

1. Introduction

This document provides guidelines on how to create PCB design and ISELED software driver setup reference with seddLED3.0 LED. All functions can be controlled via the S32K1xx microcontroller unit (MCU) from NXP that provides the ISELED serial communication protocol. seddLED (Smart Embedded Digital Driver LED) is the world's first digital LED which combines RGB LED, LED Driver and advanced communication link integrated into a single package. It is pre-calibrated to D65 white point with an accuracy within 3 SDCM steps at 1,400mcd.

List of Abbreviations and Acronyms

API	Application Programming Interface
CAN	Controller Area Network
EMI	Electromagnetic Interference
ESD	Electrostatic Discharge
GND	Ground
LIN	Local Interconnect Network
MCU	Microcontroller Unit
PCB	Printed Circuit Board

Contents

1. Introduction	1
2. Customer Application Development Software Setup	3
2.1. S32 Design Studio for ARM 2018.R1 and S32K1xx SDK ISELED Driver Installation Procedure	3
2.2. ISELED free master program template	4
2.3. Key features of Processor Expert	6
2.4. Key features of main.c and API command.	8
2.5. Steps to Compiling and Generating .SREC file	10
2.6. Real Time Debug – PE Micro	12
3. Customer Application Development Hardware Setup	13
3.1. A3A-FKG Pin out	13
3.2. NXP S32K1xx Series Microcontroller Features	14
3.3. External Component guide	15
3.4. PCB design guide, pin-per-pin explanation	16
3.5. Extended connection guide (module-to-module) explanation	18
3.6. A3A-FKG solder pad reference	19
4. Revision History	20

2. Customer Application Development Software Setup

2.1. S32 Design Studio for ARM 2018.R1 and S32K1xx SDK ISELED Driver Installation Procedure

The S32 Design Studio IDE is a complimentary integrated development environment for Automotive and Ultra-Reliable Arm® based MCUs that enables editing, compiling and debugging of designs. The S32 Design Studio IDE offers designers a straightforward development tool with no code-size limitations.

The S32K SDK ISELED driver BETA 2.9.2 provides the user with necessary API to initialize and drive the ISELED LEDs. It provides two options for communication interface, highly effective FlexIO interface and standard LPSPI interface. The S32K SDK ISELED driver is not part of standard S32K SDK installation and a special activation code is needed to download the installation package from NXP web.

Users are encouraged to follow NXP ***“S32K SDK ISELED driver installation guide”*** for the complete guide to download and install S32 Design Studio for ARM 2018.R1 and S32K1xx SDK ISELED Driver.

The ***“S32K SDK ISELED driver installation guide”*** can be obtained from Dominant Sales Representative. This document contained S32K ARM Version 2018.R1 (or newer) and ISELED S32K14x BETA 2.9.2 (or newer) proper installation guide. Users are advised to check the latest ISELED driver software with gpis.software@nxp.com or your NXP Sales representative.

Next chapter describe in detail steps to begin writing ISELED programming code and correct compiling procedure to generate needed flash file (.srec) using NXP S32K Version 2.0 Design Studio.

Note: Despite this document might refer to particular revision of NXP Software, it is highly recommended to use the latest available versions of the software. Those versions might not be listed in the screenshots presented in this document at the time of creation.

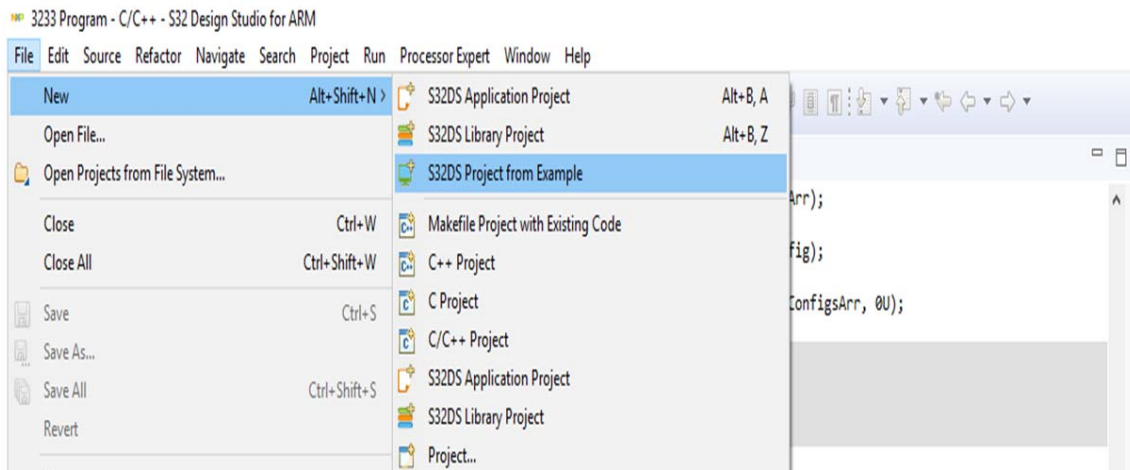
2.2. ISELED freemaster program template

Depending on the microcontroller version that you have purchased, there are dedicated iseled_freemaster file within each dedicated microcontroller variant folder.

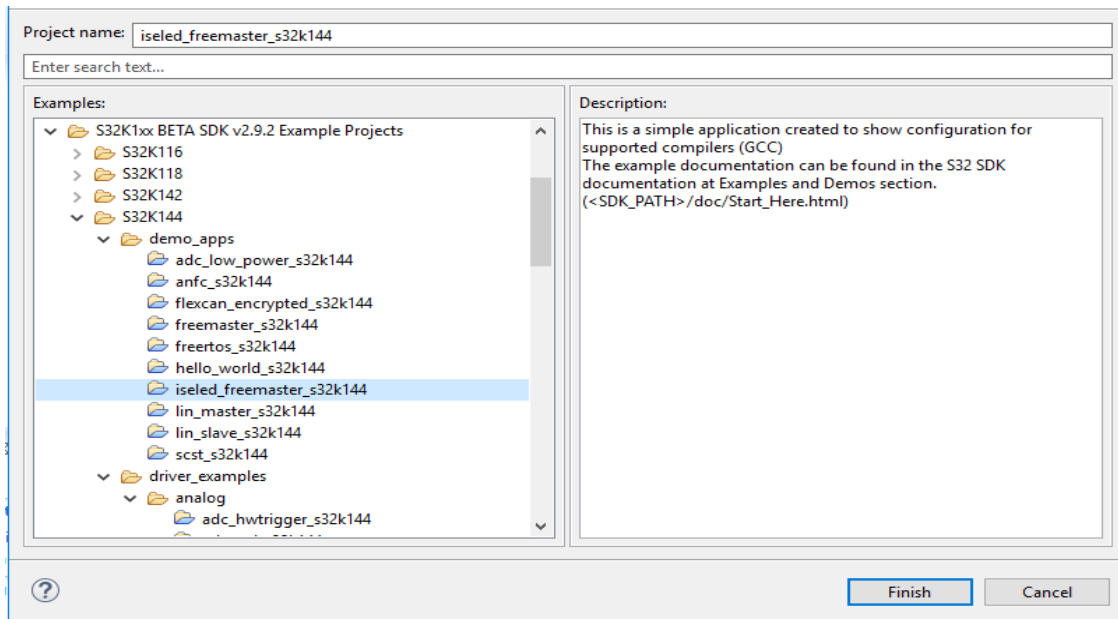
Note: For this application guide, we will use S32K144 as reference folder.

