bit offset in the
specific word which was found (from which bit to start)
    Long = (pBase[Word_Ofst+1] << 16); // Upper 16 bits
which contain the bits we need to read
    Long |= pBase[Word_Ofst]; // Lower 16 bits
which contain the bits we need to read
}

```




LX24132ILQ - 32 Port Display Backlight LED Controller

```
    U32 mask      = ((1 << NumBits) - 1) << Bit_Ofst;          // Set '1' to mast
bits. All other bits are zero
    U32 res       = (Long & (~mask)) | ((U32)Data << Bit_Ofst); // clear relevant
bits and place Data bits instead ;

    pBase[Word_Ofst] = (U16)res          ;          // Update new 16 LSB
bits
    pBase[Word_Ofst+1] = (U16)(res >> 16);        // Update new 16 MSB
bits
}

/*****
* Get_Bits()
*
* This function reads up to 16 bits from WORD array. Bit offset is
* continues(for example Bit100). Bit location 0-15 apply to pBase[0]
*
* NOTE: Limit test was omitted due to performance considerations
*
* Parameters:
*   U16 *pBase : Pointer to start address of U16[] WORD array
*   U16 BitOfst: Bit offset inside array
*   U8  NumBits: Number of bits to read (1-16). Please note that
*               limit test was omitted due to performance considerations
* Return:
*   U16 Data   : Extracted data from WORD array starting from xy bit offset
*****/
U16 Get_Bits(VU16 *pBase, U16 BitOfst, U8 NumBits)
{
    U32 Long;
    U16 Word_Ofst = BitOfst >> 4;          // div offset by 16 to find on
which WORD to start
    U16 Bit_Ofst  = BitOfst - (Word_Ofst<<4); // bit offset in the specific
word which was found (from which bit to start)
    Long          = (pBase[Word_Ofst+1] << 16); // Upper 16 bits which contain
the bits we need to read
    Long          |= pBase[Word_Ofst];        // Lower 16 bits which contain
the bits we need to read
    U32 mask      = ((1 << NumBits): 1) << Bit_Ofst; // Set '1' to mast bits. All
other bits are zero
    U32 res       = (Long & (mask)) >> Bit_Ofst ; // clear all other bits and
shift right so that result will be from bit 0

    return (U16)res;
}

/*****
* SPI_Master_TxRx()
* Send Operation/Configuration SPI bit pattern to/from LX24132
*
* Parameters:
*   enum eCOMM_MODE_LX24132 eMode: Operation/Configuration mode
*
* Parameters:
*
* Return:
*****/
void LX24132_SPI_TxRx( enum eCOMM_MODE_LX24132 eSPI_Mode)
```

**LX24132ILQ - 32 Port Display Backlight LED Controller**

```
{
    if (eSPI_Mode == eCOMM_MODE_CFG)
    {
        SPI_Master_TxRx( eCOMM_MODE_CFG,
                        (U16*)LX24132_Cfg_Tx_Bits,
                        (U16*)LX24132_Cfg_Rx_Bits,
                        MAX_NUM_LX24132_IC * LX24132_CFG_SPI_SIZE);
    }
    else
    {
        SPI_Master_TxRx( eCOMM_MODE_OPR,
                        (U16*)LX24132_Cfg_Tx_Bits,
                        (U16*)LX24132_Cfg_Rx_Bits,
                        MAX_NUM_LX24132_IC * LX24132_OPR_SPI_SIZE);
    }
}

/*****
* SPI_Master_TxRx()
*
* Read/Write to/from LX24132 by sending and reading
* <N> WORDS as SPI master
*
* Parameters:
*   enum eCOMM_MODE_LX24132 eMode: Operation/Configuration mode
*   U16 *Tx_Buf      : Tx word ptr. Tx from end to start
*   U16 *Rx_Buf      : Rx word ptr. Tx from end to start
*   U8   Num_Words  : number of words (16 bit each) to transmit/receive
*
* Parameters:
*
* Return:
*****/
void SPI_Master_TxRx( enum eCOMM_MODE_LX24132 eSPI_Mode, U16 *TxBuf, U16 *RxBuf, U16
Num_Words )
{
    USER_LX24132_SetMode_Pins(eSPI_Mode); // HW specific. For 'eCOMM_MODE_CFG' set
LX24132 MOD1 pin to '1'

