AN3766

ISELED Microchip Driver Communication Protocol - Control Commands

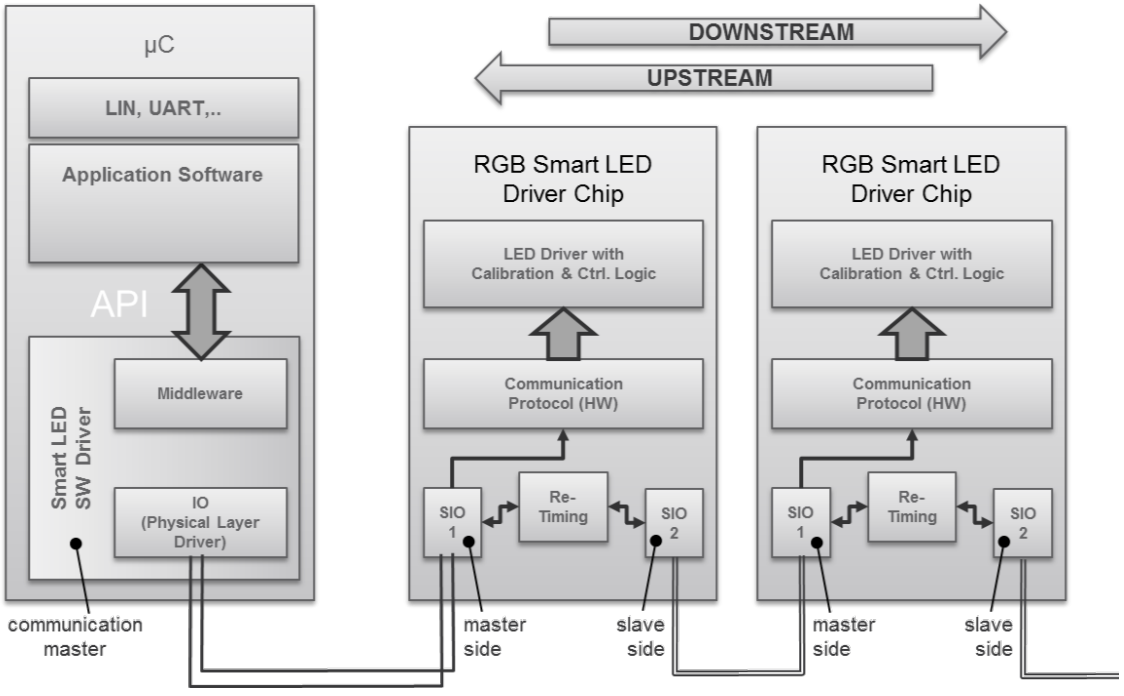
Introduction

This document describes the ISELED® communication protocol command set and its implementation in the Application Programming Interface (API) on selected Microchip microcontrollers.

Microchip provides complimentary evaluation version ISELED stacks with limited APIs upon signing a license agreement. The evaluation stacks are bound to a 90-day trial renewable license. The evaluation stacks are also limited to 30,000 counts of API function calls per Power-on Reset. The evaluation APIs are a subset of the full APIs described in this document and are outlined in 4. List of APIs Under Evaluation License.

Once a project is ready to expand into production development, the customer can request the full license driver APIs. The full license stacks do not have the API counts limitation and provide the full set of ISELED APIs. It is important to note that the full license stacks only work on MCUs with a special ISELED part number. You will need to contact your sales representative to request these MCUs. For the latest information regarding the ISELED solution from Microchip, visit www.microchip.com/iseled.

Figure 1. ISELED Serial Configuration



The ISELED communication protocol implements a half-duplex, bidirectional, high-speed serial master-slave communication between an LED strip Microchip 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 toward 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 2 Mbit/s, i.e., each bit has a nominal duration of 500 ns. As the on-die oscillator in the ISELED has 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 of the oscillator on the ISELED is 11.2-20.8 MHz.

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 an open-drain interface is used. The single-ended mode is intended to allow for an easy attachment to Microchip 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.