With proper installation of S32K1xx BETA v2.9.2 ISELED Driver and ARM 2018.R1, user will be able to prepare ISELED program layout using available template.

In order to use this template:



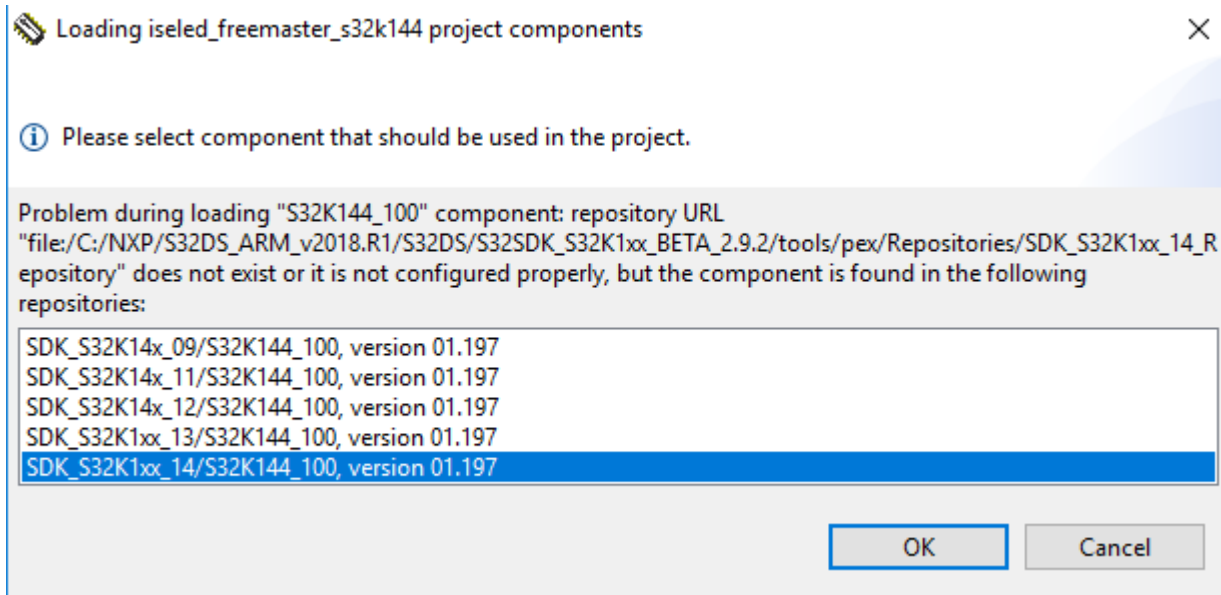
Select S32K1xx BETA SDK v2.9.2 Example Projects --> S32K144 --> demo_apps --> iseled_freemaster_s32k144.



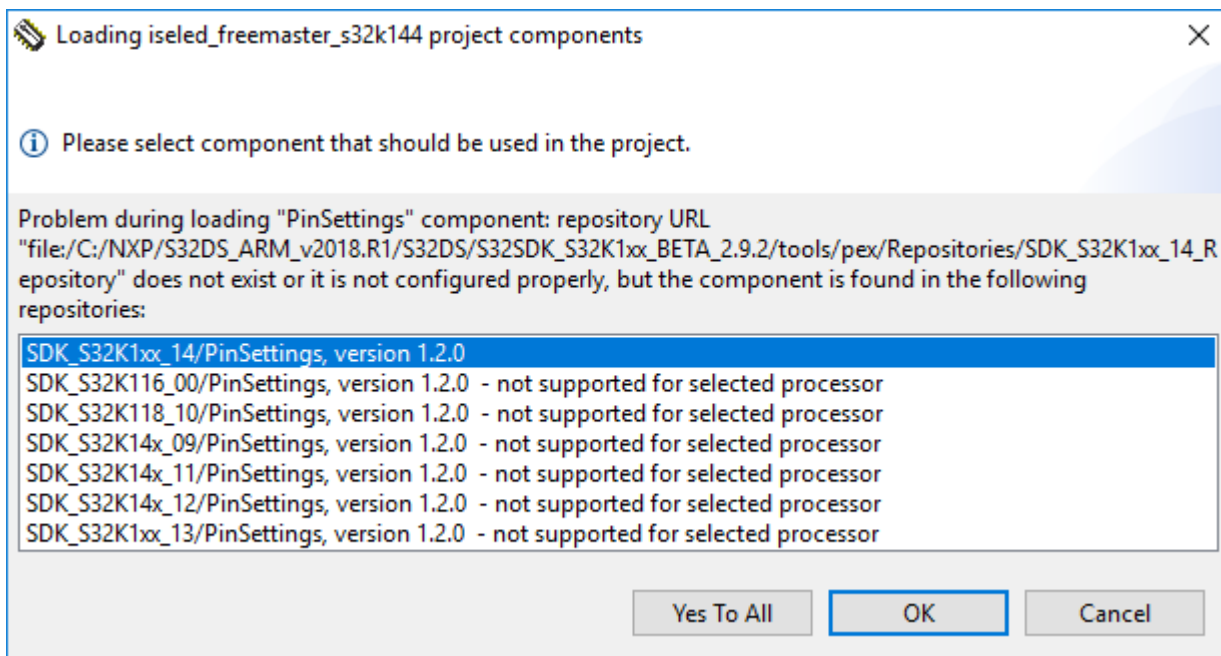
When requested to choose project components, please select the latest version.

In the case for S32K144 folder, we use the latest version as highlighted:

SDK_S32K1xx_14/S32K144_100, version01.197.