    USER_Set_SPI_CS(0); // HW specific. Lower LX24132 SPI chip select to zero

    // send all SPI bits from last to 1st
    while( Num_Words )
    {
        USER_Tx_16Bit_SPI( &TxBuf[ Num_Words-1] ) ; // Send SPI Data
        RxBuf[ Num_Words-1 ] = USER_Rx_16Bit_SPI() ; // read SPI data in
        Num_Words-- ; // Dec num of words transaction
    }
    which were left
}

    USER_Set_SPI_CS(1); // HW specific. Restore LX24132 SPI chip select to high

    // If LX24132 operation mode SPI bit stream is transmitted, and there is a new
valid min FET Vd value,
    // trim LED power supply in order to get proper LED current with minimum power
dissipation
    if ( eSPI_Mode == eCOMM_MODE_OPR )
    {

```



LX24132ILQ - 32 Port Display Backlight LED Controller

```
U16 ADC_VdMinReg    ;
U8  IC, ADC_VdMin, Ch;

for (IC=0 ; IC < MAX_NUM_LX24132_IC ; IC++)
{
    if ( LX24132_GetReg(IC, eREG_ADC_MEAS_DONE) )
    {
        ADC_VdMinReg = LX24132_GetReg(IC, eREG_ADC_MIN_VD);

        if ( ADC_VdMinReg & ADC_MINVD_VALID_BIT )
        {
            ADC_VdMin = ADC_VdMinReg & 0x7F          ; // Vd min ADC reading as
reported by Rose (7 bit ADC)
            Ch          = (ADC_VdMinReg >> 8) & 0x1F; // String number as
reported by Rose (5 bit)

            Adj_LED_PowerSupply(IC, ADC_VdMin, Ch, LED_I mA,
FET_R_SENSE_OHM);
        }
    }
}

/*****
* Adj_LED_PowerSupply()
*
* This function adjusts LX24132 Power Supply trim DAC per min FET Vd report
*
* NOTE: User has to fill in the last section regarding handling
*       FET Vd voltage error
*
* Parameters:
*
* U8  IC      : Which Power Supply DAC should be used
* U8  ADC_VdMin: FET min Vd ADC reading
* U8  Ch      : LX24132 channel having min FET Vd voltage reading
* U16 I_mA    : Required LED current
* float FET_Rs : LX23108L sense resistor
*
* Return:
*****/
void Adj_LED_PowerSupply(U8 IC, U8 ADC_VdMin, U8 Ch, U16 I_mA, float FET_Rs)
{
    U16 Target_FET_Vd_mV;
    U16 FET_Min_Vd_mV    ;
    S16 Err_mV          ;
    U16 PS_DAC_Val = 700;

    // Calculate what should be FET Vd reading for the lowest channel
    Target_FET_Vd_mV = (U16)((float)I_mA*( FET_R_SENSE_OHM+FET_RDS_ON_ONH ) +
FET_VD_EXTRA_mV );

    // Convert min FET Vd ADC reading to mVolt
    FET_Min_Vd_mV = (2000 * ADC_VdMin)/128;

    Err_mV = Target_FET_Vd_mV - FET_Min_Vd_mV;

    // PS_DAC_Val = USER: calc how to trim PS_DAC per power supply respond rate,
last set value, etc
```

**LX24132ILQ - 32 Port Display Backlight LED Controller**