Table 1. 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
LDO	Low Dropout
LED	Light Emitting Diode
LUT	Lookup Table
PWM	Pulse Width Modulation
RLC	Run Length Code
TC	Temperature Compensation

Table 2. 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

.....continued		
Symbol	**Description**	**Size**
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	3x8-bit
TC_{Base}	Temperature compensation lookup table base value	9-bit
TC_{Offset}	Temperature compensation lookup table offset value	9-bit

Table of Contents

1. Data Types

1.1 digLED_InitType

Table 1-1. digLED_InitType API Specification

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.		

1.2 digLED_ConfigType

Table 1-2. digLED_ConfigType API Specification

Type Name	digLED_ConfigType		
Type	Structure		
Object	uint8_t	nrOfStrips	Number of LED strips assigned on this interface
	bool	SPI_Master_Slave or EUSART_Master_Slave	Setting the corresponding module (SPI or EUSART) to become either Master (downstream) or Slave (upstream) mode. Note: This parameter is modified for Microchip implementation
Description	A pointer to an instance of this structure, used to initialize the shared resources used for the Smart LED driver. The structure differs from the original Inova version.		

1.3 digLED_ReadDataResultType

Table 1-3. digLED_ReadDataResultType API Specification

Type Name	digLED_ReadDataResultType
Type	Structure

Object	uint16_t	chainLength	This is not implemented in the Microchip stack. #define NumberOfLEDs is used instead. Refer to 3. ISELED Driver Setup [Microchip Specific]
	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. The structure differs from the original Inova version.		

1.4 digLED_ReturnType

Table 1-4. digLED_ReturnType API Specification

Type Name	**digLED_ReturnType**	
Type	Enum	
Object	DIGLED_ERROR	Function returns with an ERROR.
	DIGLED_OK	Function returns successful.
	DIGLED_BUSY	Ongoing function.
Description	Error code of a function call.	

1.5 digLED_SendCmdBlockType

Table 1-5. digLED_SendCmdBlockType API Specification

Type Name	**digLED_SendCmdBlockType**
Type	Structure

.....continued			
Type Name	**digLED_SendCmdBlockType**		
Object	unit8_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
	unit8_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
	unit8_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
	unit16_t	addr	Address of the LED
	unit8_t	cmd	Command to be sent. 0x1 for Set_RGB, 0x2 for Set_DIM
	unit8_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.		

1.6 digLEDCtrlStruct [Microchip Specific]

Table 1-6. digLEDCtrlStruct API Specification (Microchip Specific)

Type Name	**digLEDCtrlStruct**		
Type	Structure		
Object	digLED_Instructions	Inst	Instruction code enumeration structure
	unit16_t	Addr	Address of the LED
	unit8_t	Data_Red	0-255: Set the intensity of the red LED
	unit8_t	Data_Green	0-255: Set the intensity of the green LED
	unit8_t	Data_Blue	0-255: Set the intensity of the blue LED
Description	A pointer to an instance of this structure will be used in the construction of a command for a single ISELED.		

2. Control Commands

2.1 Initialization Functions

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

2.1.1 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 SPI or EUSART. After reset or power up, the `digLED_Init_Interface` function has to be called before any other control command is called. Depending on the device family (8, 16, 32-bit), the SPI or EUSART or SPI and USART module is used for ISELED transmission. Note: Devices such as tinyAVR® MCUs will need both SPI and USART combined to generate an ISELED transmission.

Table 2-1. digLED_Init_Interface API Specification

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, it returns with DIGLED_ERROR. While executing the command, the return value is DIGLED_BUSY.

2.1.2 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. While debugging, we suggest sending a `digLED_Reset` command each time prior to the digLED_Init_Strip command. The command initializes a chain of devices by assigning the address of the device and by enabling 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 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 just initialized device's address.

All upstream devices wait for the responses to be received and forwards them to 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 checking the frame lengths) to the adjacent node in the chain.

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

Table 2-2. digLED_Init_Strip API Specification

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, it returns with DIGLED_ERROR. While executing the command, the return value is DIGLED_BUSY.

2.2 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.

2.2.1 digLED_Define_Mcast

The `digLED_Define_Mcast` command defines the multicast group membership for an individual device. There are 16 multicast address groups starting at address 4080 up to address 4095. Each device may be member in any number of these groups. The group membership is defined as a bit vector (refer to Table 2-3). Each bit represents the membership in one group. If the addressed device is to be a member of a group, the corresponding bit must be set. To define a device's 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 2-5 shows how to define the group membership of three LEDs.

