AN_INLC_04

April 2020
Revision 1.0

ISELED Communication Protocol - Control Commands

1. Introduction

This document describes the ISELED communication protocol command set and their implementation in the application programming interface (API).

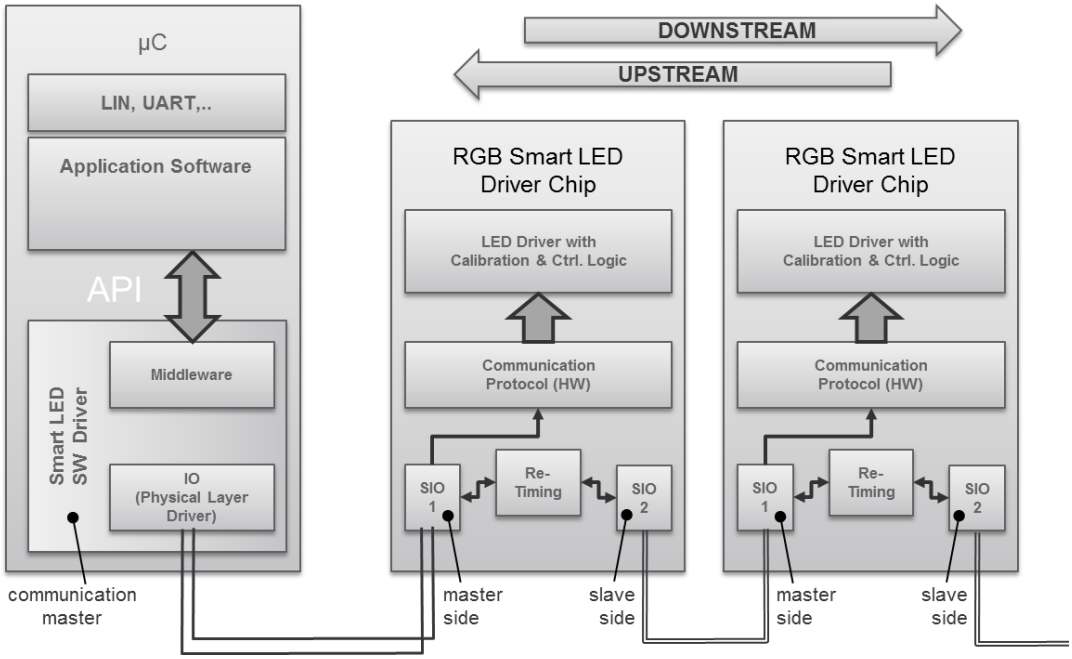


Figure 1 - ISELED serial communication

The ISELED communication protocol implements a half-duplex, bidirectional, high speed serial master-slave communication between a LED strip controller unit and up to 4079 ISELEDs.

The attachment to the adjacent devices in the chain is made up by two bidirectional differential serial communication lines. The direction towards the controlling microcontroller device is referred to as the “upstream” connection. The opposite direction towards the end of the chain is the “downstream” link. Both links are controlled by the communication unit. Incoming command frames from upstream and responses from downstream are passed to the main unit which is responsible for command processing and overall device control. Commands always originate from the controlling microcontroller. The microcontroller is referred to as the “host” in this document.

The gross data rate on the serial line is 2Mbit/s, i.e. each bit has a nominal duration of 500 ns. As the on-die oscillator has a very limited accuracy, the actual bit time may vary significantly. The whole system is designed for a maximum oscillator variance of ±30%. With the nominal oscillator frequency being 16 MHz, the actual frequency range is 11.2-20.8MHz.

The device directly attached to the host does not use the differential line mode on the upstream side. Instead a single-ended line mode with open-drain interface is used. The single-ended mode is intended to allow for an easy attachment to industry standard microcontrollers. Both single-ended lines require an external pull-up at the microcontroller to 5V.

During start-up the master interface detects single-ended or differential communication and enables termination to GND in case of differential mode. The slave interface operates differentially except for sleep mode. During initialization the slave interface physically checks for cable disconnects, respectively end of chain. In normal operation the ISELED software driver can regularly check for cable disconnects.

The protocol provides a set of control commands which are run length coded (RLC) and embedded in a serial frame structure.

List of Abbreviations and Acronyms

ADC	Analog Digital Converter
API	Application Programming Interface
BG	Bandgap
CRC	Cyclic Redundancy Check
DAC	Digital Analog Converter
DWL	Dominant Wavelength
ISELED	Integrated Smart Ecosystem Light Emitting Diode
LDO	Low Dropout
LED	Light Emitting Diode
LUT	Lookup Table
PWM	Pulse Width Modulation
RLC	Run Length Code
TC	Temperature Compensation

List of Symbols

Symbol	Description	Size
ADC_{5V_PRG}	ADC value of 5V_PRG pin voltage	9 bit
ADC_{BG}	ADC value of bandgap voltage	9 bit
ADC_{Blue}	ADC value of blue LED cathode voltage	9 bit
ADC_{Green}	ADC value of green LED cathode voltage	9 bit
ADC_{LDO}	ADC value digital 1.5V supply voltage	9 bit
ADC_{Red}	ADC value of red LED cathode voltage	9 bit
ADT_{Temp}	ADC value of chip temperature	9 bit
PWM_{Max}	Maximal PWM value	12 bit
RGB	Color intensity value of red, green and blue LED	3 x 8 bit
TC_{Base}	Temperature compensation lookup table base value	9 bit
TC_{Offset}	Temperature compensation lookup table offset value	9 bit

Contents

Index of Figures

Index of Tables

2. Data Types

2.1. digLED_InitType

Type Name	digLED_InitType		
Type	structure		
Object	uint16_t	firstLEDAdr	Address of the first LED in the chain. Has to be >=1
	Bool	crcEnable	0: CRC disabled, 1: CRC enabled
	Bool	tempCmpEnable	0: TC disabled, 1: TC enabled
	Bool	voltSwing	Must be set to zero
	Bool	phaseShift	0: Phase Shift enabled, 1: Phase Shift disabled
Description	A pointer to an instance of this structure will be used in the initialization of the whole chain of Smart LED drivers.		

Table 1 - digLED_InitType API specification

2.2. digLED_ConfigType

Type Name	digLED_ConfigType		
Type	structure		
Object	uint8_t	nrOfStrips	Number of LED strips assigned on this interface
	void(*callback)	(digLED_CommEventType event, digLED_StripNumberType strip)	Callback function that notifies the user about events on LED strips
	digLED_PinConfigType	*pinConfig	Pointer to the structure containing the initialization values for physical pins configuration
Description	A pointer to an instance of this structure, used to initialize the Smart LED drivers and shared resources used for the Smart LED driver. Structure might contain additional vendor-specific configuration options.		

Table 2 - digLED_ConfigType API specification

2.3. digLED_PinConfigType

Type Name	digLED_PinConfigType		
Type	structure		
Object	uint8_t	dataPin	Instance of I/O pin used for data line
	uint8_t	clockPin	Instance of I/O pin used for clock line
Description	A pointer to an instance of this structure, used to initialize the pins used for communication with the Smart LEDs. Structure might contain additional vendor-specific items.		

