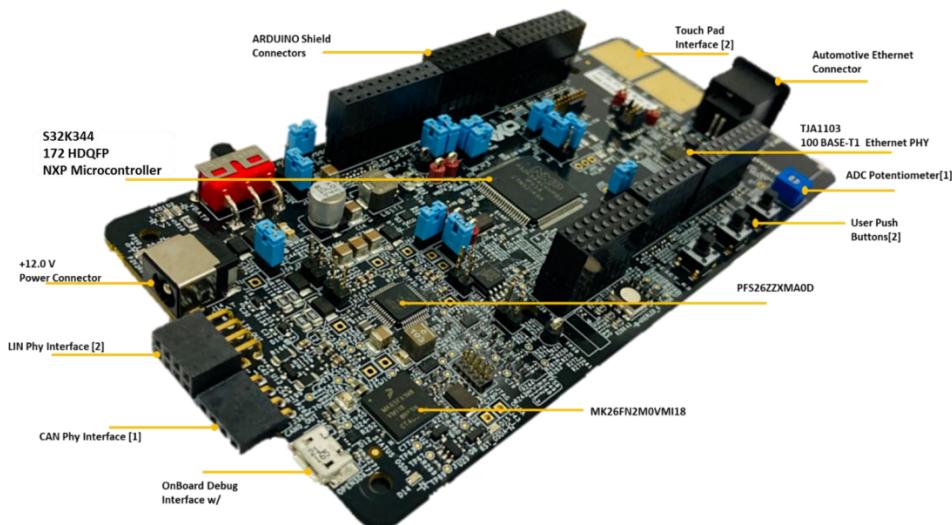


# HOWTO: Create a Blinking LED application project for S32K344 using S32 RTD AUTOSAR By KOUKI FEDI

Here will be a detailed guide to make a blinking LED from scratch on S32K344EVB-T172



## Installing S32 Design studio

1. You will find the link for S32 studio here : [Download](#)

**NXP** PRODUCTS APPLICATIONS DESIGN CENTER SUPPORT COMPANY

NXP > Design > Software Terms and Conditions

**Software & Support**

- [Product List](#)
- [Product Search](#)
- [Order History](#)
- [Recent Product Releases](#)
- [Recent Updates](#)

**Licensing**

- [License Lists](#)
- [Offline Activation](#)

**FAQ**

- [Download Help](#)
- [Table of Contents](#)
- [FAQs](#)

**Software Terms and Conditions**

**S32 Design Studio for S32 Platform v.3.6.0**

Please read the following agreement and click "I AGREE" at the bottom before downloading your software.

LA\_OPT\_NXP\_Software\_License v60 February 2025

IMPORTANT. Read the following NXP Software License Agreement ("Agreement") completely. By selecting the "I Accept" button at the end of this page, or by downloading, installing, or using the Licensed Software, you indicate that you accept the terms of the Agreement, and you acknowledge that you have the authority, for yourself or on behalf of your company, to bind your company to these terms. You may then download or install the file. In the event of a conflict between the terms of this Agreement and any license terms and conditions for NXP's proprietary software embedded anywhere in the Licensed Software file, the terms of this Agreement shall control. If a separate license agreement for the Licensed Software has been signed by you and NXP, then that agreement shall govern your use of the Licensed Software and shall supersede this Agreement.

[NXP SOFTWARE LICENSE AGREEMENT](#)

This is a legal agreement between your employer, of which you are an authorized representative, or, if you have no

**I Agree** **Cancel**

2. Press **I Agree**

## Software &amp; Support

[Product List](#)
[Product Search](#)
[Order History](#)
[Recent Product Releases](#)
[Recent Updates](#)

## Licensing

[License Lists](#)
[Offline Activation](#)

## FAQ

[Download Help](#)
[Table of Contents](#)
[FAQs](#)

## Product Download

### S32 Design Studio for S32 Platform v.3.6.0

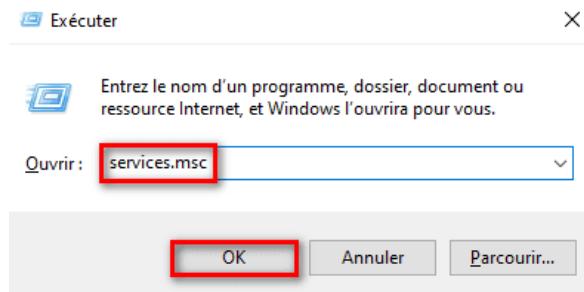
[Files](#)
[License Keys](#)
[Notes](#)
[? Download Help](#)
[Show All Files](#)

4 Files

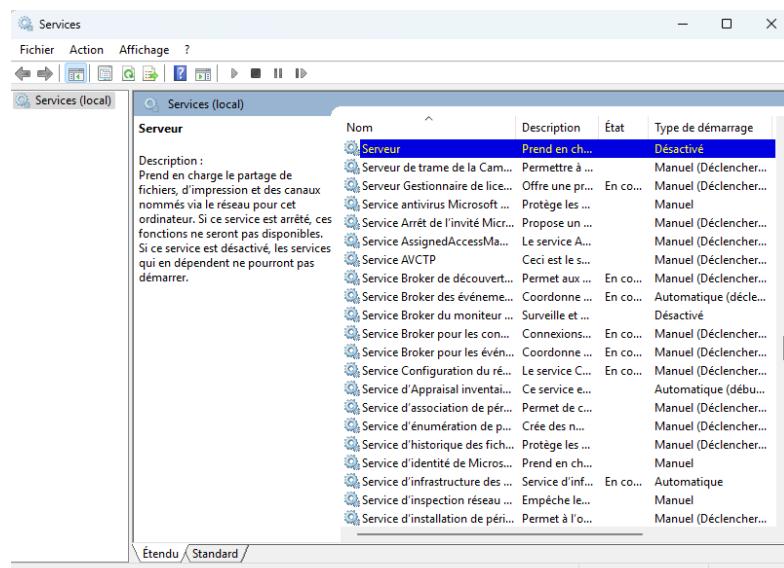
+ File Description	File Size	File Name
+ S32 Design Studio 3.6.0 Release Notes	77 KB	<a href="#">S32DS_3.6.0_Release_Notes.pdf</a>
+ S32 Design Studio 3.6.0 Software Content Register	23.1 KB	<a href="#">SCR_S32DS_3.6.0.txt</a>
+ S32 Design Studio v3.6.0 Linux installer	1.9 GB	<a href="#">S32DS_3.6.0_linux.x86_64.bin</a>
+ S32 Design Studio v3.6.0 Windows installer	2.6 GB	<a href="#">S32DS_3.6.0_win32.x86_64.exe</a>