Table 2-3. digLED_Define_Mcast Bit Vector

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 2-4. digLED_Define_Mcast API Specification

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 2-5. digLED_Define_Mcast Code Example

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
...			

2.2.2 digLED_Reset

The `digLED_Reset` command reinitializes 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 after power-up or a hardware reset.

The `digLED_Reset` command does not transmit any response.

Table 2-6. digLED_Reset API Specification

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 an error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

2.2.3 digLED_Set_RGB

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

There is no response to this command.

Table 2-7. digLED_Set_RGB API Specification

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 an error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

2.2.4 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 scaling (refer to Figure 2-1). 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 not applied 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 2-1. PWM over RGB

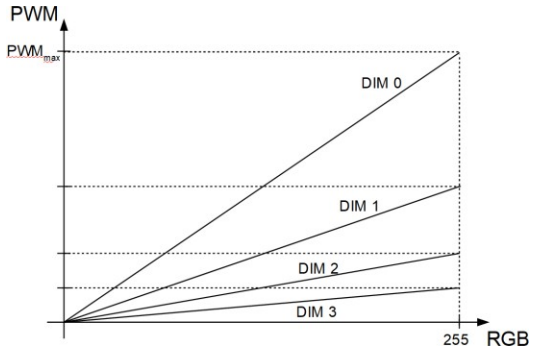


Table 2-8. digLED_Set_Dim API Specification

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	Number of bits to shift (0..3) the red LED PWM value
	Green	Number of bits to shift (0..3) the green LED PWM value
	Blue	Number 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 an error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

2.2.5 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.

Table 2-9. digLED_Set_PWM_Red API Specification

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: PWMMax 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 an error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

Table 2-10. digLED_Set_PWM_Green 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: PWMmax 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 an error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

Table 2-11. digLED_Set_PWM_Blue 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: PWMMax 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 an error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

2.2.6 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 25 ms with a tolerance of ±30%.

There is no response to these commands.

Table 2-12. digLED_Set_Cur_Green API Specification

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 an error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

Table 2-13. digLED_Set_Cur_Blue 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 an error occurs during the function call.

2.2.7 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's 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.

Table 2-14. digLED_Set_Bias API Specification

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 an error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

2.2.8 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.

Table 2-15. digLED_Set_Config API Specification

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 an error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

2.2.9 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.

Table 2-16. digLED_Set_Trg_Adc_Cal API Specification

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 an error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

2.2.10 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.

Table 2-17. digLED_Set_Adc_Dac API Specification

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 an error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

2.2.11 digLED_Set_Temp_Offset

The `digLED_Set_Temp_Offset` command sets the temperature offset TempOffset 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 only true when the temperature compensation is enabled.

There is no response to this command.

For red LED's temperature compensation, contact INOVA semiconductors.

Table 2-18. digLED_Set_Temp_Offset API Specification

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: TempOffset 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 an error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

2.2.12 digLED_Set_TC_Base

The `digLED_Set_TC_Base` command sets the base value TCBase for the initialization of the temperature compensation (TC) look-up table. It is intended to be used during the device calibration process. TCBase 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. TCBase is stored in a volatile register which can be burned into the OTP memory.

There is no response to this command

For the red LED's temperature compensation, contact INOVA semiconductors.

Table 2-19. digLED_Set_TC_Base API Specification

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: TCBase
	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 an error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

2.2.13 digLED_Set_TC_Offset

The `digLED_Set_TC_Offset` command sets the temperature offset value TCOffset for the initialization of the temperature compensation look-up table. TCOffset determines the offset of two adjacent values in the table. Note that TCOffset 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 offset value is stored in a volatile register which can to be burned into the OTP memory.

There is no response to this command.

For red LED's temperature compensation, contact INOVA semiconductors.

Table 2-20. digLED_Set_TC_Offset API Specification

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: TCOffset
	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 an error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

2.2.14 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 TCBase and TCOffset.

There is no response to this command.

For red LED's temperature compensation, contact INOVA semiconductors.

Note: Refer to specific ISELED manufactuerer datasheets for compensation values.