Table 3 - digLED_PinConfigType

2.4. digLED_ReadDataResultType

Type Name	digLED_ReadDataResultType		
Type	structure		
Object	uint16_t	chainLength	This parameter is bidirectional. Utillized as an input by the application during initialization of the strip or for read access, it shall hold the number of expected LED responses. Utilized as an output by the driver, it will be populated with the number of LEDs found on the strip, based on address calculation. 0:4079 Range of values of this parameter
	uint16_t	retData	Pointer to an array that will hold the status values of each smart LED driver of the chain. It is populated by the driver after initialization of the chain or after a read access of the LEDs.
Description	A pointer to an instance of this structure will be used in the initialization of the whole chain or for read access of Smart LED drivers to store the returned values of the Init or read process		

Table 4 - digLED_ReadDataResultType API specification

2.5. digLED_ReturnType

Type Name	digLED_ReturnType	
Type	enum	
Object	DIGLED_ERROR	Function returns with an ERROR.
	DIGLED_OK	Function returns successful.
	DIGLED_BUSY	On-going function.
Description	Error code of a function call.	

Table 5 - digLED_ReturnType API specification

2.6. digLED_SendCmdBlockType

Type Name	digLED_SendCmdBlockType		
Type	structure		
Object	uint8_t	red	Set_RGB 0-255: Set the intensity of the red LED Set_DIM Amounts of bits to shift (0...3) the red PWM value
	uint8_t	green	Set_RGB 0-255: Set the intensity of the green LED Set_DIM Amounts of bits to shift (0…3) the green PWM value
	uint8_t	blue	Set_RGB 0-255: Set the intensity of the blue LED Set_DIM Amounts of bits to shift (0…3) the blue PWM value
	uint16_t	addr	Address of the LED
	uint8_t	cmd	Command to be sent. 0x1 for Set_RGB, 0x2 for Set_DIM
	uint8_t	padding[10U]	Additional 10 bytes needed for command encoding for transmission.
Description	A pointer to an instance of this structure will be used in the construction of a command for a single ISELED. The command will be sent as part of a block to multiple ISELEDs.		

Table 6 - digLED_SendCmdBlockType API specification

3. Control Commands

3.1. Initialization Functions

The Initialization of the ISELED driver is split into two functions, digLED_Init_Interface and digLED_Init_Strip.

3.1.1. digLED_Init_Interface

The digLED_Init_Interface function initializes the shared resources used for communication with ISELED devices. It does not issue any communication on the ISELED communication channel. After reset or power up, the digLED_Init_Interface function has to be called before any other control command is called.

Functional Call	**digLED_Init_Interface**	
Syntax	digLED_ReturnType digLED_Init_Interface (uint8_t NoOfInterfaces, digLED_ConfigType* ConfigPtr)	
Parameters (in)	NoOfInterfaces	Number of hardware interfaces used for ISELED communication.
	ConfigPtr	Pointer to the configuration structure.
Parameters (out)	None	
Return value	digLED_ReturnType	The function checks its parameters and returns with DIGLED_OK if the pointers to the structs are not NULL otherwise with DIGLED_ERROR. While executing the command, the return value is DIGLED_BUSY.

Table 7 - digLED_Init_Interface API specification

3.1.2. digLED_Init_Strip

The digLED_Init_Strip command initializes a particular ISELED chain by issuing the command on an associated ISELED communication channel.

This command is always the first command to be transmitted after power-up or after the digLED_Reset command. The command initializes a chain of devices by assigning the address of the device and by en- or disabling the phaseshift, the CRC and temperature compensation functions. The INIT command is always executed with a CRC checksum. This is true for both, the command and the response frame.

If any command is received by a device before initialization, the command is always considered as illegal and the error status bit for an undefined command is set. This may happen in the chain's first device only, as a non-initialized device does not forward received messages.

If the first device in the chain receives an Init command, it takes the received address as its own device address and afterwards transmits another Init frame to the next device in the chain. It increments the address before the transmission. As the adjacent devices proceed in the same manner, the devices in the chain get enumerated with ascending addresses. When the final device in the chain recognizes there is no receiving device at its downstream link, it transmits a response frame upstream. The response frame to a digLED_Init_Strip command carries the configuration word read from the OTP. It also transmits the own devices address just initialized.

All upstream devices wait for the responses to be received and forward them towards the microcontroller. If a frame with an address equal to the adjacent device address (own address plus one) is received, the own response to the digLED_Init command is transmitted thereafter. If the first device has transmitted its response frame, the chain is ready to process regular commands (non-Init frames).

As soon as a device is initialized, it unconditionally forwards incoming correct frames (Frame-Sync, Freq-Sync and the RLC coding as well as the frame length are checked) to the adjacent node in the chain.

Frames received from upstream are forwarded downstream and vice versa. If an error is detected the forwarding is stopped for this frame.

Functional Call	**digLED_Init_Strip**	
Syntax	digLED_ReturnType digLED_Init_Strip (digLED_InitType* ChainInitPtr, digLED_ReadDataResultType* ChainInitResultPtr, uint8_t StripNr)	
Parameters (in)	ChainInitPtr	Pointer to structure containing the initialization values of the LED chain.
	StripNr	Strip number.
Parameters (out)	ChainInitResultPtr	Pointer to structure containing the result values of the LED chain initialization process.
Return value	digLED_ReturnType	The function checks its parameters and returns with DIGLED_OK if the pointers to the structs are not NULL otherwise with DIGLED_ERROR. While executing the command, the return value is DIGLED_BUSY.

Table 8 - digLED_Init_Strip API specification

3.2. Write Commands

Most commands of the LED controller are write-only commands. I.e. the devices receive a command frame and execute the appropriate actions without any further communication. A write access command may be directed to a single device (unicast), to all devices (broadcast), or to a defined group of devices (multicast). As every command frame is forwarded downstream irrespective of its destination address, all stations always receive all commands. Only its execution depends on the command's destination address. To avoid communication issues, it is required to wait 30% of the command length between two consecutive commands.

3.2.1. digLED_Define_Mcast

The digLED_Define_Mcast command defines the multicast group membership for an individual device. There are 16 multicast address groups. Each device may be member in any number of these groups. The group membership is defined as a bit vector (s.Table 9). Each bit represents the membership in one group. If the addressed device is to be member of a group, the corresponding bit must be set. To define a device' membership in all 16 groups, this command has to be issued twice. One time for the lower 8 and another time for the upper 8 address groups. There is no response to this command.

The example in Table 11 shows how to define the group membership of 3 LEDs.