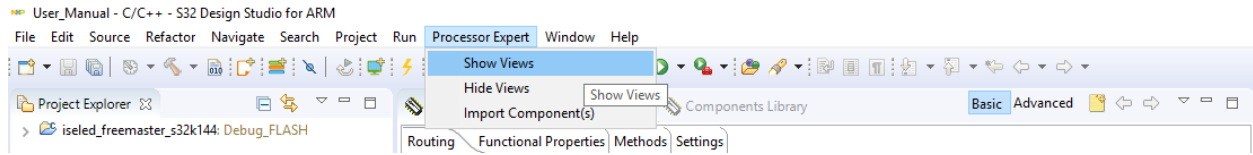


Next, please select the right processor:

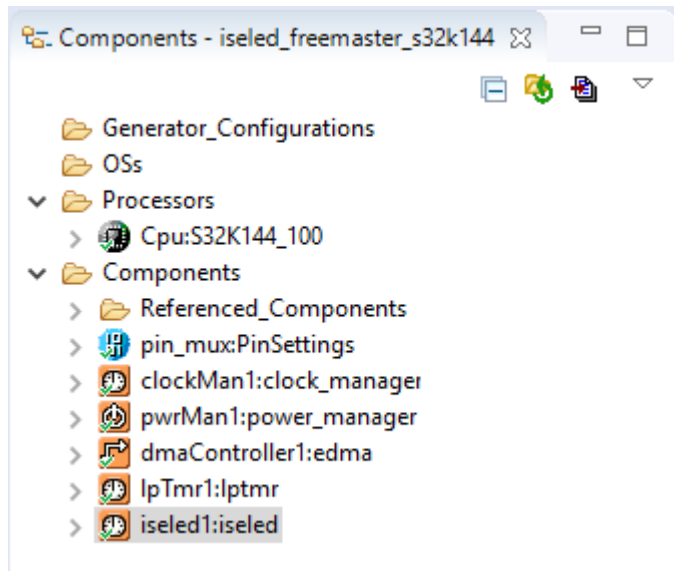


2.3. Key features of Processor Expert

Go to the “Processor Expert” tab, and click “Show Views”



At the Processor Expert Components Section, you will find a series of processor configurations including Pin Settings, Clock Manager, Power Manager, DMA Controller, Timer and ISELED1.

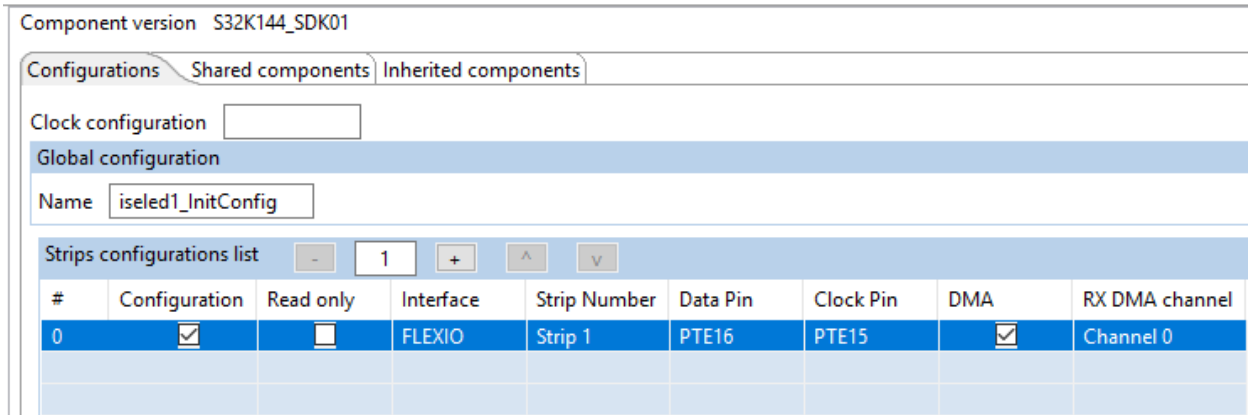


User do not need to change most of the settings at these components except ISELED1.

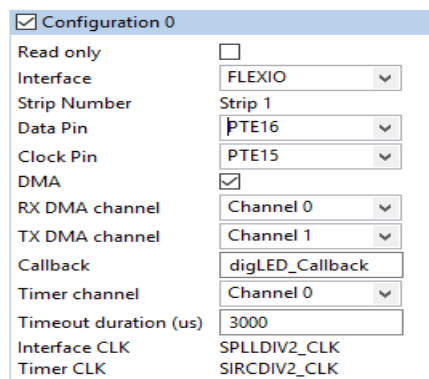
In the ISELED1, user can add more channel/strips and assign data/clock (SIO1_P/SIO1_N) pin-out.

To add more channel/strip, user will need to increase “strips configuration list”.

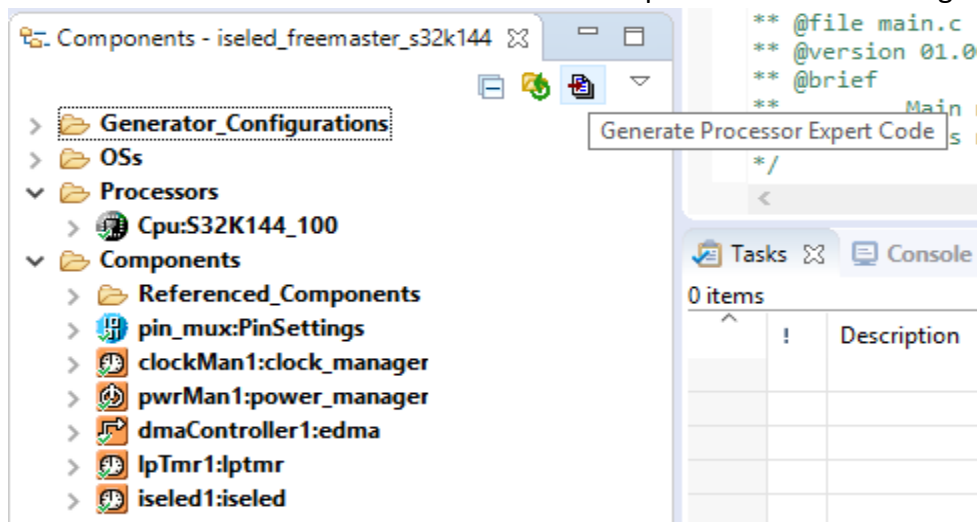
By default, Data and Clock Pin will be automatically assigned.



However, user can also manually change the data/clock pin assignment at the configuration setting.



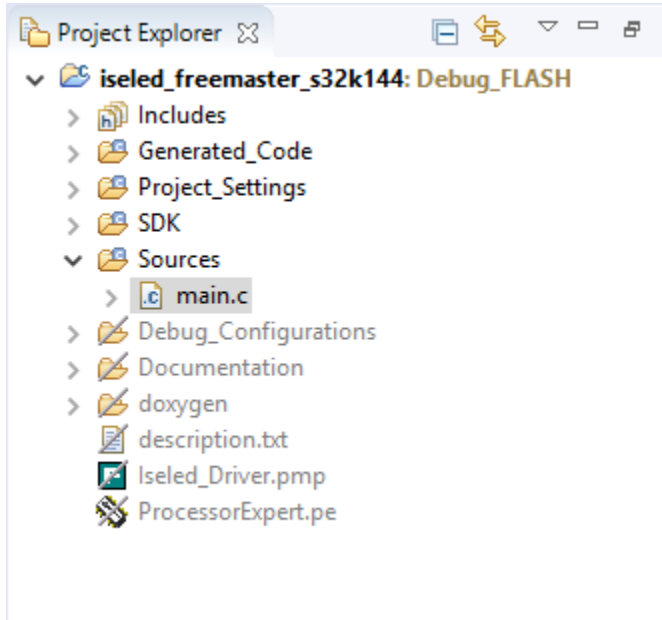
Please remember to click “Generate Processor Expert Code” once setting done.