Table 2-21. digLED_Set_TC_Lookup API Specification

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 an error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

2.2.15 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 2-22. After the TEST command has finished, the result may be retrieved using a `Read_Diagnostic` command. 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.

Table 2-22. Multiplexer Analog Voltage Sources

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 2-23. digLED_test API Specification

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	

.....continued		
Functional Call	**digLED_Test**	
Return value	digLED_ReturnType	Function returns DIGLED_OK in case of successful function call and DIGLED_ERROR in case an error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

2.3 Read Access

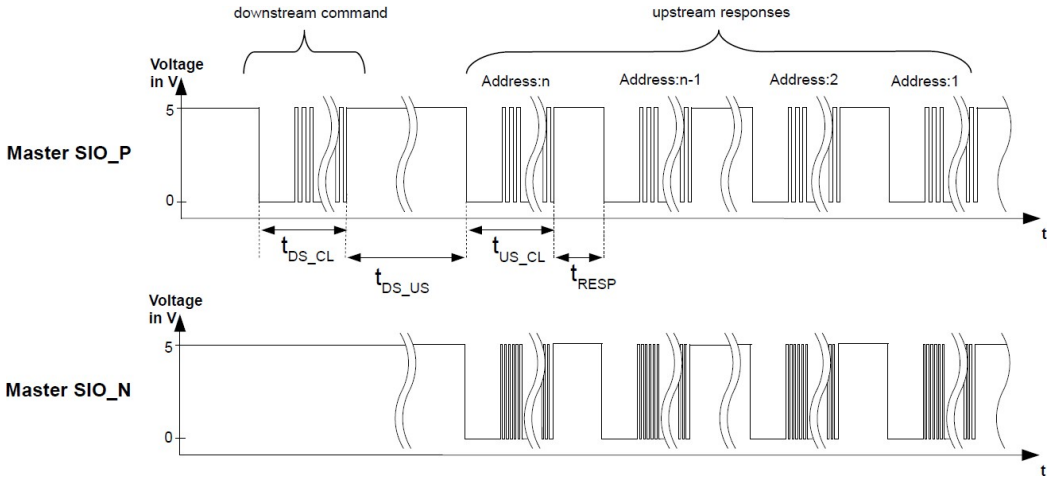
Read access consists of two phases, the command and the response phase. The command phase uses downstream communication and the response phase uses upstream communication. Commands for read access do not use the command address, i.e., these commands may not be directed to a device based on the device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

There are two commands for read access, digLED_Read_ and digLED_Ping. The digLED_Read commands retrieve 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 the 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 2-2. Single-Ended Read Command and Responses