Bit Vector									Description
MSB	7	6	5	4	3	2	1	LSB	u = 0: Define lower 8 groups u = 1: Define upper 8 groups g = 0: Do not join group g = 1: Join group
u	g	g	g	g	g	g	g	g	

Table 9 - digLED_Define_Mcast bit vector

Functional Call	**digLED_Define_Mcast**	
Syntax	digLED_ReturnType digLED_Define_Mcast(uint16_t Param, uint16_t Address, uint16_t StripNr)	
Parameters (in)	Param	0-511: Group membership
	Address	0-4079: Address of the target LED. 0 addresses all LEDs of the chain.
	StripNr	Strip number.
Parameters (out)	None	
Return value	digLED_ReturnType	Function returns DIGLED_OK in case of successful function call and DIGLED_ERROR in case of error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

Table 10 - digLED_Define_Mcast API Specification

API Code Example			
...			
digLED_Define_Mcast(0x055, 0x001, 1);	//address 1;	join group 1,3,5,7;	strip number 1
digLED_Define_Mcast(0x155, 0x001, 1);	//address 1;	join group 9,11,13,15;	strip number 1
digLED_Define_Mcast(0x0ff, 0x002, 1);	//address 2;	join group 1,2,3,4,5,6,7,8;	strip number 1
digLED_Define_Mcast(0x1ff, 0x002, 1);	//address 2;	join group 9,10,11,12,13,14,15,16;	strip number 1
digLED_Define_Mcast(0x0aa, 0x003, 1);	//address 3;	join group 2,4,6,8;	strip number 1
digLED_Define_Mcast(0x1aa, 0x003, 1);	//address 3;	join group 10,12,14,16;	strip number 1
…			

Table 11 - digLED_Define_Mcast code example

3.2.2. digLED_Reset

The digLED_Reset command re-initializes the communication links. It is intended for error recovery, if e.g. a part of the chain has temporarily been disconnected. This command resets the communication link state back to its state after power-up or a hardware reset. The remaining internal device state is untouched by this command. E.g. the intensities of the LEDs do not change.

The digLED_Reset command must be followed by another digLED_Init_Strip command to allow for further communication. The CRC checksum is re-enabled as after power-up or a hardware reset.

The digLED_Reset command does not transmit any response.

Functional Call	**digLED_Reset**	
Syntax	digLED_ReturnType digLED_Reset(uint8_t StripNr)	
Parameters (in)	StripNr	Strip number.
Parameters (out)	None	
Return value	digLED_ReturnType	Function returns DIGLED_OK in case of successful function call and DIGLED_ERROR in case of error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

Table 12 - digLED_Reset API specification

3.2.3. digLED_Set_RGB

The digLED_Set_RGB command is used to control intensity of an LED device. The resolution for each colour is 8 bit. The PWM channels for the three LEDs are updated independently, i.e. the temporary colour error caused by this command is kept at the possible minimum.

There is no response to this command.

Functional Call	**digLED_Set_RGB**	
Syntax	digLED_ReturnType digLED_Set_RGB (uint8_t Red, uint8_t Green, uint8_t Blue, uint16_t Address, uint8_t StripNr)	
Parameters (in)	Red	0-255: Sets the intensity of the red LED
	Green	0-255: Sets the intensity of the green LED
	Blue	0-255: Sets the intensity of the blue LED
	Address	0-4095: Address of the target LED. 0 addresses all LEDs of the chain. 4080-4095 addresses multicast group.
	StripNr	Strip number.
Parameters (out)	None	
Return value	digLED_ReturnType	Function returns DIGLED_OK in case of successful function call and DIGLED_ERROR in case of error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

Table 13 - digLED_Set_RGB API specification

3.2.4. digLED_Set_Dim

The digLED_Set_Dim command is used to extend the digLED_Set_RGB command's resolution with low LED intensities. The PWM duty cycles computed from the RGB setting are scaled depending on the parameters given to this command. There are four values available for the scaling (s. Figure 2). The correct command sequence to select an LED intensity is to first issue a digLED_Set_Dim command followed by a digLED_Set_RGB command. The digLED_Set_Dim scaling is applied not before the digLED_Set_RGB command is received. The digLED_Set_Dim command alone has no visible effect.

There is no response to this command.

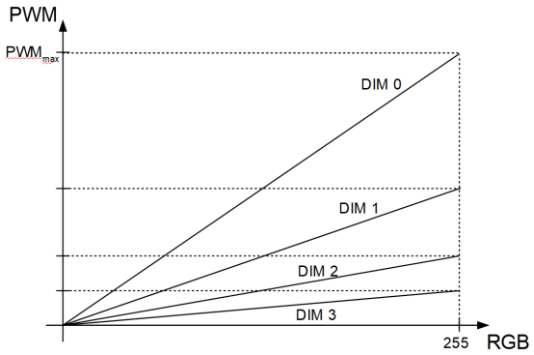


Figure 2 - PWM over RGB

Functional Call	**digLED_Set_Dim**	
Syntax	digLED_ReturnType digLED_Set_Dim (uint8_t Red, uint8_t Green, uint8_t Blue, uint16_t Address, uint8_t StripNr)	
Parameters (in)	Red	amount of bits to shift (0..3) the red LED PWM value
	Green	amount of bits to shift (0..3) the green LED PWM value
	Blue	amount of bits to shift (0..3) the blue LED PWM value
	Address	0-4095: Address of the target LED. 0 addresses all LEDs of the chain. 4080-4095 addresses multicast group.
	StripNr	Strip number.
Parameters (out)	None	
Return value	digLED_ReturnType	Function returns DIGLED_OK in case of successful function call and DIGLED_ERROR in case of error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

Table 14 - digLED_Set_Dim API specification

3.2.5. digLED_Set_PWM_Red/Green/Blue

These commands set the maximum PWM values for the respective LED channel. They are intended for white-point calibration. The maximum PWM values are stored in volatile registers which may be burned into the OTP memory. The registers may in turn be initialized from the OTP at device startup.

There is no response to these commands.

Functional Call	**digLED_Set_PWM_Red**	
Syntax	digLED_ReturnType digLED_Set_PWM_Red(uint16_t Param, uint16_t Address, uint8_t StripNr)	
Parameters (in)	Param	0-4095: PWM_{Max} red LED
	Address	0-4095: Address of the target LED. 0 addresses all LEDs of the chain. 4080-4095 addresses multicast group.
	StripNr	Strip number.
Parameters (out)	None	
Return value	digLED_ReturnType	Function returns DIGLED_OK in case of successful function call and DIGLED_ERROR in case of error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

Table 15 - digLED_Set_PWM_Red API specification

Functional Call	digLED_Set_PWM_Green	
Syntax	digLED_ReturnType digLED_Set_PWM_Green(uint16_t Param, uint16_t Address, uint8_t StripNr)	
Parameters (in)	Param	0-4095: PWM_{max} green LED
	Address	0-4095: Address of the target LED. 0 addresses all LEDs of the chain. 4080-4095 addresses multicast group.
	StripNr	Strip number.
Parameters (out)	None	
Return value	digLED_ReturnType	Function returns DIGLED_OK in case of successful function call and DIGLED_ERROR in case of error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

Table 16 - digLED_Set_PWM_Green API specification

Functional Call	digLED_Set_PWM_Blue	
Syntax	digLED_ReturnType digLED_Set_PWM_Blue(uint16_t Param, uint16_t Address, uint8_t StripNr)	
Parameters (in)	Param	0-4095: PWM_{Max} blue LED
	Address	0-4095: Address of the target LED. 0 addresses all LEDs of the chain. 4080-4095 addresses multicast group.
	StripNr	Strip number.
Parameters (out)	None	
Return value	digLED_ReturnType	Function returns DIGLED_OK in case of successful function call and DIGLED_ERROR in case of error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

Table 17 - digLED_Set_PWM_Blue API specification

3.2.6. digLED_Set_Cur_Green/Blue

These commands set the peak current for the green or the blue LED respectively. They are intended to calibrate the LED's dominant wave length (DWL) before the white-point is calibrated. The peak current values are stored in volatile registers which may be burned into the OTP memory. The registers may in turn be initialized from the OTP at device startup. When one of these commands is issued, all PWM channels are turned off and then a one-shot pulse is generated for the green or blue channel respectively. The duration of the pulse is nominal 25ms with a tolerance of ±30%.