### 3. Download **S32DS\_3.6.0\_win32.x86\_64.exe**

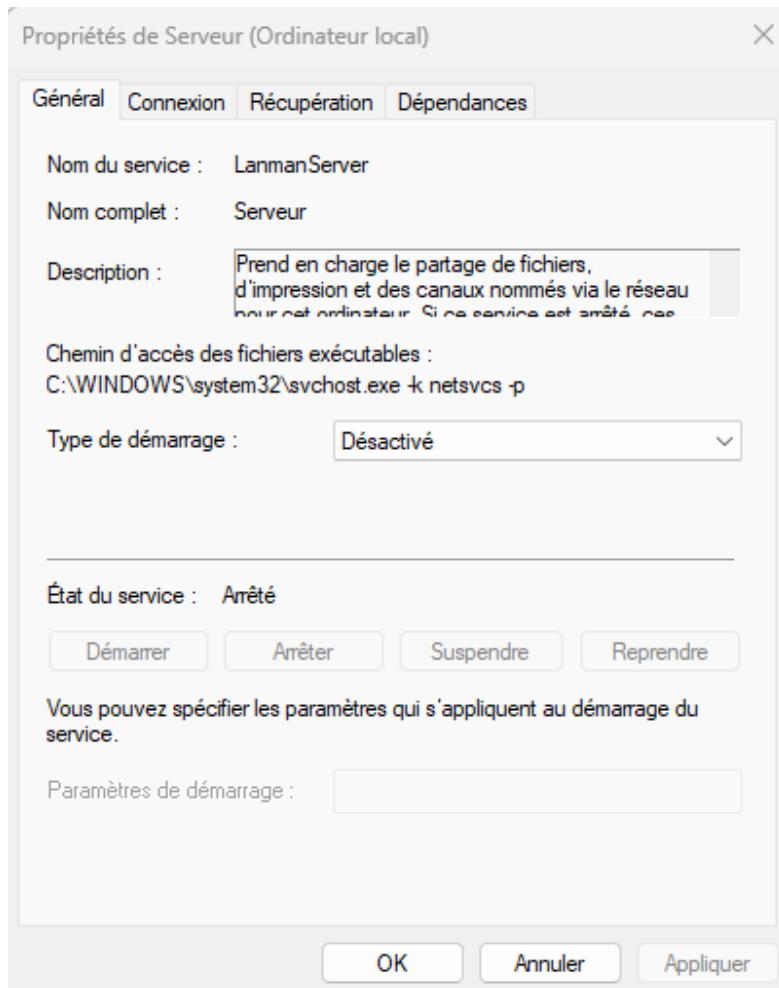
### 4. Before Runnig the installation Press : **Windows+R** then Search for **services.msc**