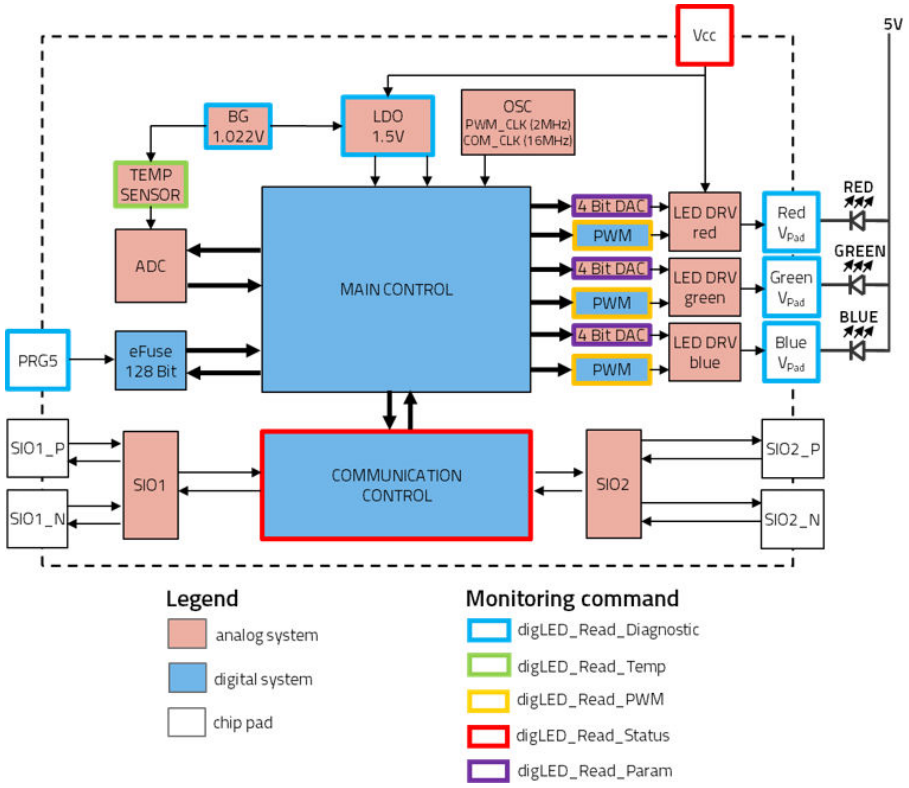


Table

Name	Description	Equation
t_{DS_US}	Delay between downstream 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%	$0.43 \times t_{US\_CL}$ $0.7 \times t_{US\_CL}$

The figure below shows which command can read the corresponding function block.

Figure 2-3. Diagnostic Functions



2.3.1 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 ensure that the effuse memory content is transferred to the registers correctly.

Table 2-25. digLED_Read_Diagnostic

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 an error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

Table 2-26. Analog Digital Converter

Voltage Source	**Voltage Range**	**ADC Value**	**Min.**	**Typ.**	**Max.**

PRG5	0V … 1.2V	VPRG5 x εPRG5	0x0	-	0x1ff
LDO	-	VLDO	0x168	0x17c	0x190
RED	0V … 5.5V	VRED x εRED	0x0	-	0x1ff
GREEN	0V … 5.5V	VGREEN x εGREEN	0x0	-	0x1ff
BLUE	0V … 5.5V	VBLUE x εBLUE	0x0	-	0x1ff
BG	-	VBG	0x11c	0x12d	0x13e

Table 2-27. Voltage Detection Sensitivities

Parameter		Min.	Typ.	Max.	Unit
εPRG5		0.48	0.51	0.53	digits/mV
εRED		89	93	97	digits/V
εGREEN		89	93	97	digits/V
εBLUE		89	93	97	digits/V

2.3.2 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.

2.3.3 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.

2.3.4 Monitoring of RGB Pin Voltage (TestNr 3-5)

LEDs

The chip allows monitoring of the voltage applied to the RGB pins.

Open circuits can be detected, which have arisen, for example, by dissolved (see Figure 2-4a) or broken bond wires Figure 2-4b or faulty solder joints (Figure 2-4c). 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 Figure 2-5