There is no response to these commands.

Functional Call	**digLED_Set_Cur_Green**	
Syntax	digLED_ReturnType digLED_Set_Cur_Green(uint8_t Param, uint16_t Address, uint8_t StripNr)	
Parameters (in)	Param	0-15: Peak current value
	Address	0-4095: Address of the target LED. 0 addresses all LEDs of the chain. 4080-4095 addresses multicast group.
	StripNr	Strip number.
Parameters (out)	None	
Return value	digLED_ReturnType	Function returns DIGLED_OK in case of successful function call and DIGLED_ERROR in case of error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

Table 18 - digLED_Set_Cur_Green API specification

Functional Call	digLED_Set_Cur_Blue	
Syntax	digLED_ReturnType digLED_Set_Cur_Blue(uint8_t Param, uint16_t Address, uint8_t StripNr)	
Parameters (in)	Param	0-15: Peak current value
	Address	0-4095: Address of the target LED. 0 addresses all LEDs of the chain. 4080-4095 addresses multicast group.
	StripNr	Strip number.
Parameters (out)	None	
Return value	digLED_ReturnType	Function returns DIGLED_OK in case of successful function call and DIGLED_ERROR in case of error occurs during the function call.

Table 19 - digLED_Set_Cur_Blue API specification

3.2.7. digLED_Set_Bias

The digLED_Set_Bias command sets the whole device bias reference value. The bias controls e.g. the oscillator's frequency and the red LED's peak current. The command is intended to be used during the device' initial calibration only. The bias value is stored in a volatile register which may be burned into the OTP memory. The register may in turn be initialized from the OTP at device start-up.

There is no response to this command.

Functional Call	**digLED_Set_Bias**	
Syntax	digLED_ReturnType digLED_Set_Bias (uint8_t Param, uint16_t Address, uint8_t StripNr)	
Parameters (in)	Param	0-15: Bias reference value
	Address	0-4095: Address of the target LED. 0 addresses all LEDs of the chain. 4080-4095 addresses multicast group.
	StripNr	Strip number.
Parameters (out)	None	
Return value	digLED_ReturnType	Function returns DIGLED_OK in case of successful function call and DIGLED_ERROR in case of error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

Table 20 - digLED_Set_Bias API specification

3.2.8. digLED_Set_Config

The digLED_Set_Config command sets the configuration register. It is intended to be used before the configuration register's content is burned into the OTP.

There is no response to this command.

Functional Call	**digLED_Set_Config**	
Syntax	digLED_ReturnType digLED_Set_Config (uint16_t Param, uint16_t Address, uint8_t StripNr)	
Parameters (in)	Param	0-4095: Configuration register value
	Address	0-4095: Address of the target LED. 0 addresses all LEDs of the chain. 4080-4095 addresses multicast group.
	StripNr	Strip number.
Parameters (out)	None	
Return value	digLED_ReturnType	Function returns DIGLED_OK in case of successful function call and DIGLED_ERROR in case of error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

Table 21 - digLED_Set_Config API specification

3.2.9. digLED_Set_Trg_Adc_Cal

The digLED_Set_Trg_Adc_Cal triggers an automatic calibration of the analog digital converter. Depending on the given parameter only the ADC offset value or both the offset and the reference voltage are calibrated. The calibration values are stored in volatile registers which may be burned into the OTP memory. The registers may in turn be initialized from the OTP at device start-up.

There is no response to this command.

Functional Call	**digLED_Set_Trg_Adc_Cal**	
Syntax	digLED_ReturnType digLED_Set_Trg_Adc_Cal(uint8_t Mode, uint16_t Address, uint8_t StripNr)	
Parameters (in)	Mode	0: Trigger ADC offset self-calibration only 1: Trigger ADC offset + gain self-calibration
	Address	0-4095: Address of the target LED. 0 addresses all LEDs of the chain. 4080-4095 addresses multicast group.
	StripNr	Strip number.
Parameters (out)	None	
Return value	digLED_ReturnType	Function returns DIGLED_OK in case of successful function call and DIGLED_ERROR in case of error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

Table 22 - digLED_Set_Trg_Adc_Cal API specification

3.2.10. digLED_Set_Adc_Dac

The digLED_Set_Adc_Dac command sets the ADC's DAC value. I.e. the value given in this command is applied to the R-2R ladder. This command is intended to be used before an automatic ADC offset calibration.

There is no response to this command.

Functional Call	**digLED_Set_Adc_Dac**	
Syntax	digLED_ReturnType digLED_Set_Adc_Dac(uint16_t Param, uint16_t Address, uint8_t StripNr)	
Parameters (in)	Param	0-511: DAC value
	Address	0-4095: Address of the target LED. 0 addresses all LEDs of the chain. 4080-4095 addresses multicast group.
	StripNr	Strip number.
Parameters (out)	None	
Return value	digLED_ReturnType	Function returns DIGLED_OK in case of successful function call and DIGLED_ERROR in case of error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

Table 23 - digLED_Set_Adc_Dac API specification

3.2.11. digLED_Set_Temp_Offset

The digLED_Set_Temp_Offset command sets the temperature offset $Temp_{Offset}$ used in the calculation for the temperature compensation of the red LED channel. The temperature offset is stored in a volatile register which may be burned into the OTP memory. The register may in turn be initialized from the OTP at device start-up.

When this command is issued, a new temperature compensation is executed. I.e. the red LED's PWM duty cycle is updated according to the temperature calculation using the new temperature offset. However, this is true only, when the temperature compensation is enabled.

There is no response to this command.

Please refer to the application note ANINLC03 for a description of the red LED's temperature compensation.

Functional Call	**digLED_Set_Temp_Offset**	
Syntax	digLED_ReturnType digLED_Set_Temp_Offset(uint16_t Param, uint16_t Address, uint8_t StripNr)	
Parameters (in)	Param	0-511: $Temp_{Offset}$ value
	Address	0-4095: Address of the target LED. 0 addresses all LEDs of the chain. 4080-4095 addresses multicast group.
	StripNr	Strip number.
Parameters (out)	None	
Return value	digLED_ReturnType	Function returns DIGLED_OK in case of successful function call and DIGLED_ERROR in case of error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

Table 24 - digLED_Set_Temp_Offset API specification

3.2.12. digLED_Set_TC_Base

The digLED_Set_TC_Base command sets the base value TC_{Base} for the initialization of the temperature compensation (TC) look-up table. It is intended to be used during the device calibration process. TC_{Base} determines the value of the first address of the TC look-up table. It is used only when the device initializes the TC look-up table during device startup. TC_{Base} is stored in a volatile register which can be burned into the OTP memory.

There is no response to this command

Please refer to the application note AN_INLC_03 for a description of the red LED's temperature compensation.