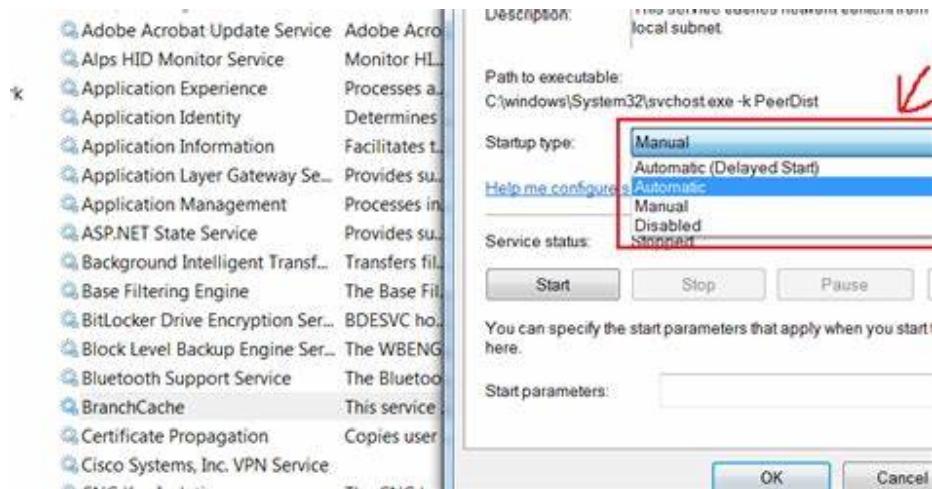
### 5. Search for **Server (Serveur)**



### 6. Right CLick on Server -> Properties ( Propriétés )

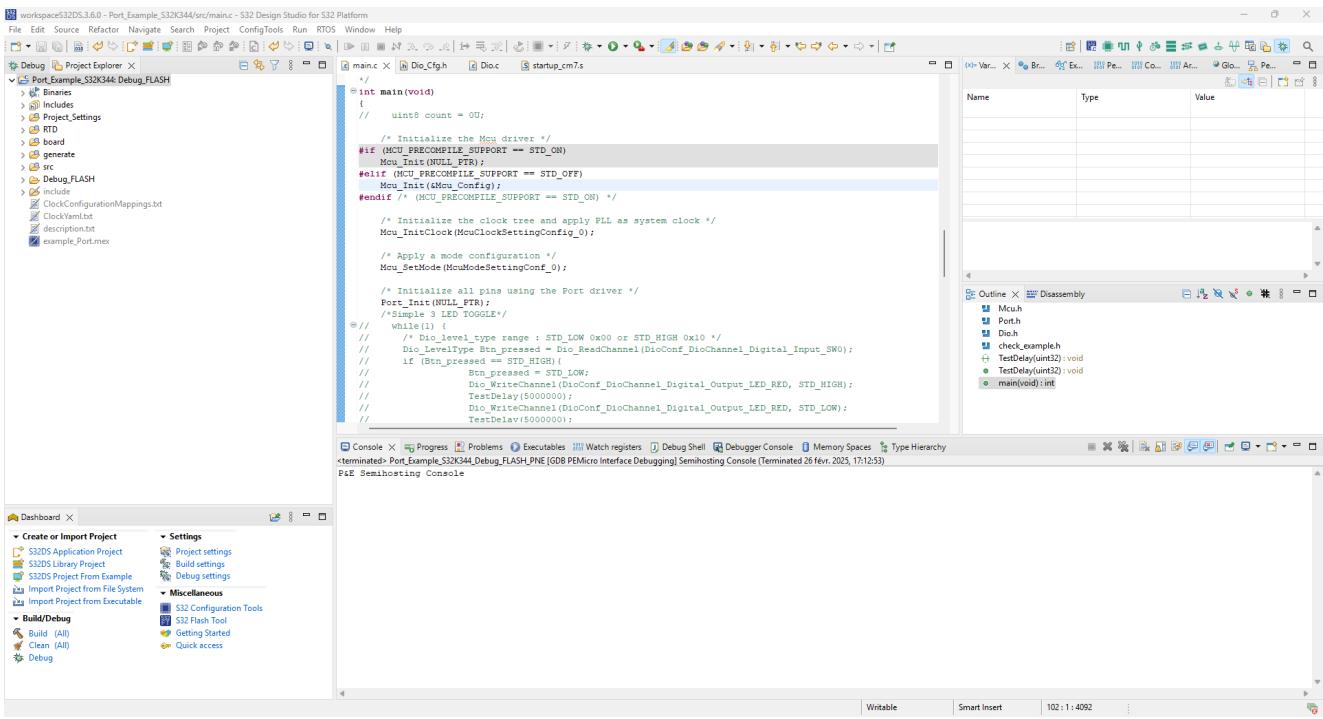


#### 7. Startup Type (Type de démarrage) :



#### 8. Know go and Run **S32DS\_3.6.0\_win32.x86\_64.exe**

9. If Everything work properly you will get this window after launching s32 studio

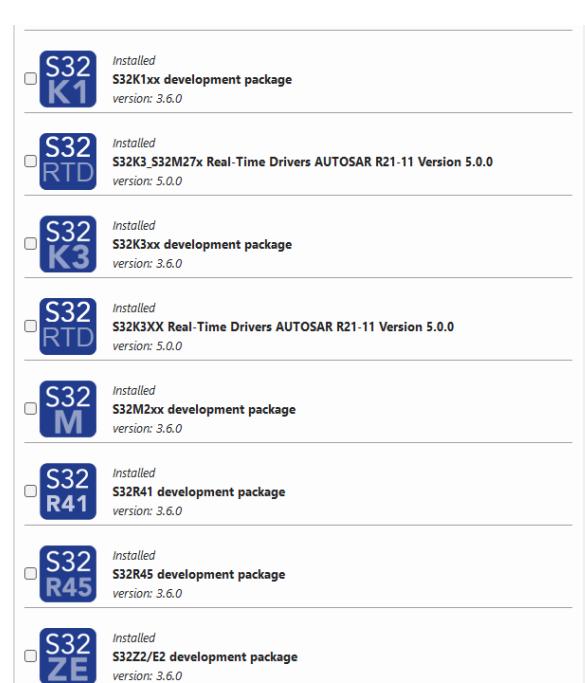
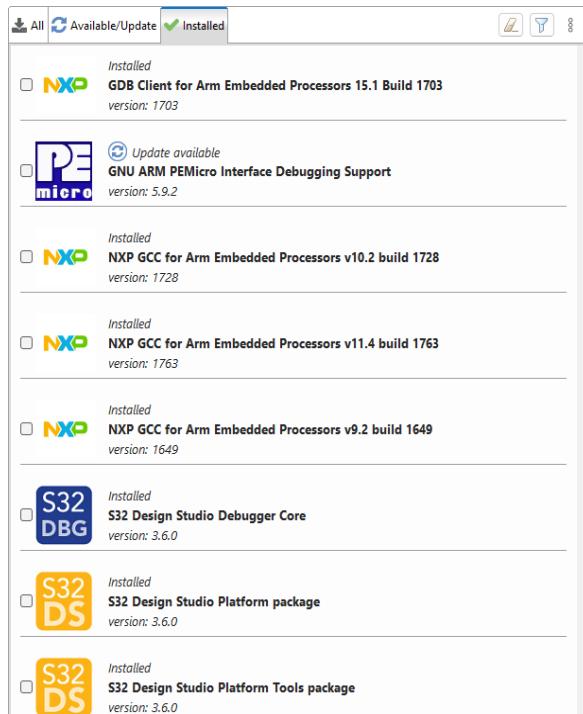


# Preparing The s32 Studio

We need to install some packages on S32 Studio

1. Go to **HELP**

2. Go to **S32ds Extensions and Update**



3. For **S32K3XX Real-TimeDrivers AUTOSAR R21-11 Version 5.0.0 Package** You need to add it manually

4. Go to RTD NXP from this link : [Automotive SW - S32K3/S32M27x - Real-Time Drivers for Cortex-M](#)

## 5. Download **SW32K3\_S32M27x\_RTD\_4.4\_4.0.0\_P24\_D2405.exe** and **SW32K3\_S32M27x\_RTD\_4.4\_4.0.0\_P24\_DS\_updatesite\_D2405.zip**