2.4. Key features of main.c, and API command.

Again, Go to the “Processor Expert” tab, and click “Hide Views”

Then, at the “Project Explorer” click “Sources -> main.c”



Main.c :

Main.c includes all the needed modules and freemaster function. However, in this introduction, we will focus only editing int main(void) function.

In the int main (void) function, user can include to enable/disable testInitType.crcEnable, testInitType.firstLedAdr, testInitType.tempCmpEnable, testInitType.voltSwing.

Please **remember to turn on/enable** testInitType.tempCmpEnable to get accurate, factory calibrated LED brightness setting. By Default:

```
testInitType.crcEnable = 1;
```

```
testInitType.firstLedAdr = 1;
```

```
testInitType.tempCmpEnable = 1;
```

```
testInitType.voltSwing = 0;
```


Next, the following segments are the default code. Do not change!!

```
CLOCK_SYS_Init(&g_clockManConfigsArr, CLOCK_MANAGER_CONFIG_CNT, &g_clockManCallbacksArr, CLOCK_MANAGER_CALLBACK_CNT);
CLOCK_SYS_UpdateConfiguration(0U, CLOCK_MANAGER_POLICY_FORCIBLE);

PINS_DRV_Init(NUM_OF_CONFIGURED_PINS, g_pin_mux_InitConfigArr);

EDMA_DRV_Init(&dmaController1_State, &dmaController1_InitConfig0, edmaChnStateArray, edmaChnConfigArray, EDMA_CONFIGURED_CHANNELS);

digLED_Init_Interface(NUMBER_OF_INTERFACES, iseled1_InitConfig);

TIMING_Init(&timing_pall_instance, &timing_pall_InitConfig);

LPTMR_DRV_Init(INST_LPTMR1, &lpTmr1_config0, false);
/* Install IRQ handler for LPTMR interrupt */
INT_SYS_InstallHandler(LPTMR0_IRQn, &lpTmrISR, (isr_t *)0);
/* Enable IRQ for LPTMR */
INT_SYS_EnableIRQ(LPTMR0_IRQn);

/* Setup GPIO pins
 * - LED0 and LED1 as output
 */
PINS_DRV_SetPinsDirection(LED_GPIO, (1 << LED0)|(1 << LED1));

/* Turn off both LEDs on EVB */
PINS_DRV_SetPins(LED_GPIO, (1 << LED0)|(1 << LED1));

POWER_SYS_Init(&powerConfigsArr, 2U, &powerStaticCallbacksConfigsArr, 0U);

#ifdef CPU_IN_HSRUN_MODE
{
    POWER_SYS_SetMode(1U, POWER_MANAGER_POLICY_AGREEMENT);
}
#else
{
    POWER_SYS_SetMode(0U, POWER_MANAGER_POLICY_AGREEMENT);
}
#endif

delay(100000);
/*Start the timer necessary for the 50fps required by the updating functions (scanColor and dmaBlock)*/
LPTMR_DRV_StartCounter(INST_LPTMR1);
...
..
```

API Command

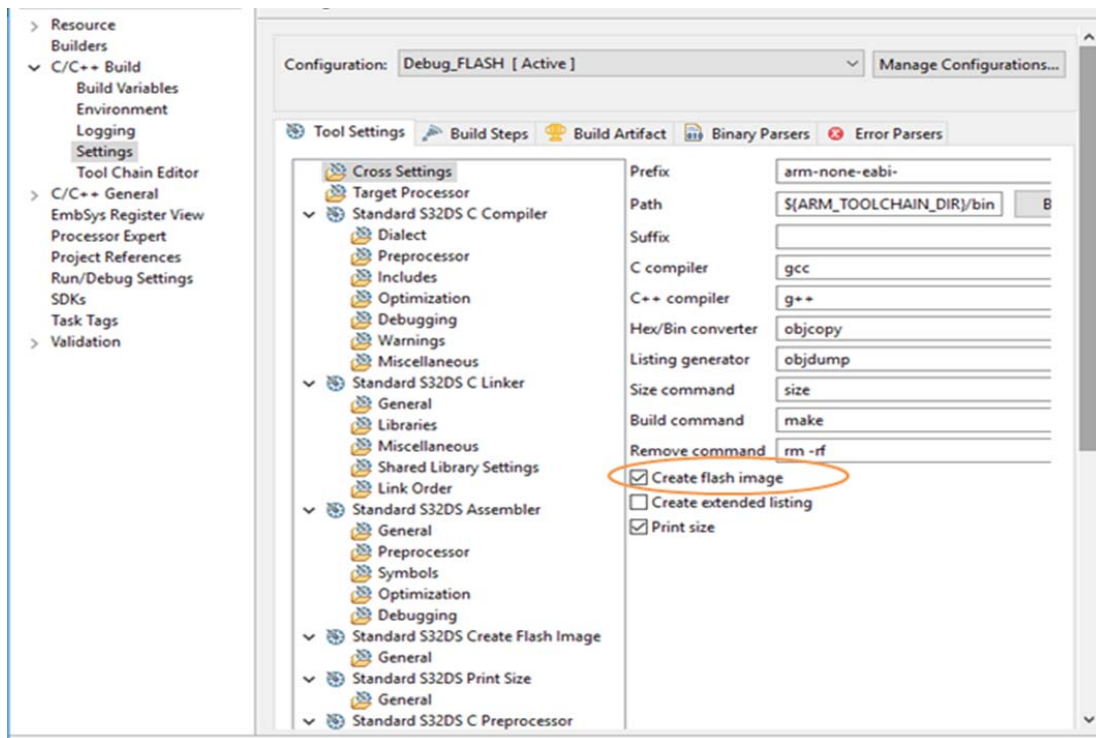
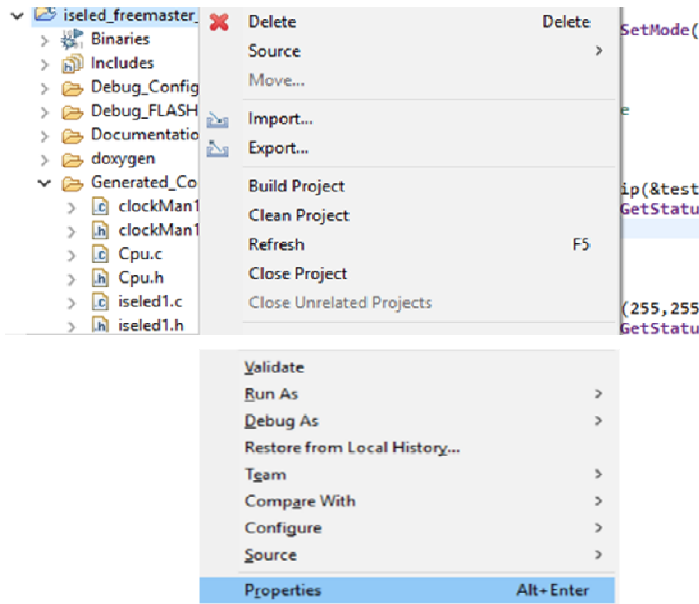
User can locate all the API command at the folder /SDK/platform/drivers/inc/iseled_driver.h

However, there are many API lists here are dedicated for factory level calibration. Editing these parameters will amend seddLED3.0 factory setting.