Functional Call	**digLED_Set_TC_Base**	
Syntax	digLED_ReturnType digLED_Set_TC_Base (uint16_t Param, uint16_t Address, uint8_t StripNr)	
Parameters (in)	Param	0-511: TC_{Base}
	Address	0-4095: Address of the target LED. 0 addresses all LEDs of the chain. 4080-4095 addresses multicast group.
	StripNr	Strip number.
Parameters (out)	None	
Return value	digLED_ReturnType	Function returns DIGLED_OK in case of successful function call and DIGLED_ERROR in case of error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

Table 25 - digLED_Set_TC_Base API specification

3.2.13. digLED_Set_TC_Offset

The digLED_Set_TC_Offset command sets the temperature offset value TC_{Offset} for the initialization of the temperature compensation look-up table. TC_{Offset} determines the offset of two adjacent values in the table. Please note that TC_{Offset} is considered a signed value. I.e., if the MSB is set, the TC look-up table is initialized with descending values, which is the regular case. The TC offs value is stored in a volatile register which can to be burned into the OTP memory.

There is no response to this command.

Please refer to the application note AN_INLC_03 for a description of the red LED's temperature compensation.

Functional Call	**digLED_Set_TC_Offset**	
Syntax	digLED_ReturnType digLED_Set_TC_Offset (uint16_t Param, uint16_t Address, uint8_t StripNr)	
Parameters (in)	Param	0-511: TC_{Offset}
	Address	0-4095: Address of the target LED. 0 addresses all LEDs of the chain. 4080-4095 addresses multicast group.
	StripNr	Strip number.
Parameters (out)	None	
Return value	digLED_ReturnType	Function returns DIGLED_OK in case of successful function call and DIGLED_ERROR in case of error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

Table 26 - digLED_Set_TC_Offset API specification

3.2.14. digLED_Set_TC_lookup

The digLED_Set_TC_lookup command is used to write an individual entry of the temperature compensation look-up table for the red LED channel. With the command an entry is addressed and the given value is written to the table. This command allows a finer tuning of the temperature compensation than the equally distanced values as initialized from TC_{Base} and TC_{Offset}.

There is no response to this command.

Please refer to the application note ANINLC03 for a description of the red LED's temperature compensation.

Functional Call	**digLED_Set_TC_Lookup**	
Syntax	digLED_ReturnType digLED_Set_TC_Lookup (uint8_t LUT_Adr, uint16_t LUT_Value, uint16_t Address, uint8_t StripNr)	
Parameters (in)	LUT_Adr	0-10: Lookup table address
	LUT_Value	0-511: Lookup table value
	Address	0-4095: Address of the target LED. 0 addresses all LEDs of the chain. 4080-4095 addresses multicast group.
	StripNr	Strip number.
Parameters (out)	None	
Return value	digLED_ReturnType	Function returns DIGLED_OK in case of successful function call and DIGLED_ERROR in case of error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

Table 27 - digLED_Set_TC_Lookup API specification

3.2.15. digLED_Test

The digLED_Test command triggers an analog digital conversion. The multiplexer for the measured analog value is determined by the command parameter TestNr. The possible analog sources are given in Table 28. After the TEST command has finished, the result may be retrieved using a Read_Diagnostic command. Please note the temperature must always be retrieved using a Read_Temp command. It is not possible to read the temperature via a digLED_Read_Diagnostic command! There is no response to this command.

TestNr	Analog Voltage Source
-	Temperature sensor
1	5V_PRG pin
2	Digital 1.5V supply
3	Red LED cathode
4	Green LED cathode
5	Blue LED cathode
6	Bandgap

Table 28 - Multiplexer analog voltage sources

Functional Call	**digLED_Test**	
Syntax	digLED_ReturnType digLED_Test (uint8_t TestNr, uint16_t Address, uint8_t StripNr)	
Parameters (in)	TestNr	1: Measure 5V_PRG pin voltage 2: Measure digitial 1.5V supply 3: Measure red LED cathode voltage 4: Measure green LED cathode voltage 5: Measure blue LED cathode voltage 6: Measure bandgap voltage
	Address	0-4095: Address of the target LED. 0 addresses all LEDs of the chain. 4080-4095 addresses multicast group.
	StripNr	Strip number.
Parameters (out)	None	
Return value	digLED_ReturnType	Function returns DIGLED_OK in case of successful function call and DIGLED_ERROR in case of error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

Table 29 - digLED_Test API specification

3.3. Read Access

A read access consists of two phases, the command and the response phase. The command phase uses downstream communication and the response phase uses upstream communication. Commands for read access do not use the command address, i.e. these commands may not be directed to a device based on the device address.

There are two commands for read access, digLED_Read_ and digLED_Ping. The digLED_Read commands retrieve a status information from all devices and the digLED_Ping command is used to check the device chain's integrity. Only the final node in the chain responds to a digLED_Ping command.

A digLED_Read command is first received by all devices via the frame in downstream direction. The last node in the chain then immediately transmits its response frame upstream. The response frame's data field depends on the actual digLED_Read command. The response frame's address field is set according to the own device's address. All the nodes upstream forward all received response frames until a frame with the address of their adjacent node is received. Then the respective node transmits its own response frame. This procedure lasts until the chain's first node has transmitted its response frame.

A digLED_Ping command is similar to a digLED_Read command, but only the last device in the chain responds. Thus, the digLED_Ping command is executed much faster than a regular digLED_Read command.

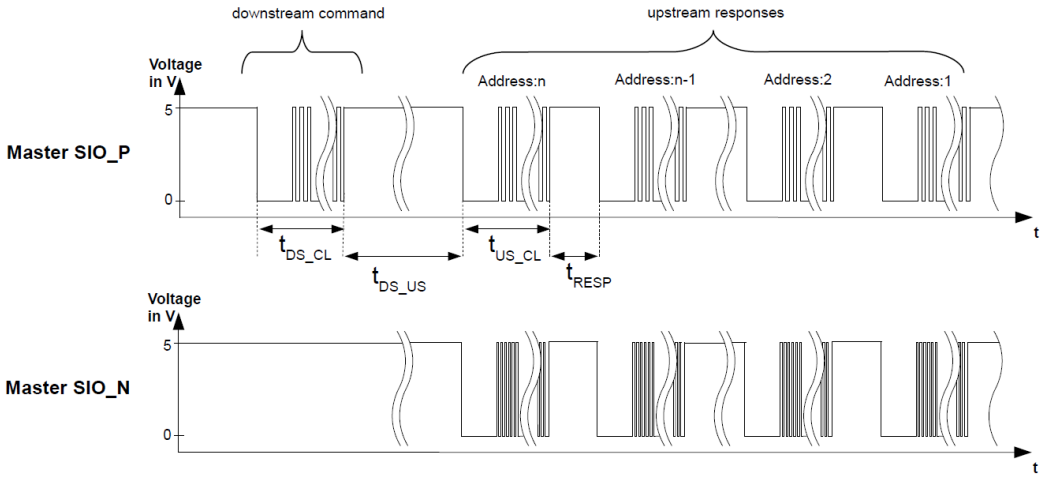


Figure 3 - Single ended read command and responses

Name	Description	Equation
t_{DS_US}	Delay between down- and upstream	$t_{DS\_CL} + t_{US\_CL} + 2 \times n \times t_{PD}$
t_{RESP}	Delay between responses Oscillator variation of adjacent devices < ±30% Oscillator variation of adjacent devices > ±30%	 $0.43 \times t_{US\_CL}$ $0.7 \times t_{US\_CL}$