The screenshot shows the NXP Product Download page for the S32K3\_S32M27x Real-Time Drivers AUTOSAR 4.4 Version 4.0.0 P24. The 'Files' tab is selected, displaying a table of files:

File Description	File Size	File Name
SW32K3_S32M27x_RTD_4.4_4.0.0_P24_D2405.exe	84.4 MB	SW32K3_S32M27x_RTD_4.4_4.0.0_P24_D2405.exe
SW32K3_S32M27x_RTD_4.4_4.0.0_P24_D2405_QualityPackage.zip	4.1 MB	SW32K3_S32M27x_RTD_4.4_4.0.0_P24_D2405_QualityPackage.zip
SW32K3_S32M27x_RTD_4.4_4.0.0_P24_D2405_ReleaseNotes.pdf	165.3 KB	SW32K3_S32M27x_RTD_4.4_4.0.0_P24_D2405_ReleaseNotes.pdf
SW32K3_S32M27x_RTD_4.4_4.0.0_P24_D2405_ReleaseNotes_Updated_D241202.pdf	165.4 KB	SW32K3_S32M27x_RTD_4.4_4.0.0_P24_D2405_ReleaseNotes_Updated_D241202.pdf
SW32K3_S32M27x_RTD_4.4_4.0.0_P24_D2405_SafetyPackage.zip	213 KB	SW32K3_S32M27x_RTD_4.4_4.0.0_P24_D2405_SafetyPackage.zip
SW32K3_S32M27x_RTD_4.4_4.0.0_P24_D2405_SCR.txt	1.7 KB	SW32K3_S32M27x_RTD_4.4_4.0.0_P24_D2405_SCR.txt
SW32K3_S32M27x_RTD_4.4_4.0.0_P24_DS_updateSite_D2405.zip	213.8 MB	SW32K3_S32M27x_RTD_4.4_4.0.0_P24_DS_updateSite_D2405.zip

## 6. IF In the installation you need the License that you can find under **License Keys Panel**

The screenshot shows the NXP Product Download page for the S32K3\_S32M27x Real-Time Drivers AUTOSAR 4.4 Version 4.0.0 P24. The 'License Keys' tab is selected, displaying a note:

If there is no serial number information displayed here, your product does not require a license key.  
Contact NXP Semiconductors directly for more information.

## 7. Then GO S32 Studio and in **S32ds Extensions and Update** Press **Add Update Sites**

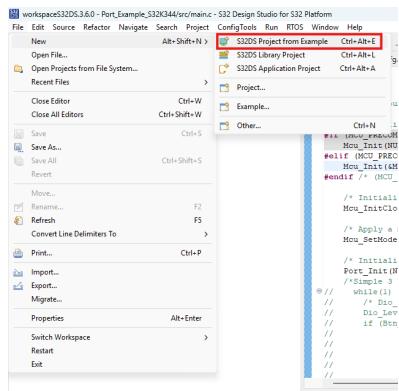
The screenshot shows the S32ds Extensions and Updates interface. The 'BBE31 DSP Add-On Package for S32R41' is selected for download, with the status 'Available' and 'Installed'. The interface includes a search bar and a list of other available packages like S32K1, S32K2, S32K3, S32M, S32R41, S32R45, and S32Z.

## 8. Ten look for **SW32K3\_S32M27x\_RTD\_4.4\_4.0.0\_P24\_DS\_updatesite\_D2405.zip**

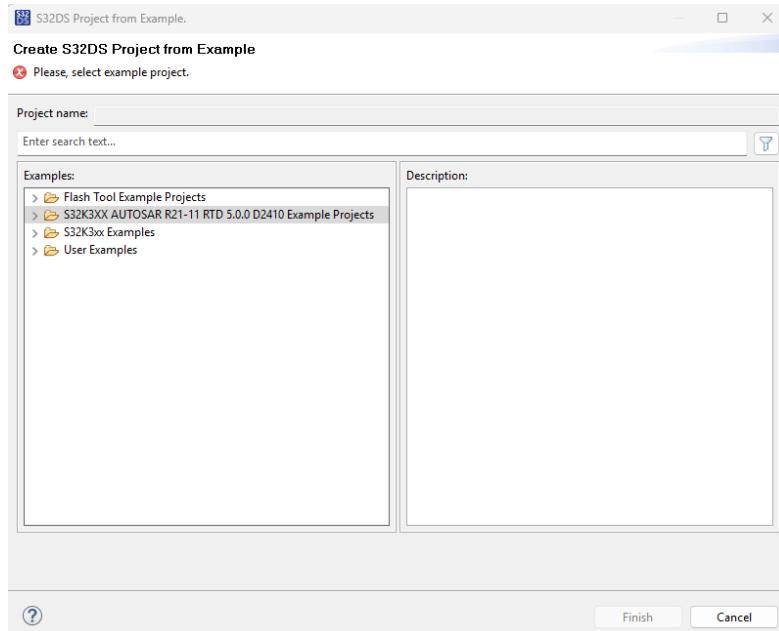
9. Finally the **S32K3XX Real-TimeDrivers AUTOSAR R21-11 Version 5.0.0 Package** Should appear

## Let's Start Our application

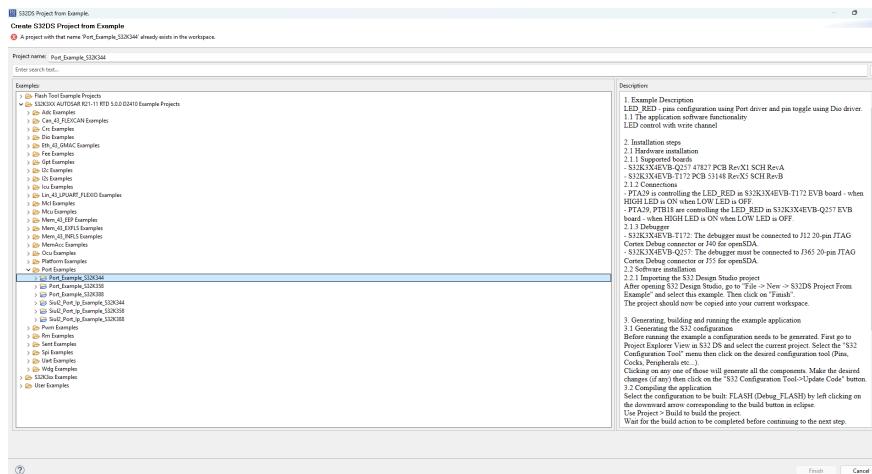
1. Open S32 Design Studio
2. Go to **File > New > S32DS Project From Example**