Thus, user advised to follow **SmartRGB User API Programming Manual** for the most accurate use of API command list.

2.5. Compiling and Generating .SREC file

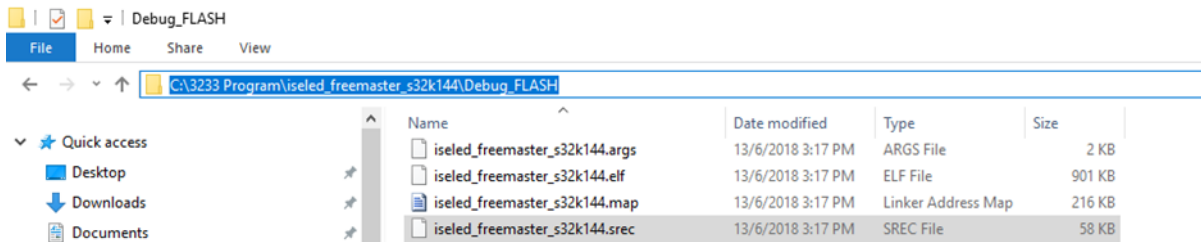
Once complete code write up, please remember to select “create flash image” under properties
 --> C/C++ Build --> Settings



Perform right click at the `iseled_freemaeter_s32k144` folder and click “Build Project”.



Look at the local folder and search for the latest `iseled_freemaster_s32k144.srec`

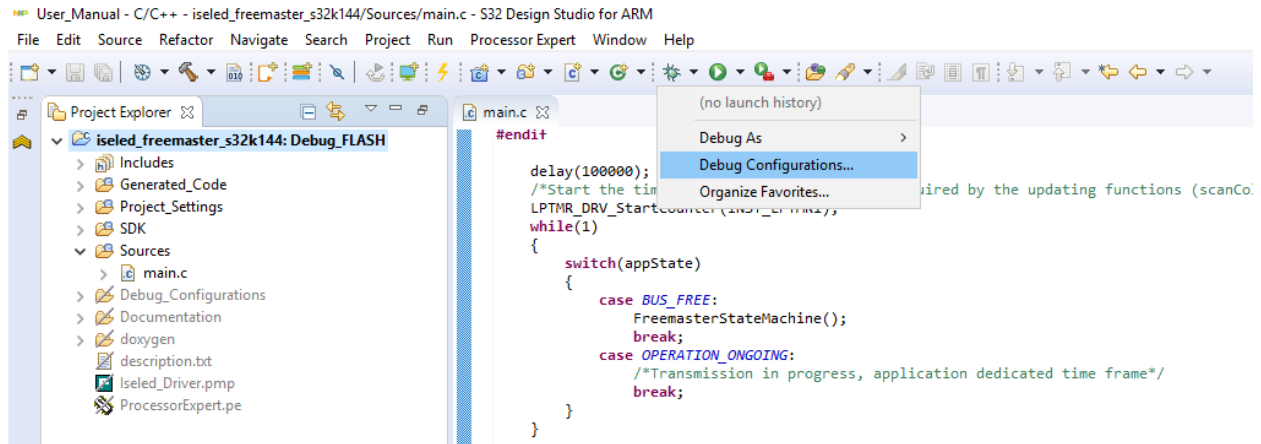


2.6. Real Time Debug – PE Micro

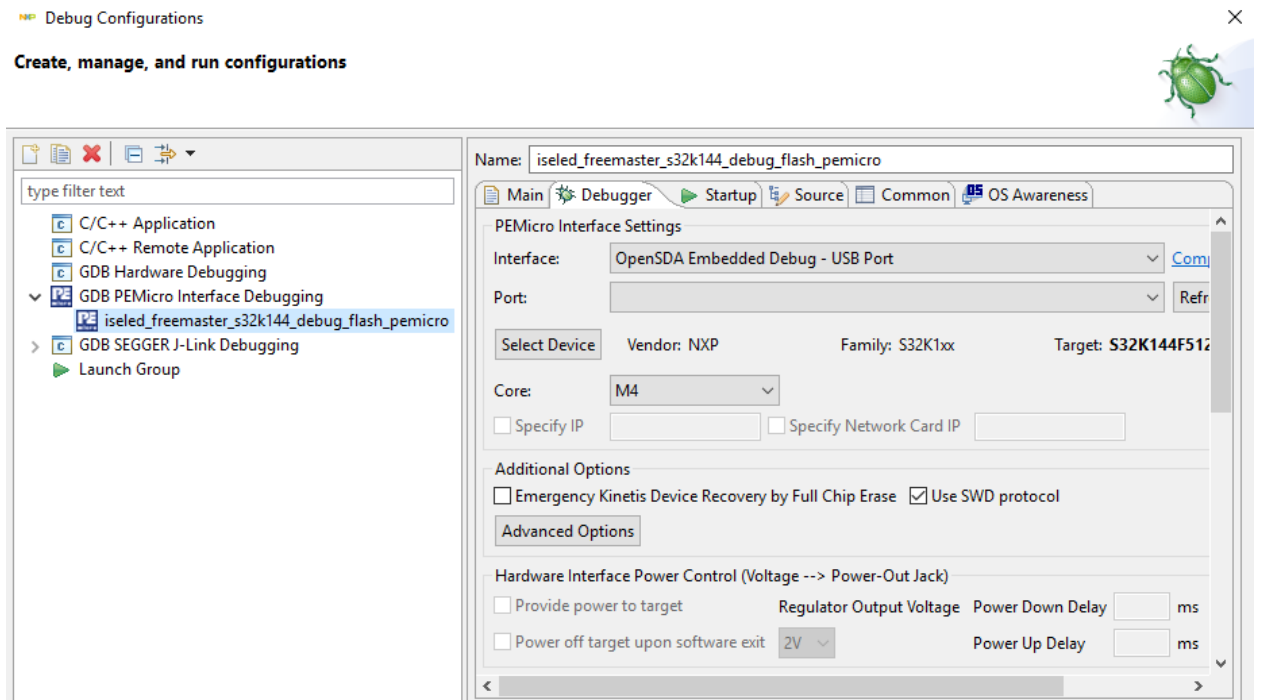
Real time debug can help user to perform real time diagnostic of the seddLED chain via API command such as **digLED_Read_Temp** and **digLED_Read_Status**.