Figure 2-4. Detectable Open Circuits

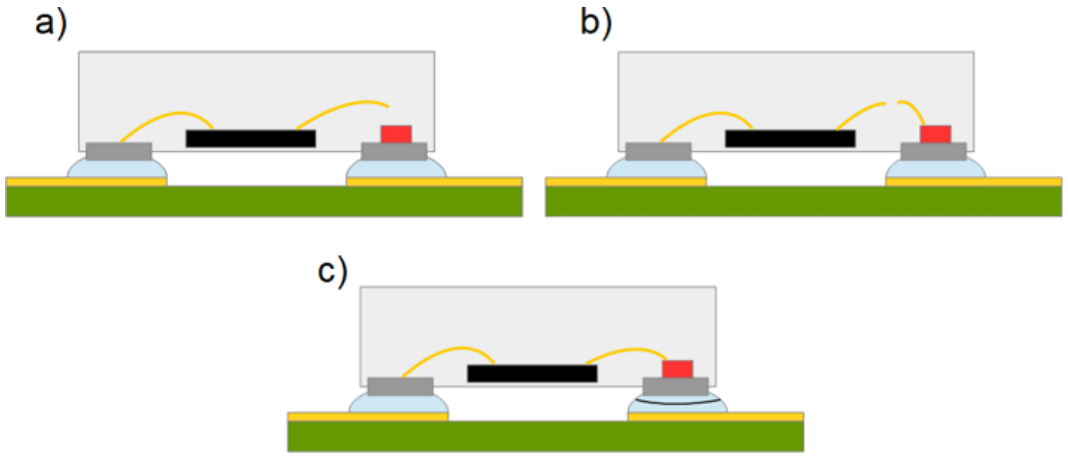


Figure 2-5. Forward Voltage Calculation Example

Green pad voltage calculation example

Vcc = 5.0V

V_f =2.0V

I) $V_{GREEN} = Vcc - V_f$

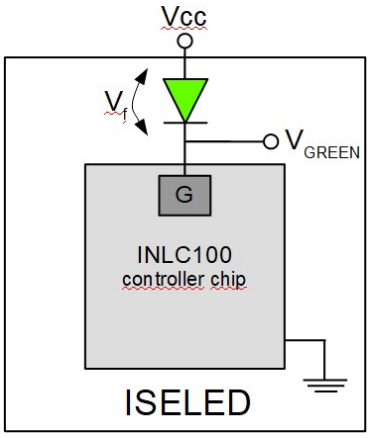
= 5.0V - 2.0V

= **3.0V**

I) $ADC_{GREEN} = V_{GREEN} \times \varepsilon_{GREEN}$

= 3V x 93digits/V

= **279**



2.3.5 Monitoring of Bandgap Voltage (TestNr 6)

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

2.3.6 digLED_Read_Temp

The `digLED_Read_Temp` command reads the temperature value ADCTemp 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.

Table 2-28. digLED_Read_Temp API Specification

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 an error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

Table 2-29. Temperature Detection

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

Table 2-30. Temperature Detection Accuracy

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

2.3.7 digLED_Read_Param

The `digLED_Read_Param` command allows reading of 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 2-32 describes the response frame format and data content.

Table 2-31. digLED_Read_Param API Specification

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 an error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

Table 2-32. digLED_Read_Param Response Structure

ParamNumber	Upstream Data Structure
1	Data[11:0] = Configuration register
2	Data[11:0] = PWMmax red LED
3	Data[11:0] = PWMmax green LED
4	Data[11:0] = PWMmax 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] = TempOffset

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] = TCBase
11	Data[11:9] = Reserved Data[8:0] = TCOffset
12	Data[11:1] = Reserved Data[0] = Last fuse

2.3.8 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 2-34 describes the response frame format and data content.

Table 2-33. digLED_Read_Status API Specification

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 an error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

Table 2-34. digLED_Read_Status Response Structure

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

2.3.9 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's error status in the data field.

Table 2-35. digLED_Ping API Specification

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 an error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

2.3.10 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 (see Table 2-36).

Table 2-36. PWM Scaling

Parameter	**Description**
PWMMax	12-bit brightness calibration value
RGB	8-bit color intensity value
DIM	2-bit dimming value
TC	9-bit temperature compensation factor

PWMGreen/Blue	12-bit value: ((PWMMax x RGB + 128) >> 8) >> DIM
PWMRed (TC disabled)	12-bit value: ((PWMMax x RGB + 128) >> 8) >> DIM
PWMRed (TC enabled)	12-bit value: ((((PWMMax x RGB + 128) >> 8) x TC + 256) >> 9) >> DIM

Table 2-37. digLED_Read_PWM_Red API Specification

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 an error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

Table 2-38. digLED_Read_PWM_Green 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 an error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

Table 2-39. digLED_Read_PWM_Blue 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 an error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

2.4 Block Command

2.4.1 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.

The Microchip ISELED stack includes an empty placeholder function for this command; it is up to the user to use and setup this function according their application requirements.

Table 2-40. digLED_Send_Cmd_Block API Specification

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 an error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

2.5 Control Functions

2.5.1 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.

The Microchip ISELED stack includes an empty placeholder function for this command; the timeout handling is part of the protocol API function `digLED_ReturnType DRV_digLED_Read_Rec` and it is adapted to the microcontroller specific hardware resources for the protocol message transfer.

Table 2-41. digLED_ Set_Timeout API Specification

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 an error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

2.6 Protocol Functions [Microchip Proprietary]

2.6.1 DRV_digLED_Read_Rec

The `DRV_digLED_Read_Rec` function is used in the Read Access commands (except the `digLED_Ping`). It initiates the transmit of the read access command and receives the responses coming from the LEDs. It checks the CRC of the LED responses and the initialization status by sending the `digLED_Init_Strip` command. It includes the timeout handling in case an LED is not responding (damaged LED or a connection failure).