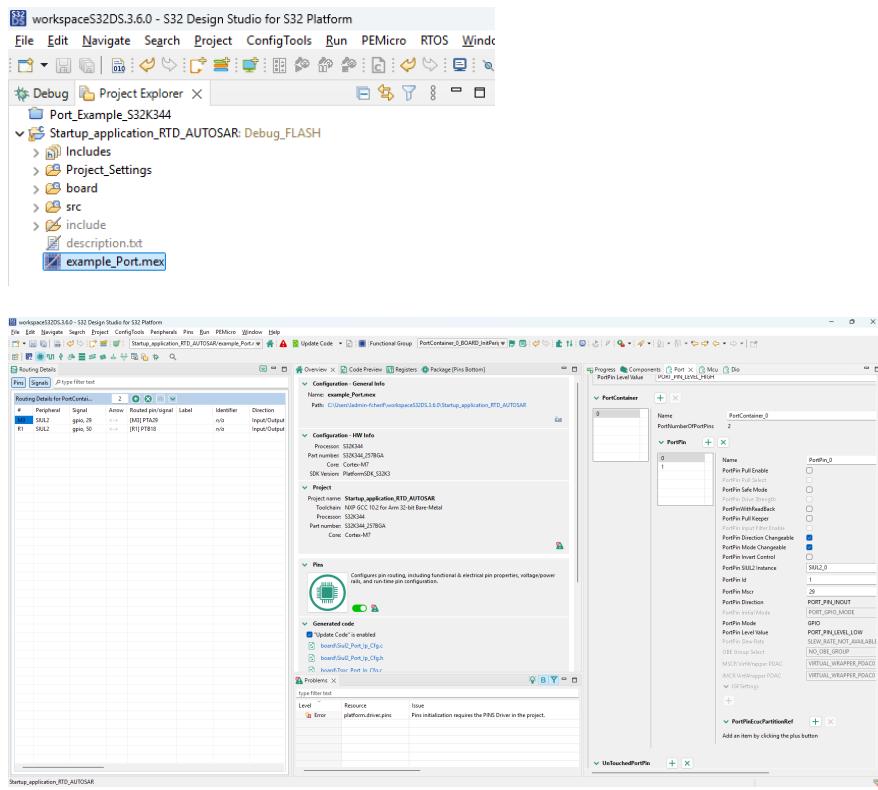
### 3. This window should appear



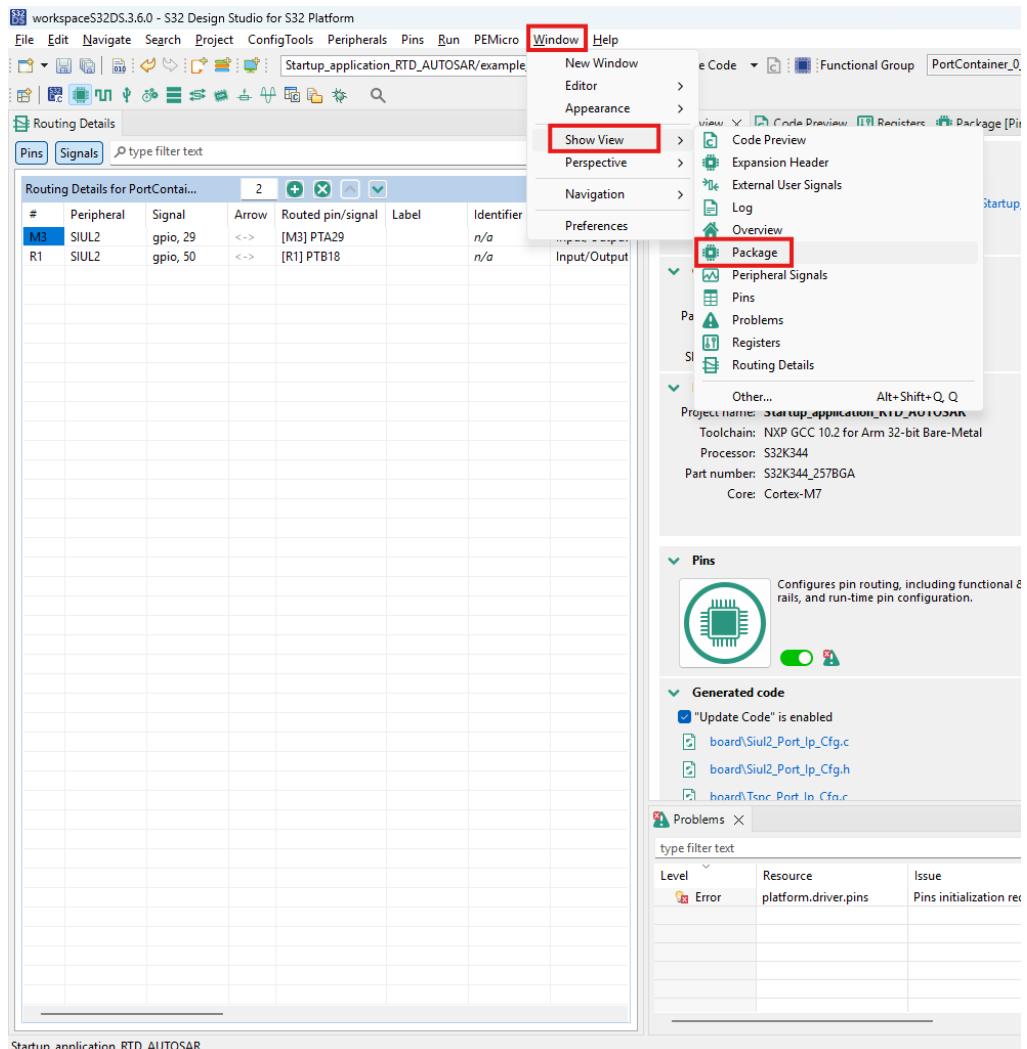
### 4. Look for Port\_Example



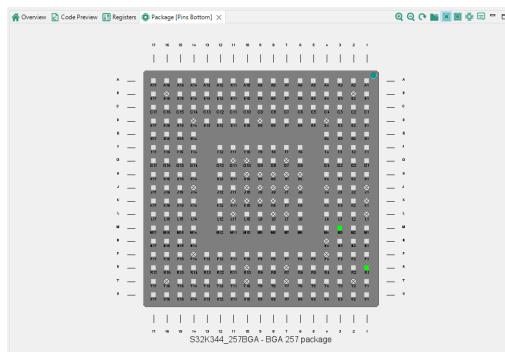
### 5. Open \*\*mex file\*\*



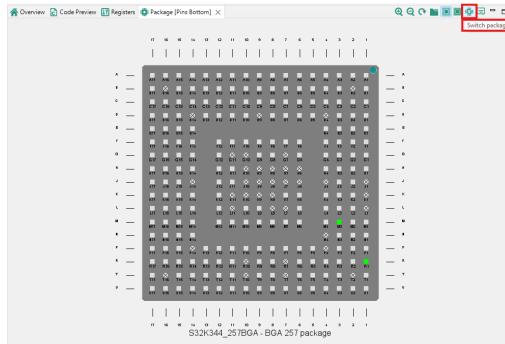