This section only provides directory to enable Debug environment. Subsequent steps will rely on the user own experience to carry out microcontroller debugging skills.

To use PE Micro debug, click “Debug Configurations” as shown below:



Choose GDB PEMicro Interface Debugging →
iseled_freemaster_s32k144_debug_flash_pemicro.

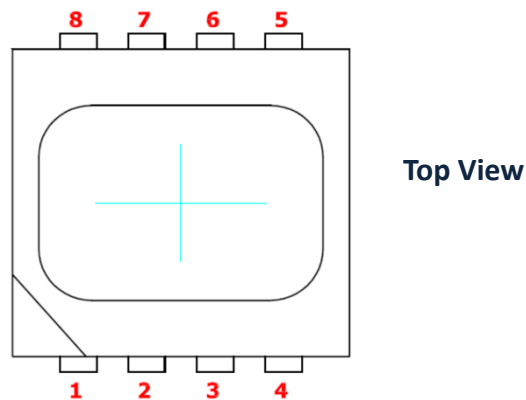


3. Customer Application Development Hardware Setup

This section provides guidelines to the development of seddLED3.0 LED typical hardware application setup. Selection of right external component and PCB layout is critical to ensure proper functionality and heat dissipation when daisy-chain many seddLED3.0 LED together onto a small PCB surface.

3.1 A3A-FKG Pin out

Please follow this pin-out for connection and pin functionality:



Pin	Pin	Direction	Description
1	PRG	IN	Has to be connected to GND for proper operation
2	SiO1_N	IO	Serial Communication Interface Master Side, Negative Polarity
3	SiO1_P	IO	Serial Communication Interface Master Side, Positive Polarity
4	GND	Supply	Ground
5	GND	Supply	Ground
6	SiO2_P	IO	Serial Communication Interface Slave Side, Positive Polarity
7	SiO2_N	IO	Serial Communication Interface Slave Side, Negative Polarity
8	VCC_5V	Supply	5V Supply

3.2. NXP S32K1xx Series Microcontroller Features

The S32K1xx family of 32-bit AEC-Q100 qualified MCUs combines a breakthrough suite of production-grade tools and software with a scalable family of Arm Cortex-M based MCUs built on future-proof features.

S32K1xx family comes with a variant of memory size, pin-out, and key features. Some of the brief features:

S32K116 - 48MHz + DMA + FlexIO + ISELED + 128k Flash

S32K118 - 48MHz + DMA + FlexIO + ISELED + 256k Flash

S32K142 - 112MHz + DMA + FlexIO + ISELED + 256k Flash

S32K144 - 112MHz + DMA + FlexIO + ISELED + 512k Flash

S32K146 - 112MHz + DMA + FlexIO + ISELED + 1MB Flash

S32K148 - 112MHz + DMA + FlexIO + ISELED + 2MB Flash

Standard S32K1xx parts will NOT support ISELED features in production mode. There is special ISELED S32K1xx Part Number offering which need to use together with ISELED SDK Driver. Please consult your nearest NXP/Dominant Sales representative to enquire S32K1xx ISELED Part Offering.

3.3. Basic External Component guide

Figure 1 shows a typical setup layout for seddLED3.0 LED, connecting S32K NXP microcontroller to a series of seddLED3.0 LEDs.

- Each device 5V_LED (pin 8) will need to be connected with 1uF low ESR ceramic capacitor. Example: GRT033R60J105ME13D, 0201, 1uF, 6.3V, +/- 20%, X5R. Please place capacitor as close as possible to the target seddLED3.0 LED.
- Incoming data (SIO1_P) and clock (SIO1_N) can be connected with ESD protection diode in parallel to GND, for protection against voltage spike to the SIO1_P and SIO1_N. Example: PESD5V0F1BL, ESD protection up to 10kV.
- First communication connection in between microcontroller to the seddLED3.0 is based on single-ended format. Thus, for each line, please add-on 1K pull-up resistor in parallel to the 5V supply.
- Last device's SIO2_P and SIO2_N can be terminated with ESD protection device as well.
- As the incoming supply must be kept within 4.5V – 5.5V, it's recommended to install 5.5V zener to protect against over-voltage.

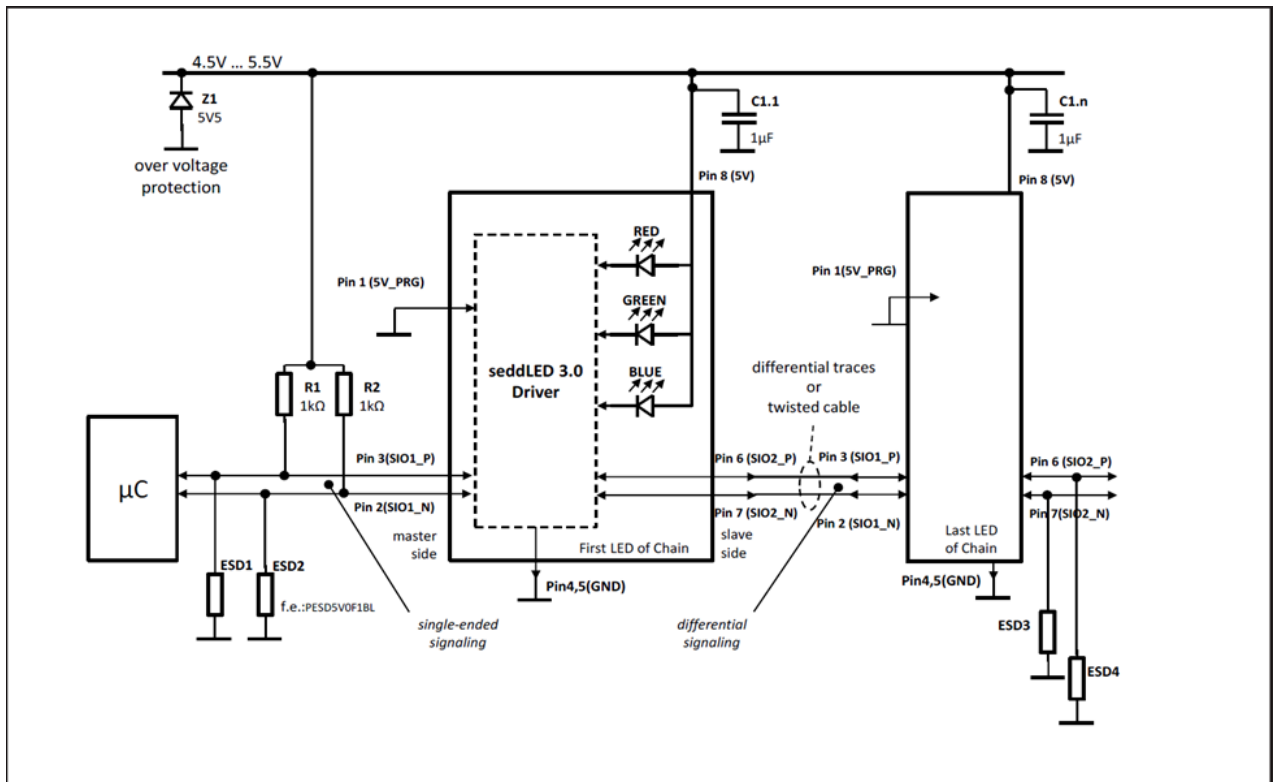


Figure 1: Typical Application Layout

3.4 PCB design guide, pin-per-pin explanation:

Figure 2 below shows a brief wiring setup from S32K1xx to the first and second seddLED3.0 LED.