Table 30 - Down- and upstream delay, delay between responses

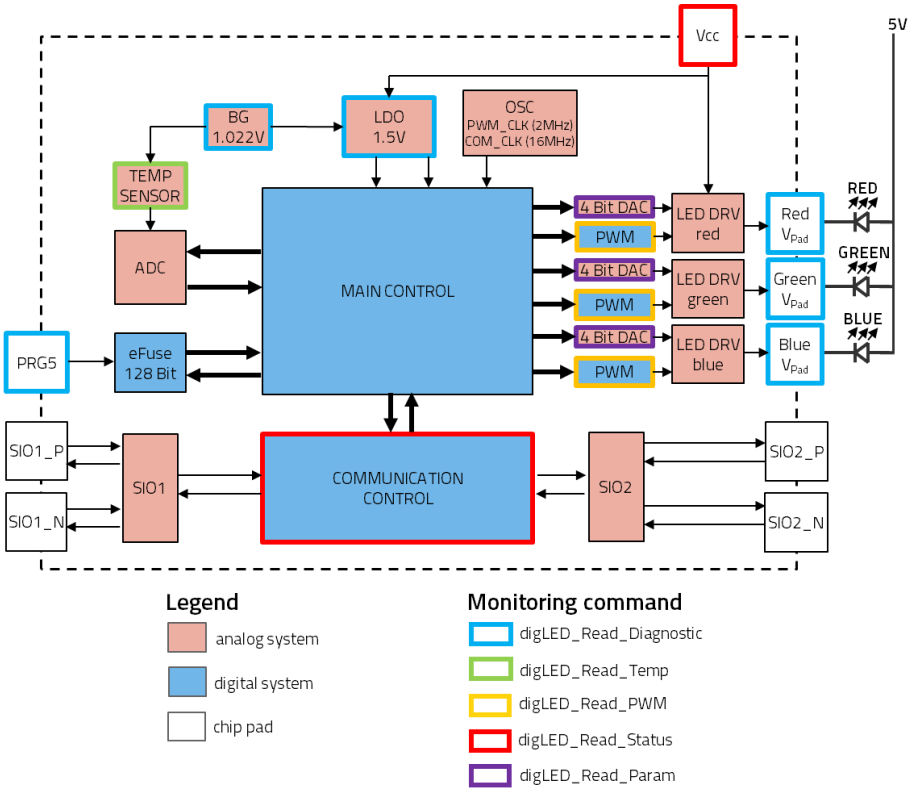


Figure 4 – Diagnostic functions

Figure 4 shows which command can read the corresponding function block.

3.3.1. digLED_Read_Diagnostic

The digLED_Read_Diagnostic command retrieves the result of the analog digital conversion last triggered by the Test command. The value may be retrieved as many times as desired.

IMPORTANT: In operation, the PRG5 pin must be grounded in order to guarantee that the effuse memory content is transferred to the registers correctly.

Functional Call	**digLED_Read_Diagnostic**	
Syntax	digLED_ReturnType digLED_Read_Diagnostic (digLED_ReadDataResultType* ChainDiagPtr, uint8_t StripNr)	
Parameters (in)	StripNr	Strip number.
Parameters (out)	ChainDiagPtr	Pointer to structure containing the diagnostic values of all LEDs in the chain.
Return value	digLED_ReturnType	Function returns DIGLED_OK in case of successful function call and DIGLED_ERROR in case of error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

Table 31 - digLED_Read_Diagnostic

Voltage Source	Voltage Range	ADC Value	Min.	Typ.	Max.
PRG5	0V … 1.2V	$V_{PRG5} \times \varepsilon_{PRG5}$	0x0	-	0x1ff
LDO	-	V_{LDO}	0x168	0x17c	0x190
RED	0V … 5.5V	$V_{RED} \times \varepsilon_{RED}$	0x0	-	0x1ff
GREEN	0V … 5.5V	$V_{GREEN} \times \varepsilon_{GREEN}$	0x0	-	0x1ff
BLUE	0V … 5.5V	$V_{BLUE} \times \varepsilon_{BLUE}$	0x0	-	0x1ff
BG	-	V_{BG}	0x11c	0x12d	0x13e

Table 32 – Analog digital converter

Parameter	Min.	Typ.	Max.	Unit
ε_{PRG5}	0.48	0.51	0.53	digits/mV
ε_{RED}	89	93	97	digits/V
ε_{GREEN}	89	93	97	digits/V
ε_{BLUE}	89	93	97	digits/V

Table 33 – Voltage detection sensitivities

3.3.1.1. Monitoring of 5V_PRG Pin Voltage (TestNr 1)

The 5V_PRG pin must be grounded to ensure that the contents of the eFuse memory are transferred to the registers correctly. It is therefore recommended to check the 5V_PRG pin immediately after power-up.

3.3.1.2. Monitoring of LDO voltage (TestNr 2)

The chip has an LDO which supplies the 1.5V logic. It is recommended to check the LDO voltage immediately after power-up.

3.3.1.3. Monitoring of RGB Pin Voltage (TestNr 3-5)

3.3.1.3.1. LEDs

The chip allows to monitor the voltage applied to the RGB pins.

Open circuits can be detected, which have arisen, for example, by dissolved (s. Figure 5 a) or broken bond wires (Figure 5 b) or faulty solder joints (Figure 5 c). In these cases, and in the case of a short circuit to GND, the return value is zero.

If the Vcc voltage is known precisely, it is also possible to detect the LED forward voltage. This is described in the example calculation of Table 34.

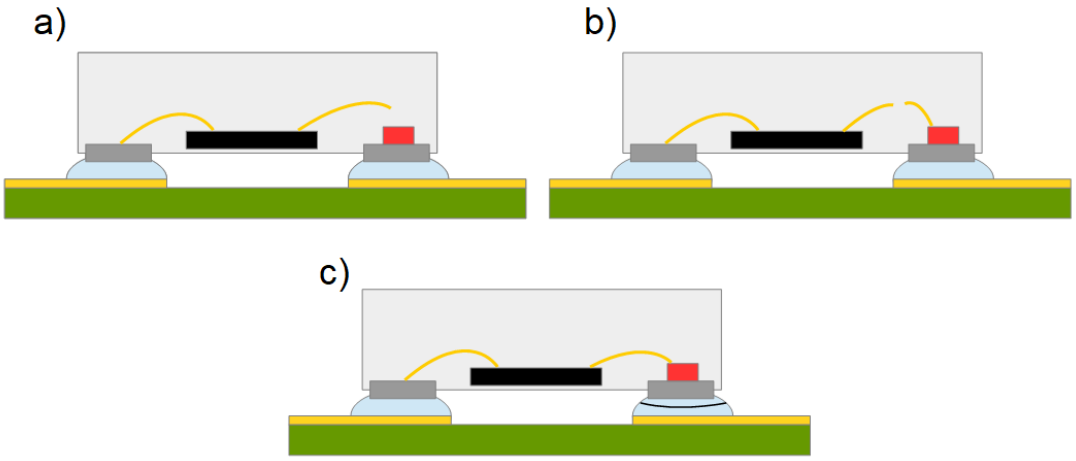


Figure 5 - Detectable open circuits

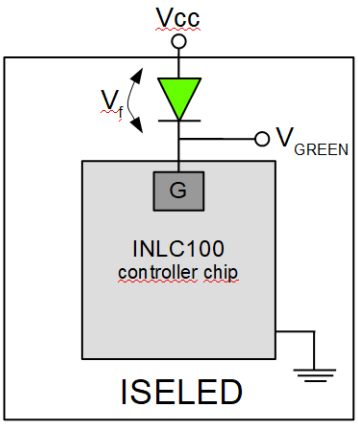
Green pad voltage calculation example	
Vcc = 5.0V V_f =2.0V I) $V_{GREEN} = Vcc - V_f$ = 5.0V - 2.0V = **3.0V** II) $ADC_{GREEN} = V_{GREEN} \times \varepsilon_{GREEN}$ = 3V x 93digits/V = **279**	