## 6. We need now to specify our Board , Go to Window > Show View > Package



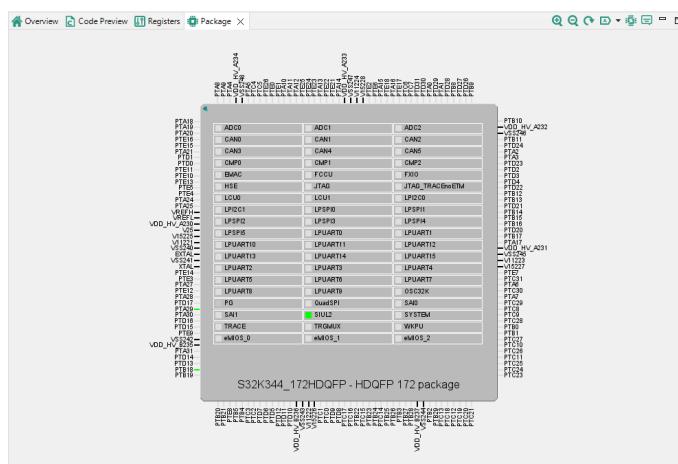
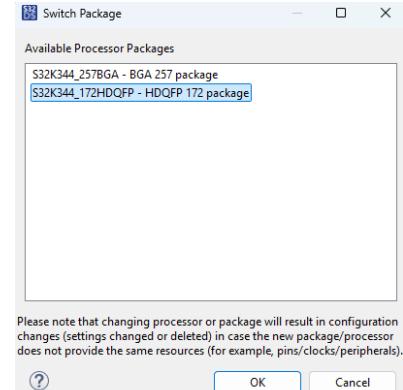
7. this window should appear



8. Go to **Switch Package**

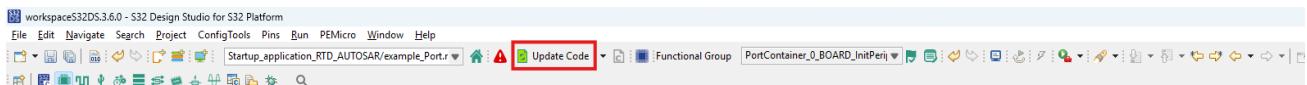


Select the S32K344\_172HDQFP - HDQFP 172 package because we are working with this board