```
LX24132_SetReg(IC, eREG_GLB_PS_CTRL , PS_DAC_Val); // Trim DEL PS to minimize
error
}

void USER_LX24132_SetMode_Pins(enum eCOMM_MODE_LX24132 eMode)
{
    // -- Fill in your code --
}

void USER_Set_SPI_CS(U8 CS_Level)
{
    // -- Fill in your code --
}

void USER_Tx_16Bit_SPI(U16* pTx)
{
    // -- Fill in your code --
}

U16 USER_Rx_16Bit_SPI(void)
{
    // -- Fill in your code --
    return 0;
}

void USER_Wait_mSec(U16 Delay_mSec)
{
    // -- Fill in your code --
}
}
```



LX24132ILQ - 32 Port Display Backlight LED Controller

```
// -- Lets use our own variable type def --
typedef signed char S8 ;
typedef unsigned char U8 ;
typedef signed short S16 ;
typedef unsigned short U16 ;
typedef signed long S32 ;
typedef unsigned long U32 ;
typedef signed long long S64 ;
typedef unsigned long long U64 ;
typedef volatile S8 VS8 ;
typedef volatile U8 VU8 ;
typedef volatile S16 VS16 ;
typedef volatile U16 VU16 ;
typedef volatile S32 VS32 ;
typedef volatile U32 VU32 ;

//*****
//
// DEFINITIONS
//*****

// -- Bit definition of GLB_CCR register which is located at bits 216-222 inside
'cfg mode' pattern --
#define GLB_CCR_SCAN_MODE_SYNC 0 // Scanning signal will open FET
only if FET is closed
#define GLB_CCR_SCAN_MODE_ASYNC BIT0 // Scanning signal immediately
effect FET state

#define GLB_CCR_SYNC_IN_EN BIT0 // Rising edge of SYNC_IN pin will
restart PWM counter
#define GLB_CCR_SYNC_IN_DIS BIT1 // SYNC_IN pin has no effect on
PWM counters

#define GLB_CCR_SLEEP_OFF 0 // normal operation
#define GLB_CCR_DLEEP_ON BIT2 // all analog and partial digital
is in sleep mode

#define GLB_CCR_SYNC_OUT_DIS BIT3 // LX24132 SYNC_OUT pin will be
set to ZERO
#define GLB_CCR_SYNC_OUT_EN 0 // LX24132 SYNC_OUT pin per def of
bits 4&5

#define GLB_CCR_SYNC_OUT_PWM 0 // LX24132 SYNC_OUT pin will be
used to SYNC PWM
#define GLB_CCR_SYNC_OUT_ADC_CONV_START BIT4 // LX24132 SYNC_OUT pin will be
used to reflect ADC_CONV_START
#define GLB_CCR_SYNC_OUT_ADC_ON_DLY (BIT4|BIT5) // LX24132 SYNC_OUT pin
will be used to reflect ADC_ON_DLY
#define GLB_CCR_SYNC_OUT_SKIP_PORT_SAMPLE BIT5 // LX24132 SYNC_OUT pin will be
used to reflect 'skop pirt sample' internal LX24132 Signal

//----- END GLB_CCR bit definition -----
-----

// -- Bit definition of GLB_ERR bits located at bits 388-340 inside 'pwm mode' rx
pattern --
#define GLB_ERR_UDL_BIT BIT0 // 1=LX24132 reports that in one of
the four LX23108, one of the ports is below Vdrain threshold
```



LX24132ILQ - 32 Port Display Backlight LED Controller

```
#define GLB_ERR_OVR_VOLT_OVER_TEMP_BIT      BIT1 // 1=LX24132 reports that in one of
the four LX23108, one of the ports is over heated and shut down, or FET Vd voltage
is to high (around 7V)
#define GLB_ERR_SKIP_BIT                    BIT2 // 1=LX24132 reports that in one of
the four LX23108, one of the ports FET Vd measurement was skipped since PWM was to
low
```

**LX24132ILQ - 32 Port Display Backlight LED Controller**