Table 34 – Forward voltage calculation example

3.3.1.3.2. Sensors

The discrete component INLC10AQ offers the possibility to connect sensors to the RGB pins, which generate an analog voltage in the range of 0V-5.5V.

For example, it is possible to use pushbuttons or photodiodes (s.Figure 6).

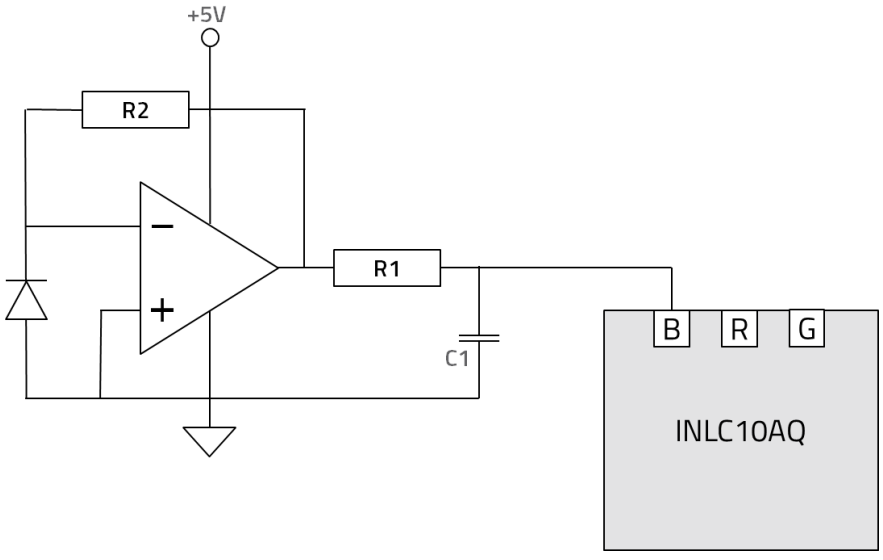


Figure 6 - Photodiode and transimpedance amplifier circuit

3.3.1.4. Monitoring of Bandgap Voltage (TestNr 6)

The chip has a bandgap reference voltage. It is recommended to check the bandgap voltage immediately after power-up.

3.3.2. digLED_Read_Temp

The digLED_Read_Temp command reads the temperature value ADC_{Temp} from all devices in the LED chain. Each device transmits a response with its own address and the last measured temperature value. Please note the temperature measurement may be either triggered periodically or by an appropriate Test command. The trigger source makes no difference to the response of the Read_Temp command.

Functional Call	**digLED_Read_Temp**	
Syntax	digLED_ReturnType digLED_Read_Temp (digLED_ReadDataResultType* ChainTempPtr, uint8_t StripNr)	
Parameters (in)	StripNr	Strip number.
Parameters (out)	ChainTempPtr	Pointer to structure containing the temperature values of all LEDs in the chain.
Return value	digLED_ReturnType	Function returns DIGLED_OK in case of successful function call and DIGLED_ERROR in case of error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

Table 35 - digLED_Read_Temp API specification

Parameter	**Description**	**Equation**	**Unit**
$T_{j,\,det}$	Detected junction temperature	$T_{j,det} = (Temp_{Offset} + 94.57 - ADC_{Temp}) / 0.87$	°C

Table 36 – Temperature detection

Ambient Temperature T_A	**Accuracy $T_{j,\,det}$**	**Unit**
-40	±17.7	°C
25	±8.8	°C
105	±4.8	°C

Table 37 - Temperature detection accuracy

3.3.3. digLED_Read_Param

The digLED_Read_Param command allows to read various device parameters. All of these parameters – except last fuse – have a corresponding digLED_Set command. This command is intended to check the integrity of the parameters when read from the OTP memory at device startup. The response format depends on the parameter selected for retrieval. Unused bits in the data field are transmitted as 0.

Table 39 describes the response frame format and data content.

Functional Call	**digLED_Read_Param**	
Syntax	digLED_ReturnType digLED_Read_Param (uint8_t ParamNumber, digLED_ReadDataResultType* ChainParamPtr, uint8_t StripNr)	
Parameters (in)	ParamNumber	Parameter number.
	StripNr	Strip number.
Parameters (out)	ChainParamPtr	Pointer to structure containing the parameter values of all LEDs in the chain.
Return value	digLED_ReturnType	Function returns DIGLED_OK in case of successful function call and DIGLED_ERROR in case of error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

Table 38 - digLED_Read_Param API specification

ParamNumber	Upstream Data Structure
1	Data[11:0] = Configuration register
2	Data[11:0] = PWM_{max} red LED
3	Data[11:0] = PWM_{max} green LED
4	Data[11:0] = PWM_{max} blue LED
5	Data[11:4] = Reserved Data[3:0] = Peak current value for green LED
6	Data[11:4] = Reserved Data[3:0] = Peak current value for blue LED
7	Data[11:9] = Reserved Data[8:0] = $Temp_{Offset}$
8	Data[11:8] = Reserved Data[7:4] = ADC offset calibration value Data[3:0] = ADC gain calibration value
9	Data[11:4] = Reserved Data[3:0] = Bias reference
10	Data[11:9] = Reserved Data[8:0] = TC_{Base}
11	Data[11:9] = Reserved Data[8:0] = TC_{Offset}
12	Data[11:1] = Reserved Data[0] = Last fuse

Table 39 - digLED_Read_Param response structure

3.3.4. digLED_Read_Status

The digLED_Read_Status command serves for retrieving the error status from the devices in the LED chain. The status is cleared to all zeros when the command is executed.

Table 41 describes the response frame format and data content.