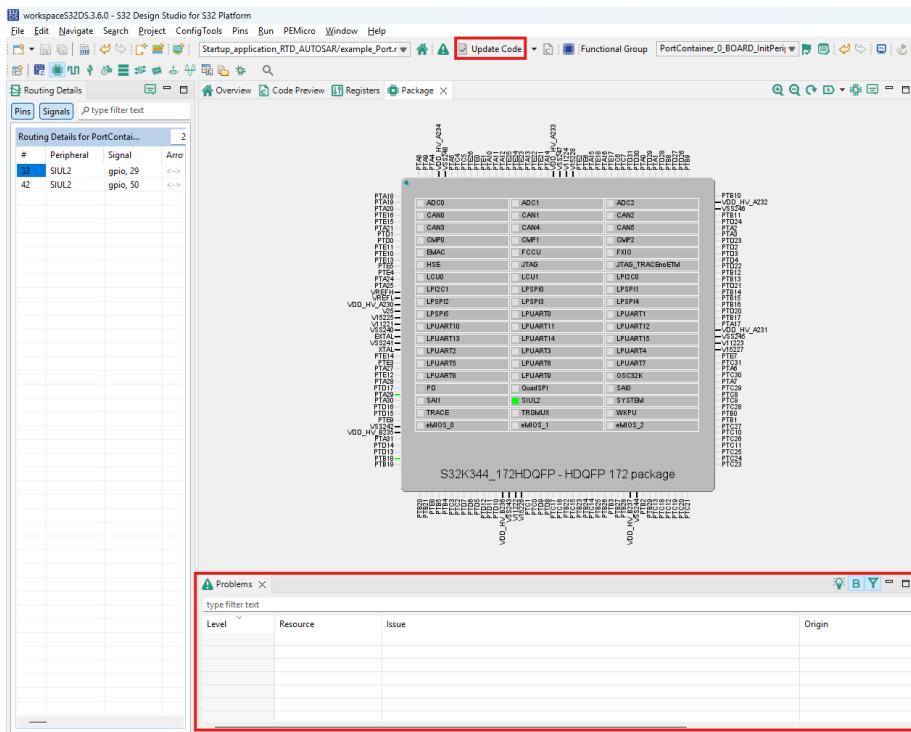


# Let's Build this Code

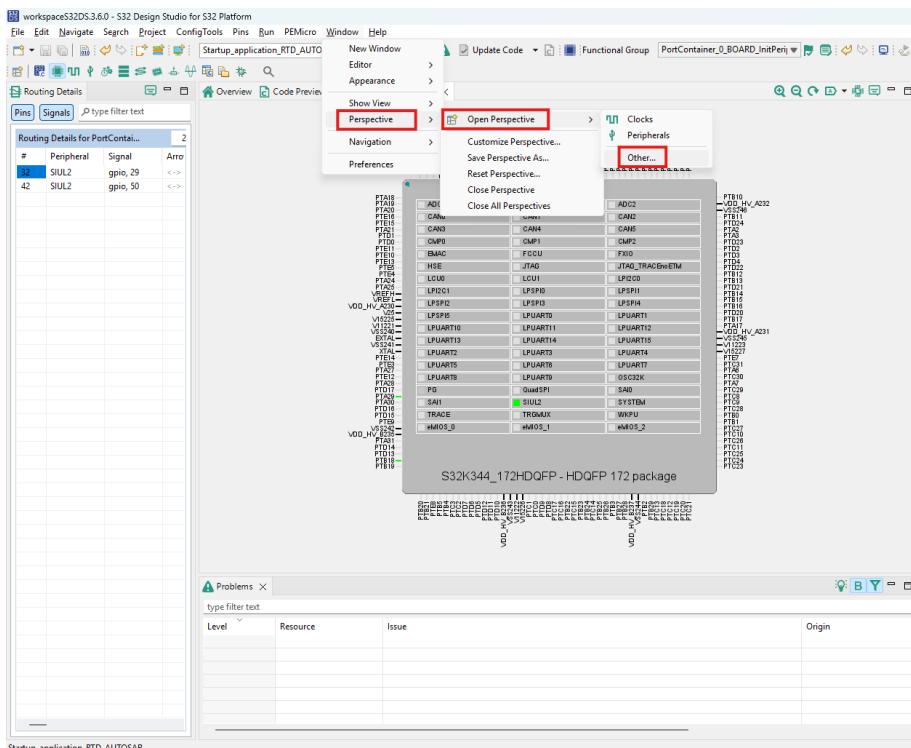
## 1. First, Update the Code



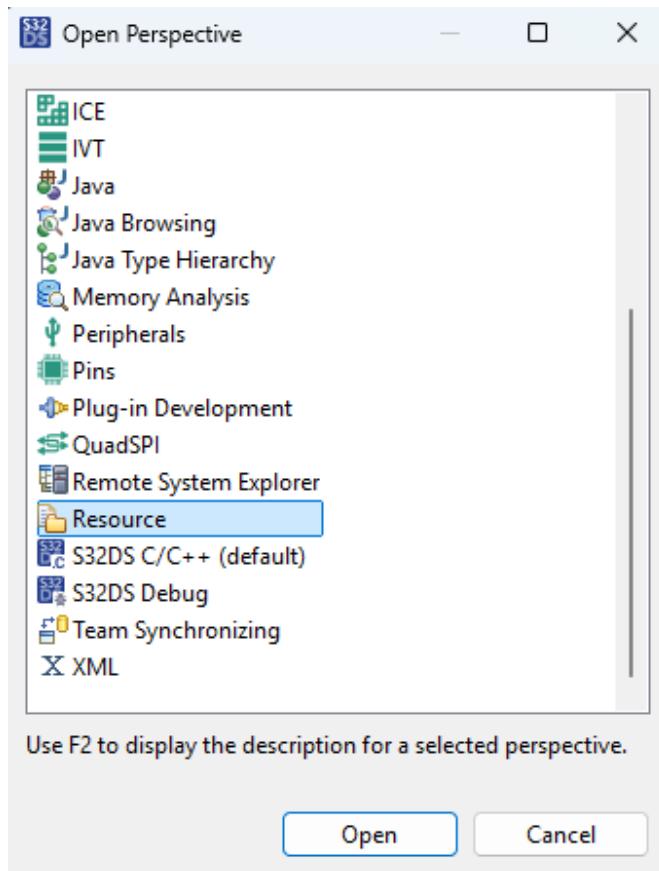
2. If every thing is OK, you should have a **GRAY Icon** on **Update Code** and **0 Problem** in **Problems Panel**