```
// -- LX24132 Configuration Mode Write Registers location & size --
#define OFST_REG_GLB_PRTEN 0
#define SIZE_REG_GLB_PRTEN 32
#define OFST_REG_PWM_FRDIV 32
#define SIZE_REG_PWM_FRDIV 5
#define OFST_REG_ADC_UV 40
#define SIZE_REG_ADC_UV 7
#define OFST_REG_GLB_CURR 52
#define SIZE_REG_GLB_CURR 8
#define OFST_REG_GLB_SCAN_0_15 64
#define SIZE_REG_GLB_SCAN_0_15 32
#define OFST_REG_GLB_SCAN_16_31 96
#define SIZE_REG_GLB_SCAN_16_31 32
#define OFST_REG_PWM_OFF3 128
#define SIZE_REG_PWM_OFF3 12
#define OFST_REG_PWM_OFF2 140
#define SIZE_REG_PWM_OFF2 12
#define OFST_REG_PWM_OFF1 152
#define SIZE_REG_PWM_OFF1 12
#define OFST_REG_PWM_OFF0 164
#define SIZE_REG_PWM_OFF0 12
#define OFST_REG_ADC_SKIP 180
#define SIZE_REG_ADC_SKIP 12
#define OFST_REG_ADC_ON 196
#define SIZE_REG_ADC_ON 12
#define OFST_REG_ADC_VDSTA 208
#define SIZE_REG_ADC_VDSTA 8
#define OFST_REG_GLB_CCR 216
#define SIZE_REG_GLB_CCR 7
#define OFST_REG_GLB_TMP_OV 223
#define SIZE_REG_GLB_TMP_OV 1

// -- LX24132 Configuration Mode Read Registers location & size --
#define OFST_REG_GLB_CHVER 168
#define SIZE_REG_GLB_CHVER 16
#define OFST_REG_CHIP_CONF 184
#define SIZE_REG_CHIP_CONF 2
#define OFST_REG_GLB_CURR_R 188
#define SIZE_REG_GLB_CURR_R 8
#define OFST_REG_ADC_VDSTA_R 200
#define SIZE_REG_ADC_VDSTA_R 8
#define OFST_REG_GLB_CCR_R 208
#define SIZE_REG_GLB_CCR_R 7
#define OFST_REG_GLB_TMP_OV_R 215
#define SIZE_REG_GLB_TMP_OV_R 1

// -- LX24132 Operation Mode Write Registers location & size --
#define OFST_REG_ADC_DATA2_CH 0
#define SIZE_REG_ADC_DATA2_CH 6
#define OFST_REG_ADC_DATA1_CH 8
#define SIZE_REG_ADC_DATA1_CH 6
#define OFST_REG_GLB_PS_CTRL 16
#define SIZE_REG_GLB_PS_CTRL 10
#define OFST_REG_ADC_RFVDMIN 28
#define SIZE_REG_ADC_RFVDMIN 1
#define OFST_REG_GLB_CLR_ERR 29
#define SIZE_REG_GLB_CLR_ERR 1
#define OFST_REG_W_PWM31 32
#define SIZE_REG_W_PWM31 12
```



LX24132ILQ - 32 Port Display Backlight LED Controller