Functional Call	**digLED_Read_Status**	
Syntax	digLED_Read_Status (digLED_ReadDataResultTyp* ChainStatusPtr, uint8_t StripNr)	
Parameters (in)	StripNr	Strip number.
Parameters (out)	ChainStatusPtr	Pointer to structure containing the status values of all LEDs in the chain.
Return value	digLED_ReturnType	Function returns DIGLED_OK in case of successful function call and DIGLED_ERROR in case of error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

Table 40 - digLED_Read_Status API specification

Data	Status	Description
Data[11:9]	-	Always transmitted as zero.
Data[8]	Protocol error	A frame has been received from the upstream or downstream interface when it was not expected in that direction.
Data[7]	Timeout error	A timeout has occurred, i.e. a response from downstream did not arrive in time.
Data[6]	Upstream CRC error	A CRC error occurred during reception from upstream.
Data[5]	Downstream CRC error	A CRC error occurred during reception from downstream.
Data[4]	Undervoltage detected	A voltage below 4.3V (typ.) has been detected on the 5V supply line.
Data[3]	Command overrun error	Another command has been received while the previous command execution was still in progress.
Data[2]	Undefined command	An undefined command or a command not allowed at the time of reception was encountered.
Data[1]	Frame/freq sync error	Frame/frequency synchronization error. A bad frame or frequency synchronization was encountered. Either the frame synchronization was longer than 235 clock cycles or the first four bits of the frequency synchronization were shorter than 16 clock cycles or longer than 62 clock cycles.
Data[0]	Symbol encoding error	An undefined symbol encoding has been received

Table 41 - digLED_Read_Status response structure

3.3.5. digLED_Ping

The digLED_Ping command may be used to check the device chain for integrity. The command frame is transmitted downstream like a digLED_Read_Status command, but only the last device in the chain sends a response frame (including its address). Thus, the whole chain is checked for connectivity without the overhead of digLED_Read commands. The address and data field of the digLED_Ping command frame are unused. The response frame transmitted by the final node holds the device' error status in the data field.

Functional Call	**digLED_Ping**	
Syntax	digLED_ReturnType digLED_Ping (digLED_ReadDataResultType* ChainPingPtr, uint8_t StripNr)	
Parameters (in)	StripNr	Strip number.
Parameters (out)	ChainPingPtr	Pointer to structure containing the ping values of the last LED in the chain.
Return value	digLED_ReturnType	Function returns DIGLED_OK in case of successful function call and DIGLED_ERROR in case of error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

Table 42 - digLED_Ping API specification

3.3.6. digLED_Read_PWM_Red/Green/Blue

The digLED_Read_PWM commands read the 12 bit PWM value for a selected LED channel back from all attached devices. The PWM calculation considers the 8 bit RGB value (s. Table 43).

Parameter	Description
PWM_{Max}	12 bit brightness calibration value
RGB	8 bit color intensity value
DIM	2 bit dimming value
TC	9 bit temperature compensation factor
$PWM_{Green/Blue}$	12 bit value: ((PWM_{Max} x RGB + 128) >> 8) >> DIM
PWM_{Red} (TC disabled)	12 bit value: ((PWM_{Max} x RGB + 128) >> 8) >> DIM
PWM_{Red} (TC enabled)	12 bit value: (((PWM_{Max} x RGB + 128) >> 8) x TC + 256) >> 9) >> DIM

Table 43 - PWM scaling

Functional Call	**digLED_Read_PWM_Red**	
Syntax	digLED_ReturnType digLED_Read_PWM_Red (digLED_ReadDataResultType* ChainPWMredPtr, uint8_t StripNr)	
Parameters (in)	StripNr	Strip number.
Parameters (out)	ChainPWMredPtr	Pointer to structure containing the red PWM values of all LEDs in the chain.
Return value	digLED_ReturnType	Function returns DIGLED_OK in case of successful function call and DIGLED_ERROR in case of error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

Table 44 - digLED_Read_PWM_Red API specification

Functional Call	digLED_Read_PWM_Green	
Syntax	digLED_ReturnType digLED_Read_PWM_Green (digLED_ReadDataResultType* ChainPWMgreenPtr, uint8_t StripNr)	
Parameters (in)	StripNr	Strip number.
Parameters (out)	ChainPWMgreenPtr	Pointer to structure containing the green PWM values of all LEDs in the chain.
Return value	digLED_ReturnType	Function returns DIGLED_OK in case of successful function call and DIGLED_ERROR in case of error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

Table 45 - digLED_Read_PWM_Green API specification

Functional Call	digLED_Read_PWM_Blue	
Syntax	digLED_ReturnType digLED_Read_PWM_Blue (digLED_ReadDataResultType* ChainPWMbluePtr, uint8_t StripNr)	
Parameters (in)	StripNr	Strip number.
Parameters (out)	ChainPWMbluePtr	Pointer to structure containing the blue PWM values of all LEDs in the chain.
Return value	digLED_ReturnType	Function returns DIGLED_OK in case of successful function call and DIGLED_ERROR in case of error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

Table 46 - digLED_Read_PWM_Blue API specification

3.4. Block Command

3.4.1. digLED_Send_Cmd_Block

The digLED_Send_Cmd_Block function is used to send multiple commands to multiple LEDs within one chain. The set of commands that can be sent with this function is limited to digLED_Set_RGB and digLED_Set_Dim. The completion of this command will be notified to the application using the callback mechanism. The transfer mechanism employed, DMA or interrupts, will depend on the configuration of the communication interface performed with digLED_Init_Interface function.

The memory allocated by the application for storing the block of commands cannot be read-only. This variable block of memory will be used by the driver for constructing the communication frames to be send to all targeted LEDs.

Functional Call	**digLED_Send_Cmd_Block**	
Syntax	digLED_ReturnType digLED_Send_Cmd_Block (digLED_SendCmdBlockType *Data, uint32_t NrOfCmds, unit8_t StripNr)	
Parameters (in)	Data	Pointer to the first element of the array of commands structures of type digLED_SendCmdBlockType.
	NrOfCmds	Number of commands.
	StripNr	Strip number.
Parameters (out)	None	
Return value	digLED_ReturnType	Function returns DIGLED_OK in case of successful function call and DIGLED_ERROR in case of error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

Table 47 - digLED_Send_Cmd_Block API specification

3.5. Control Functions

3.5.1. digLED_Set_Timeout

The digLED_Set_Timeout function is used to adjust the amount of time the driver will wait for responses from LEDs in the chain during initialization of the strip or during a read access.

Functional Call	**digLED_Set_Timeout**	
Syntax	digLED_ReturnType digLED_Set_Timeout(uint32_t Duration, unit8_t StripNr,)	
Parameters (in)	Duration	Amount of time in engineering units
	StripNr	Strip number
Parameters (out)	None	
Return value	digLED_ReturnType	Function returns DIGLED_OK in case of successful function call and DIGLED_ERROR in case of error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

Table 48 - digLED_ Set_Timeout API specification

4. Revision History

Revision	Date	Changes
0.1	February 2018	Initial Release
0.2	April 2018	Corrected table enumeration at 2.3.3/2.3.4
0.3	June 2019	• New chapter 2.0 Data types • New chapter 3.1 digLED_Init_Interface • New chapter 3.4 Block command • Update to digLED_ designation
0.4	June 2019	• Update Table 39
0.5	July 2019	• Update Table 4 • Update Table 41 • Update chapter 3.3 • New chapter 3.5 • New Chapters 3.3.1.1 – 3.3.1.3
1.0	April 2020	• Qualified release • Update chapter 3.3.1, 3.3.2 • Change Service Name to Functional Call

Table 49 - Revision history

Inova Semiconductors GmbH

Grafinger Str. 26

D-81671 Munich, Germany

Phone: +49 (0)89 / 45 74 75 - 60

Fax: +49 (0)89 / 45 74 75 - 88

E-mail: info@inova-semiconductors.de

URL: http://www.inova-semiconductors.com

is a registered trademark of Inova Semiconductors GmbH