Table 2-42. DRV_digLED_Read_Rec API Specification

Functional Call	DRV_digLED_Read_Rec	

Syntax	`digLED_ReturnType DRV_digLED_Read_Rec (`
	`  digLEDCtrlStruct * LEDraw,`
	`  digLED_ReadDataResultType * ResPtr,`
	`  unit8_t InitTest`
	`)`	

Parameters (in)	LEDraw	Pointer to the structure containing the LED command, address and color values
	InitTest	Indicator for the digLED_Init_Strip command (=1)
Parameters (out)	ResPtr	Pointer to structure containing the response values of the LEDs in the chain
Return value	digLED_ReturnType	Function returns DIGLED_OK in case of successful function call and DIGLED_ERROR in case an error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

2.6.2 DRV_digLED_Write_Command_wCRC

The `DRV_digLED_Write_Command_wCRC` function initiates the transmit of a downstream command with the CRC.

Table 2-43. DRV_digLED_Write_Command_wCRC API Specification

Functional Call	DRV_digLED_Write_Command_wCRC

Syntax	`digLED_ReturnType DRV_digLED_Write_Command_wCRC ( digLEDCtrlStruct * LEDtx )`	
Parameters (in)	LEDtx	Pointer to the structure containing the LED command, address and color values
Return value	digLED_ReturnType	Function returns DIGLED_OK in case of successful function call and DIGLED_ERROR in case an error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

2.6.3 DRV_digLED_Write_Command_woCRC

The `DRV_digLED_Write_Command_woCRC` function initiates the transmit of a downstream command without the CRC.

Table 2-44. DRV_digLED_Write_Command_woCRC API Specification

Functional Call	**DRV_digLED_Write_Command_woCRC**	
Syntax	`digLED_ReturnType DRV_digLED_Write_Command_woCRC ( digLEDCtrlStruct * LEDtx )`	
Parameters (in)	LEDtx	Pointer to the structure containing the LED command, address and color values
Return value	digLED_ReturnType	Function returns DIGLED_OK in case of successful function call and DIGLED_ERROR in case an error occurs during the function call. While executing the command, the return value is DIGLED_BUSY.

3. ISELED Driver Setup [Microchip Specific]

As shown in the figure below, there are four #defines for you to set up the ISELED application.

Figure 3-1. ISELED Driver Defines

```
/* define the number of LEDs on the string */
#define NumberOfLEDs  10
#define chainLength   NumberOfLEDs

/* set the string start address number */
#define START_ADDRESS 1

/* enable the ISELED message protocol CRC */
#define digLED_with_CRC
```

4. List of APIs Under Evaluation License

DATA TYPES

- digLED_InitType
- digLED_ConfigType
- digLED_ReadDataResultType
- digLED_ReturnType
- digLED_SendCmdBlockType
- digLEDCtrlStruct

INITIALIZATION FUNCTIONS

- digLED_Init_Interface
- digLED_Init_Strip

WRITE COMMANDS

- digLED_Define_Mcast
- digLED_Reset
- digLED_Set_RGB

READ ACCESS

- digLED_Read_Temp

BLOCK COMMAND

- digLED_Send_Cmd_Block

CONTROL FUNCTION

- digLED_Set_Timeout

PROTOCAL FUNCTIONS [MICROCHIP PROPRIETARY]

- DRV_digLED_Read_Rec
- DRV_digLED_Write_Command_wCRC
- DRV_digLED_Write_Command_woCRC
- digLED_Load_TC_cal

5. Revision History

Table 5-1. Revision History

Revision	Date	Changes
0	April 2020	Initial Release
1	December 2020	Modified with Microchip specifics

The Microchip Website

Microchip provides online support via our website at www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip design partner program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Product Change Notification Service

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: www.microchip.com/support

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods being used in attempts to breach the code protection features of the Microchip devices. We believe that these methods require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Attempts to breach these code protection features, most likely, cannot be accomplished without violating Microchip's intellectual property rights.
- Microchip is willing to work with any customer who is concerned about the integrity of its code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is "unbreakable." Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Legal Notice