```
#define OFST_REG_W_PWM30 44
#define SIZE_REG_W_PWM30 12
#define OFST_REG_W_PWM29 56
#define SIZE_REG_W_PWM29 12
#define OFST_REG_W_PWM28 68
#define SIZE_REG_W_PWM28 12
#define OFST_REG_W_PWM27 80
#define SIZE_REG_W_PWM27 12
#define OFST_REG_W_PWM26 92
#define SIZE_REG_W_PWM26 12
#define OFST_REG_W_PWM25 104
#define SIZE_REG_W_PWM25 12
#define OFST_REG_W_PWM24 116
#define SIZE_REG_W_PWM24 12
#define OFST_REG_W_PWM23 128
#define SIZE_REG_W_PWM23 12
#define OFST_REG_W_PWM22 140
#define SIZE_REG_W_PWM22 12
#define OFST_REG_W_PWM21 152
#define SIZE_REG_W_PWM21 12
#define OFST_REG_W_PWM20 164
#define SIZE_REG_W_PWM20 12
#define OFST_REG_W_PWM19 176
#define SIZE_REG_W_PWM19 12
#define OFST_REG_W_PWM18 188
#define SIZE_REG_W_PWM18 12
#define OFST_REG_W_PWM17 200
#define SIZE_REG_W_PWM17 12
#define OFST_REG_W_PWM16 212
#define SIZE_REG_W_PWM16 12
#define OFST_REG_W_PWM15 224
#define SIZE_REG_W_PWM15 12
#define OFST_REG_W_PWM14 236
#define SIZE_REG_W_PWM14 12
#define OFST_REG_W_PWM13 248
#define SIZE_REG_W_PWM13 12
#define OFST_REG_W_PWM12 260
#define SIZE_REG_W_PWM12 12
#define OFST_REG_W_PWM11 272
#define SIZE_REG_W_PWM11 12
#define OFST_REG_W_PWM10 284
#define SIZE_REG_W_PWM10 12
#define OFST_REG_W_PWM09 296
#define SIZE_REG_W_PWM09 12
#define OFST_REG_W_PWM08 308
#define SIZE_REG_W_PWM08 12
#define OFST_REG_W_PWM07 320
#define SIZE_REG_W_PWM07 12
#define OFST_REG_W_PWM06 332
#define SIZE_REG_W_PWM06 12
#define OFST_REG_W_PWM05 344
#define SIZE_REG_W_PWM05 12
#define OFST_REG_W_PWM04 356
#define SIZE_REG_W_PWM04 12
#define OFST_REG_W_PWM03 368
#define SIZE_REG_W_PWM03 12
#define OFST_REG_W_PWM02 380
#define SIZE_REG_W_PWM02 12
#define OFST_REG_W_PWM01 392
```


**LX24132ILQ - 32 Port Display Backlight LED Controller**

```
#define SIZE_REG_W_PWM01          12
#define OFST_REG_W_PWM00          404
#define SIZE_REG_W_PWM00          12

// -- LX24132 Operation Mode Read Registers location & size --
#define OFST_REG_ADC2_CH_R        352
#define SIZE_REG_ADC2_CH_R        6
#define OFST_REG_ADC1_CH_R        360
#define SIZE_REG_ADC1_CH_R        6
#define OFST_REG_ADC_DATA2        368
#define SIZE_REG_ADC_DATA2        8
#define OFST_REG_ADC_DATA1        376
#define SIZE_REG_ADC_DATA1        8
#define OFST_REG_ADC_DATA2_ERR    384
#define SIZE_REG_ADC_DATA2_ERR    2
#define OFST_REG_ADC_DATA1_ERR    386
#define SIZE_REG_ADC_DATA1_ERR    2
#define OFST_REG_GLB_ERR          388
#define SIZE_REG_GLB_ERR          3
#define OFST_REG_ADC_MIN_VD        392
#define SIZE_REG_ADC_MIN_VD        13
#define OFST_REG_ADC_MEAS_DONE     405
#define SIZE_REG_ADC_MEAS_DONE     1
#define OFST_REG_ADC_DATA1_SKIP    406
#define SIZE_REG_ADC_DATA1_SKIP    1
#define OFST_REG_ADC_DATA2_SKIP    407
#define SIZE_REG_ADC_DATA2_SKIP    1