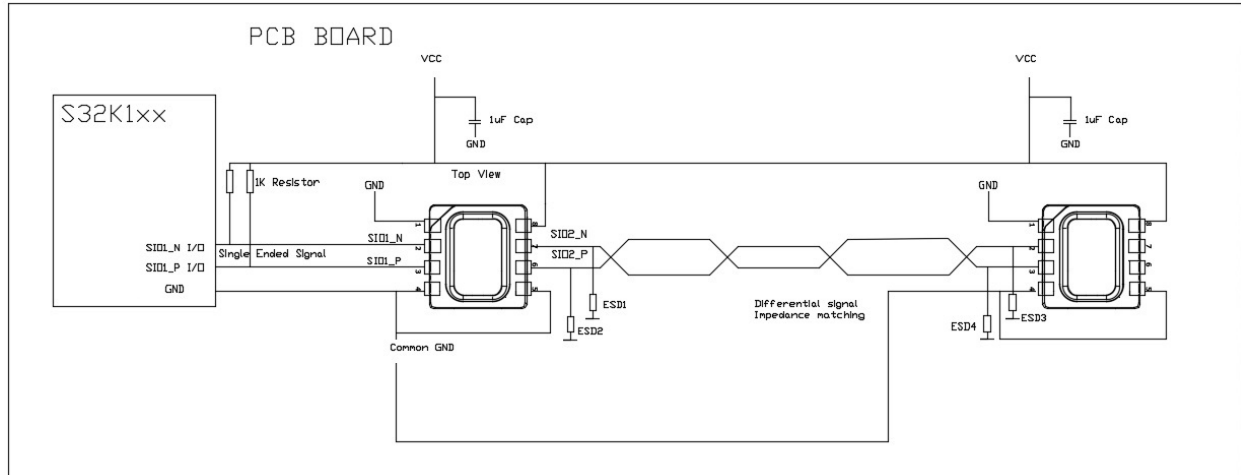


Figure 2: Wiring Setup Guidelines

seddLED3.0 LED needs to be powered up with 5V supply.

- There must be a common ground connecting seddLED3.0 and S32K1xx together.
- A dedicated ground plane is desirable as it offers lowest possible inductance for current to return to its source.
- Communication protocol from first device to microcontroller (both data and clock) is configured to be single-ended format. Most of the microcontroller I/O pins are configured to be single-ended TX/RX format. Thus, it is vulnerable to noise and interference. It's advisable to keep the connection short and best if first seddLED3.0 LED mounted on the same PCB board to the S32K1xx and communication connection is routed via impedance matching PCB trace.
- Keep both signal connection near to GND for ground shielding advantage.
- Connection from 1st seddLED3.0 LED to 2nd seddLED3.0 LED is configured with differential format. User does not need to worry about the conversion from single-ended format to differential format. All will be taken care by seddLED3.0 LED internal module.

For multiple module in vehicle installation, we recommend user to install a stand-alone, ISELED compatible, low cost microcontroller to each individual module. Each microcontroller should represent own PCB module slave connections.

Good suggestion will be NXP S32K11x/14x Arm Cortex M4+ and M0+ microcontroller.

Each microcontroller can interact with each other and connected to the automobile CAN/LIN Bus. NXP also offers a good range of CAN/LIN transceiver. E.g TJA1051 proposed for below typical 4 door panel seddLED 3.0 strip connected to NXP S32K1xx microcontroller, and routed to CAN/LIN BUS network.

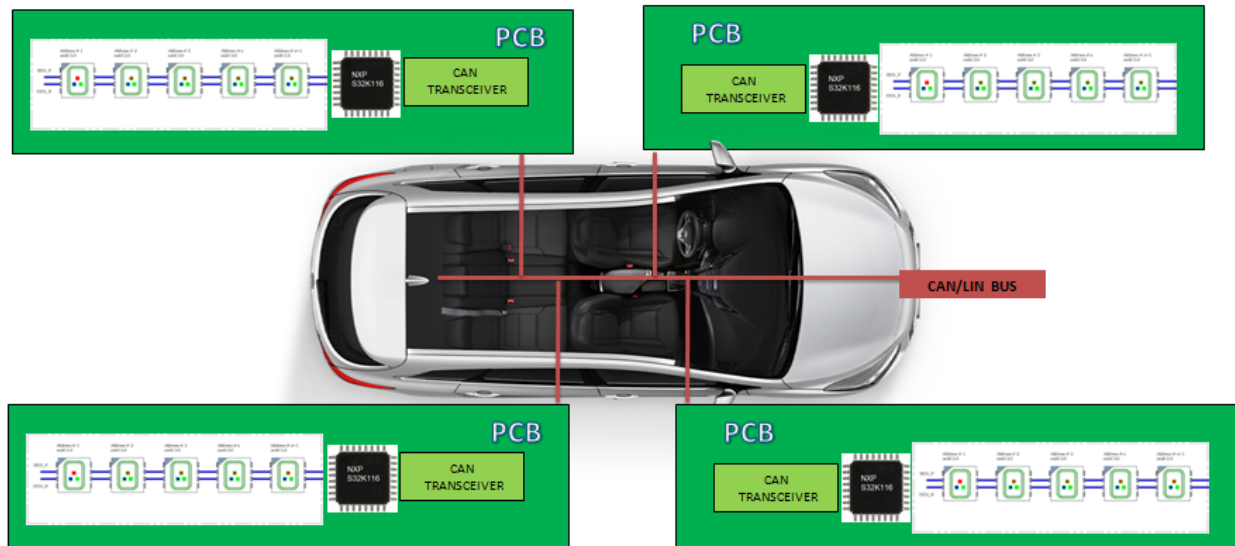


Figure 3: seddLED 3.0 Door Panel Strip with S32K1xx Microcontroller Connected to Integrated CAN/LIN Bus.