Trademarks

ISBN: 978-1-5224-7291-9

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: www.microchip.com/support Web Address: www.microchip.com **Atlanta** Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455 **Austin, TX** Tel: 512-257-3370 **Boston** Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088 **Chicago** Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075 **Dallas** Addison, TX Tel: 972-818-7423 Fax: 972-818-2924 **Detroit** Novi, MI Tel: 248-848-4000 **Houston, TX** Tel: 281-894-5983 **Indianapolis** Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380 **Los Angeles** Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 **Raleigh, NC** Tel: 919-844-7510 **New York, NY** Tel: 631-435-6000 **San Jose, CA** Tel: 408-735-9110 Tel: 408-436-4270 **Canada - Toronto** Tel: 905-695-1980 Fax: 905-695-2078	**Australia - Sydney** Tel: 61-2-9868-6733 **China - Beijing** Tel: 86-10-8569-7000 **China - Chengdu** Tel: 86-28-8665-5511 **China - Chongqing** Tel: 86-23-8980-9588 **China - Dongguan** Tel: 86-769-8702-9880 **China - Guangzhou** Tel: 86-20-8755-8029 **China - Hangzhou** Tel: 86-571-8792-8115 **China - Hong Kong SAR** Tel: 852-2943-5100 **China - Nanjing** Tel: 86-25-8473-2460 **China - Qingdao** Tel: 86-532-8502-7355 **China - Shanghai** Tel: 86-21-3326-8000 **China - Shenyang** Tel: 86-24-2334-2829 **China - Shenzhen** Tel: 86-755-8864-2200 **China - Suzhou** Tel: 86-186-6233-1526 **China - Wuhan** Tel: 86-27-5980-5300 **China - Xian** Tel: 86-29-8833-7252 **China - Xiamen** Tel: 86-592-2388138 **China - Zhuhai** Tel: 86-756-3210040	**India - Bangalore** Tel: 91-80-3090-4444 **India - New Delhi** Tel: 91-11-4160-8631 **India - Pune** Tel: 91-20-4121-0141 **Japan - Osaka** Tel: 81-6-6152-7160 **Japan - Tokyo** Tel: 81-3-6880- 3770 **Korea - Daegu** Tel: 82-53-744-4301 **Korea - Seoul** Tel: 82-2-554-7200 **Malaysia - Kuala Lumpur** Tel: 60-3-7651-7906 **Malaysia - Penang** Tel: 60-4-227-8870 **Philippines - Manila** Tel: 63-2-634-9065 **Singapore** Tel: 65-6334-8870 **Taiwan - Hsin Chu** Tel: 886-3-577-8366 **Taiwan - Kaohsiung** Tel: 886-7-213-7830 **Taiwan - Taipei** Tel: 886-2-2508-8600 **Thailand - Bangkok** Tel: 66-2-694-1351 **Vietnam - Ho Chi Minh** Tel: 84-28-5448-2100	**Austria - Wels** Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 **Denmark - Copenhagen** Tel: 45-4485-5910 Fax: 45-4485-2829 **Finland - Espoo** Tel: 358-9-4520-820 **France - Paris** Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 **Germany - Garching** Tel: 49-8931-9700 **Germany - Haan** Tel: 49-2129-3766400 **Germany - Heilbronn** Tel: 49-7131-72400 **Germany - Karlsruhe** Tel: 49-721-625370 **Germany - Munich** Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 **Germany - Rosenheim** Tel: 49-8031-354-560 **Israel - Ra'anana** Tel: 972-9-744-7705 **Italy - Milan** Tel: 39-0331-742611 Fax: 39-0331-466781 **Italy - Padova** Tel: 39-049-7625286 **Netherlands - Drunen** Tel: 31-416-690399 Fax: 31-416-690340 **Norway - Trondheim** Tel: 47-72884388 **Poland - Warsaw** Tel: 48-22-3325737 **Romania - Bucharest** Tel: 40-21-407-87-50 **Spain - Madrid** Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 **Sweden - Gothenberg** Tel: 46-31-704-60-40 **Sweden - Stockholm** Tel: 46-8-5090-4654 **UK - Wokingham** Tel: 44-118-921-5800 Fax: 44-118-921-5820