enum eLX24132_REG
{
// -- LX24132 Operation Mode Write Registers --
eREG_PWM_00 = 0 , // MUST be organized from PWM0-PWM31 !!!!
eREG_PWM_01 ,
eREG_PWM_02 ,
eREG_PWM_03 ,
eREG_PWM_04 ,
eREG_PWM_05 ,
eREG_PWM_06 ,
eREG_PWM_07 ,
eREG_PWM_08 ,
eREG_PWM_09 ,
eREG_PWM_10 ,
eREG_PWM_11 ,
eREG_PWM_12 ,
eREG_PWM_13 ,
eREG_PWM_14 ,
eREG_PWM_15 ,
eREG_PWM_16 ,
eREG_PWM_17 ,
eREG_PWM_18 ,
eREG_PWM_19 ,
eREG_PWM_20 ,
eREG_PWM_21 ,
eREG_PWM_22 ,
eREG_PWM_23 ,
eREG_PWM_24 ,
eREG_PWM_25 ,
eREG_PWM_26 ,
eREG_PWM_27 ,
```



LX24132ILQ - 32 Port Display Backlight LED Controller

```
eREG_PWM_28      ,
eREG_PWM_29      ,
eREG_PWM_30      ,
eREG_PWM_31      ,
eREG_ADC1_CH     ,
eREG_ADC2_CH     ,
eREG_GLB_PS_CTRL ,
eREG_ADC_RFVDMIN ,
eREG_GLB_CLR_ERR ,

// -- LX24132 Operation Mode Read Registers --
eREG_ADC1_CH_R   ,
eREG_ADC2_CH_R   ,
eREG_ADC_DATA1   ,
eREG_ADC_DATA2   ,
eREG_ADC_DATA1_ERR ,
eREG_ADC_DATA2_ERR ,
eREG_GLB_ERR     ,
eREG_ADC_MIN_VD  ,
eREG_ADC_MEAS_DONE ,
eREG_ADC_DATA1_SKIP ,
eREG_ADC_DATA2_SKIP ,

// -- LX24132 Configuration Mode Write Registers --
eREG_GLB_PRTEN   ,
eREG_PWM_FRDIV   ,
eREG_ADC_UV      ,
eREG_GLB_CURR    ,
eREG_GLB_SCAN_0_15 ,
eREG_GLB_SCAN_16_31 ,
eREG_PWM_OFF0    ,
eREG_PWM_OFF1    ,
eREG_PWM_OFF2    ,
eREG_PWM_OFF3    ,
eREG_ADC_SKIP    ,
eREG_ADC_ON      ,
eREG_ADC_VDSTA   ,
eREG_GLB_CCR     ,
eREG_GLB_TMP_OV  ,
// -- LX24132 Configuration Mode Read Registers --
eREG_GLB_CHVER   ,
eREG_CHIP_CONF   ,
eREG_GLB_CURR_R  ,
eREG_ADC_VDSTA_R ,
eREG_GLB_CCR_R   ,
eREG_GLB_TMP_OV_R ,
};
enum eCOMM_MODE_LX24132
{
    eCOMM_MODE_OPR = 0, // LX24132 SPI configuration mode
    eCOMM_MODE_CFG = 1 // LX24132 SPI operation mode
};

// ----- Bit definition for common in use registers -----
#define ADC_MINVD_VALID_BIT    0x80

// ----- END -----
```



LX24132ILQ - 32 Port Display Backlight LED Controller

```
#define LX24132_OPR_SPI_SIZE 26 // 26x16=416 bit - LX24132 SPI PWM Tx/Rx pattern
#define LX24132_CFG_SPI_SIZE 14 // 14x16=224 bit - LX24132 SPI ADV Tx/Rx pattern

#define LX24132_SYS_CLK_KHZ 7864 // LX24132 system clock in KHz
#define LX24132_SYS_CLK_DIV 15 // Div 7.864Mhz by 16 and by 4096 = 120Hz
#define FET_R_SENSE_OHM 10 // LX23108L sense resistors connected to S0-S7
#define FET_RDS_ON_OHM 2.6 // LX23108L FET RDS-ON in ohm
#define LED_ImA 30 // LED current = 30mA
#define FET_VD_EXTRA_mV 150 // Extra voltage above required FET Vd voltage
#define ADC_SKIP_USEC 150 // Don't measure 150 uSec from PWM-ON
#define ADC_ON_USEC 100 // Time allocated for ADC measurements