3.5 Extended connection guide (module-to-module) explanation

ISELED communication protocol is designed to enable ease of inter-component connection on the printed circuit board.

Within the same PCB, first seddLED3.0 LED and microcontroller has to be routed as close as possible as its single-ended communication format. Connection from 1st seddLED3.0 LED to 2nd seddLED3.0 LED onwards will be converted to differential format internally.

Under unavoidable circumstances where customer needs to connect from PCB module to PCB module, the length of the wire harness shall be kept not more than 40cm to ensure EMI robustness.

For the harness connection setup, it's advisable to use twisted-pair wire across SIOx_P and SIOx_N communication path.

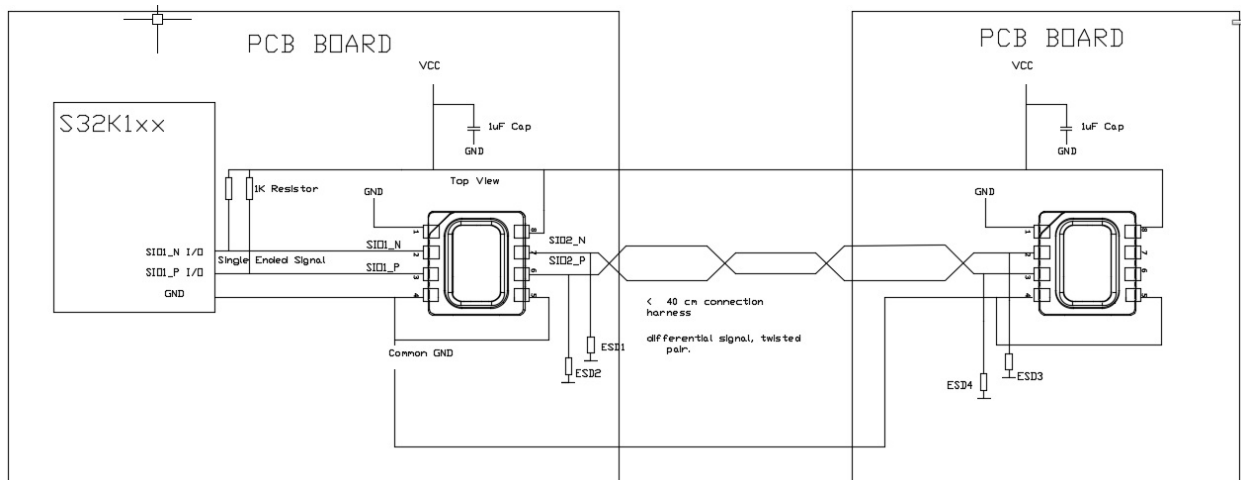
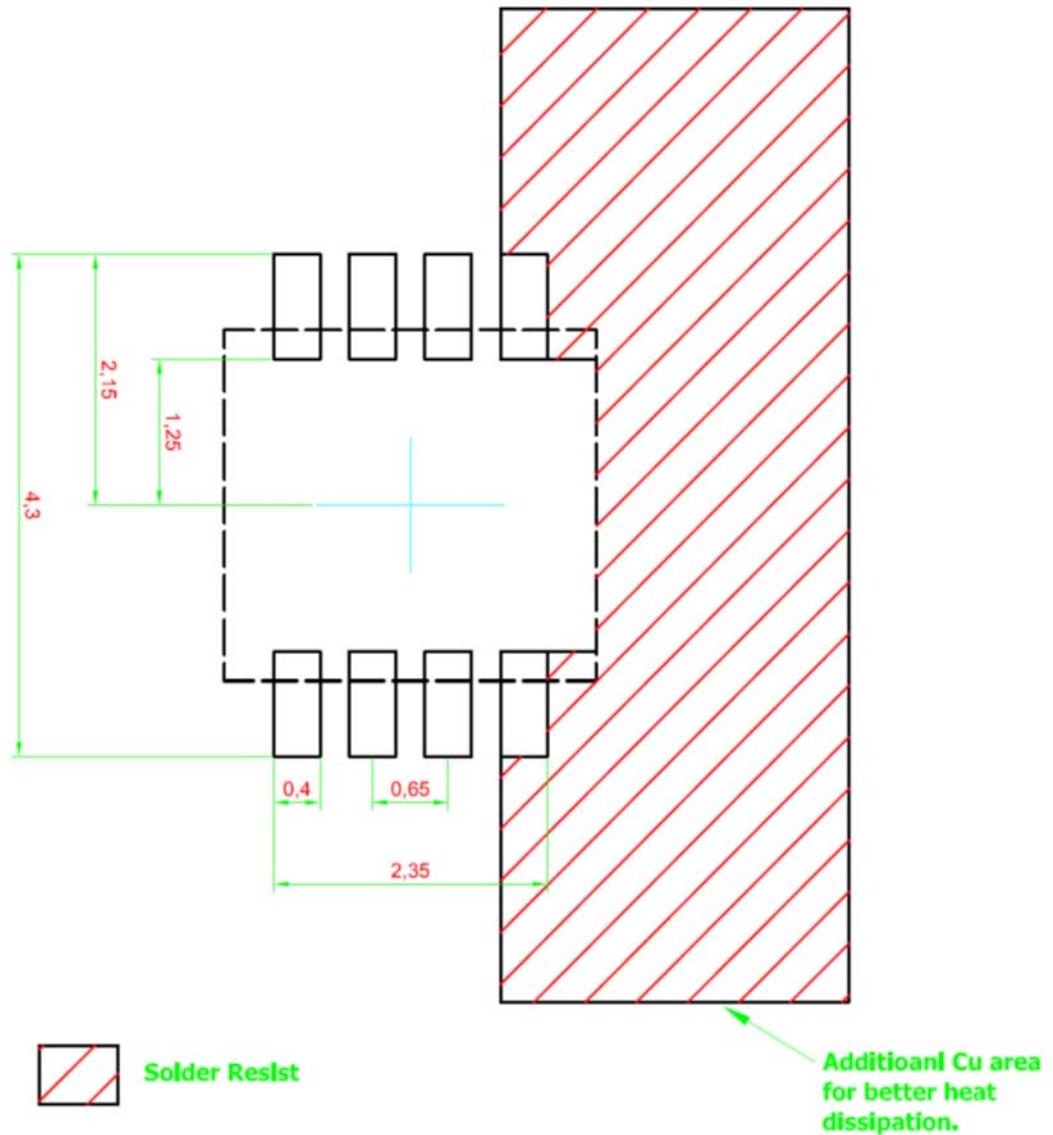


Figure 4: Module to Module connection via twisted pair harness

3.6 seddLED3.0 solder pad reference

It's advisable to connect thermal pad (pin 4 & pin 5) to the large copper surface and having this connected with multiple thru hole vias to the bottom copper pad.

Following is the recommended solder pad placement:



4. Revision History

Changes	Date
Initial Release	15 Jan 2019