#define ADC_CONV_FROM_uSEC(t) (U16) (( (float)LX24132_SYS_CLK_KHZ*t ) / (
1000*(LX24132_SYS_CLK_DIV+1) )+0.5)
#define VDSTA_CONV_FROM_uSEC(t) (U16) (( (float)LX24132_SYS_CLK_KHZ*t )/1000+0.5)

/*****
* GLOBAL DATA
*****/
#define MAX_NUM_LX24132_IC 4 // Number of LX24132 IC's in this example

#pragma pack(1)
VU16 LX24132_Opr_Tx_Bits[ MAX_NUM_LX24132_IC ][ LX24132_OPR_SPI_SIZE ]; //
26x16=416 bit - LX24132 SPI PWM Tx pattern
VU16 LX24132_Opr_Rx_Bits[ MAX_NUM_LX24132_IC ][ LX24132_OPR_SPI_SIZE ]; //
26x16=416 bit - LX24132 SPI PWM Rx Pattern

VU16 LX24132_Cfg_Tx_Bits[ MAX_NUM_LX24132_IC ][ LX24132_CFG_SPI_SIZE ]; //
14x16=224 bit - LX24132 SPI ADV Tx pattern
VU16 LX24132_Cfg_Rx_Bits[ MAX_NUM_LX24132_IC ][ LX24132_CFG_SPI_SIZE ]; //
14x16=224 bit - LX24132 SPI ADV Rx pattern
#pragma pack()

/*****
* EXPORTED FUNCTIONS PROTOTYPES
*****/
void LX24132_SetReg(U8 IC_Index, enum eLX24132_REG eReg, U32 Data);
U16 LX24132_GetReg(U8 IC_Index, enum eLX24132_REG eReg);
void Set_Bits(U16 Data, VU16 *pBase, U16 BitOfst, U8 NumBits);
U16 Get_Bits(VU16 *pBase, U16 BitOfst, U8 NumBits);
void lx24132_SetDef_Cfg_ModeBits_AllIC( U16 I_ma);
void lx24132_SetDef_Opr_ModeBits_AllIC(void);
void SPI_Master_TxRx( enum eCOMM_MODE_LX24132 eSPI_Mode, U16 *TxBuf, U16 *RxBuf,
U16 Num_Words);
void Adj_LED_PowerSupply(U8 IC, U8 ADC_VdMin, U8 Ch, U16 I_mA, float FET_Rs);
void LX24132_SPI_TxRx( enum eCOMM_MODE_LX24132 eSPI_Mode);
void USER_LX24132_SetMode_Pins(enum eCOMM_MODE_LX24132 eMode);
void USER_Set_SPI_CS(U8 CS_Level);
void USER_Tx_16Bit_SPI(U16* pTx);
U16 USER_Rx_16Bit_SPI(void);
void USER_Wait_mSec(U16 Delay_mSec);
```



LX24132ILQ - 32 Port Display Backlight LED Controller

The information contained in the document is PROPRIETARY AND CONFIDENTIAL information of Microsemi and cannot be copied, published, uploaded, posted, transmitted, distributed or disclosed or used without the express duly signed written consent of Microsemi...

Microsemi reserves the right to change the configuration, functionality and performance of its products at anytime without any notice. This product has been subject to limited testing and should not be used in conjunction with life-support or other mission-critical equipment or applications.

Revision History

Table with 3 columns: Revision Level / Date, Para. Affected, Description. Rows include: 0.1 / 17 December. 08, 0.2 / 28 December. 08, 0.3 / 26 February. 09, 1.0 / 07 May.09, 1.2/29 May 09, 1.3 / June 09, 1.4 / 05 March 2010, 1.5 / 7 June 2010, 1.5 / 27 Jul 2010, 1.5A/ 12Oct 2010.

© 2010 Microsemi Corp.

All rights reserved.

For support contact: sales_AMSG@microsemi.com

Visit our web site at: www.microsemi.com

Catalogue Number DS_LX24132