3. Now We should Go to **Resource**, If the icon is not already existed follow this steps

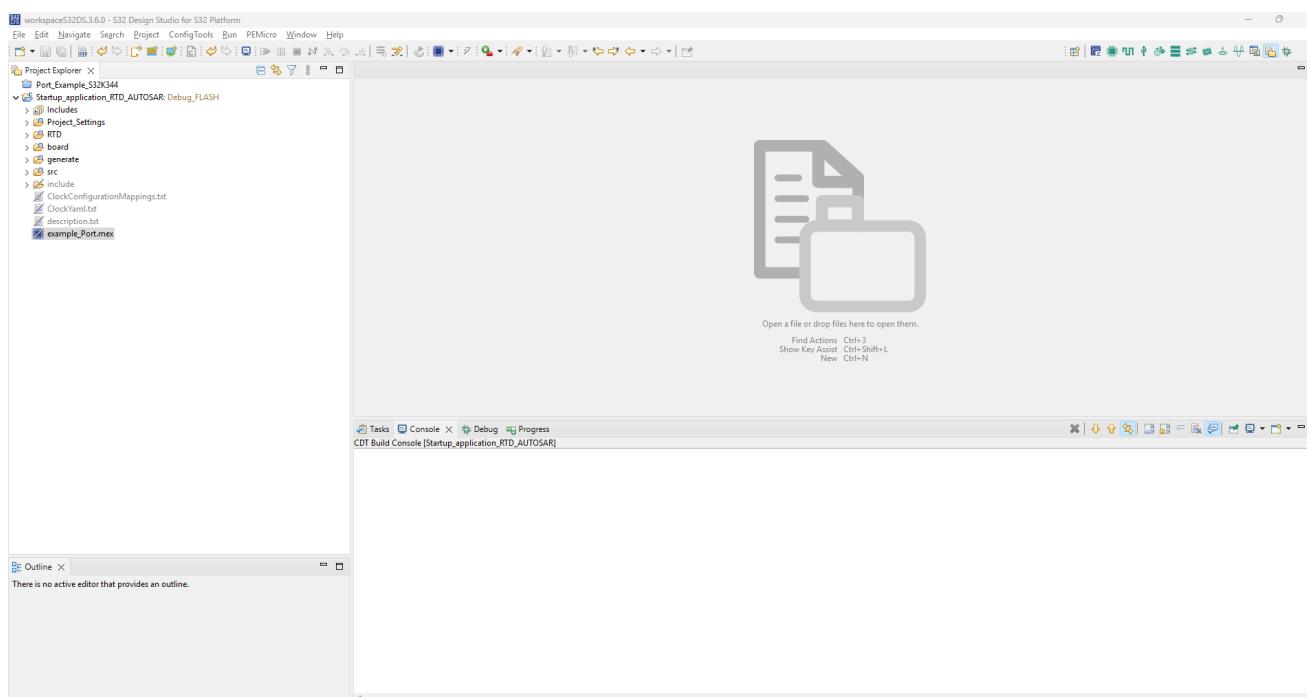


Select Resource and Click open

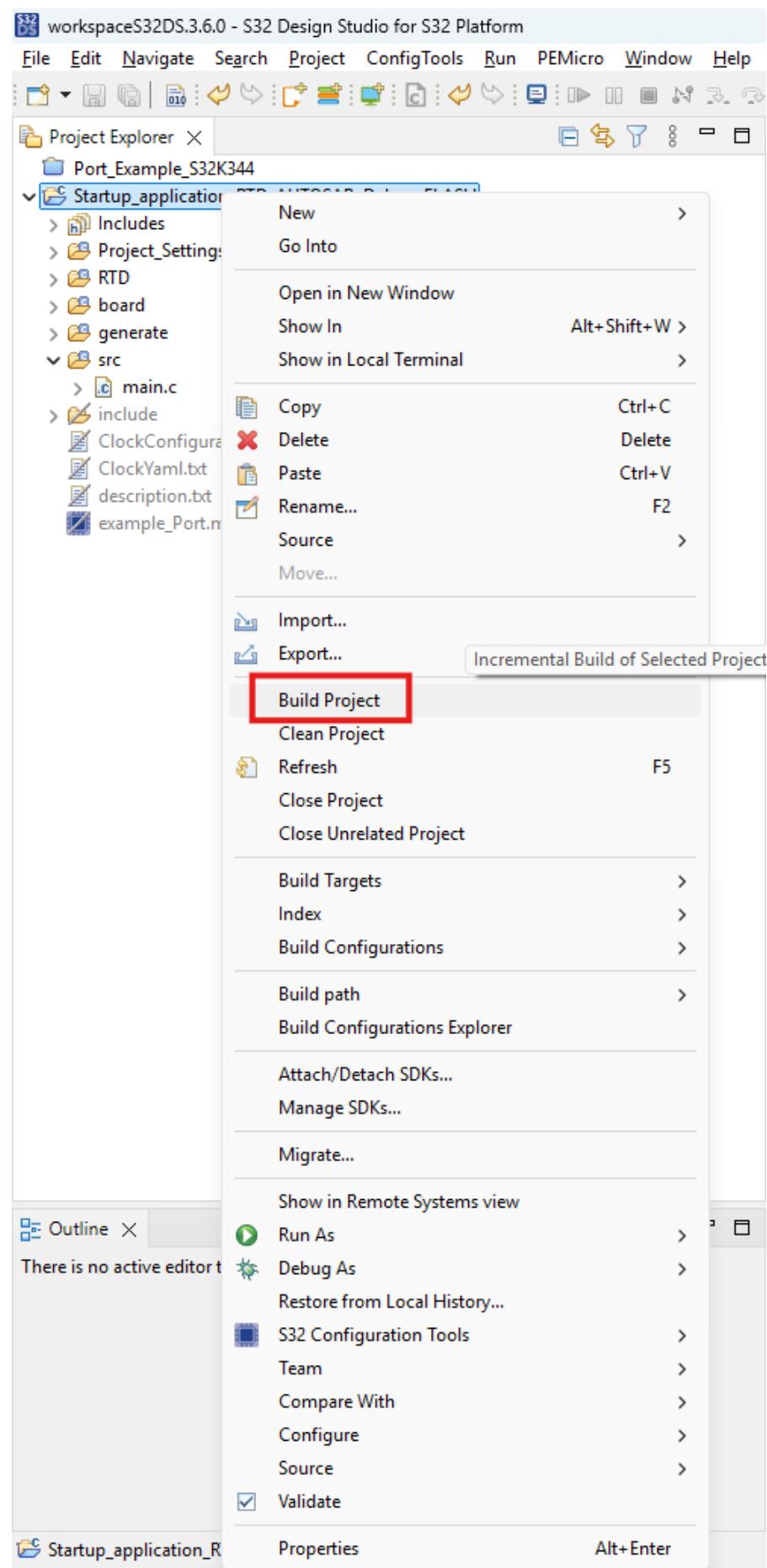


Use F2 to display the description for a selected perspective.

4. You should get something like this

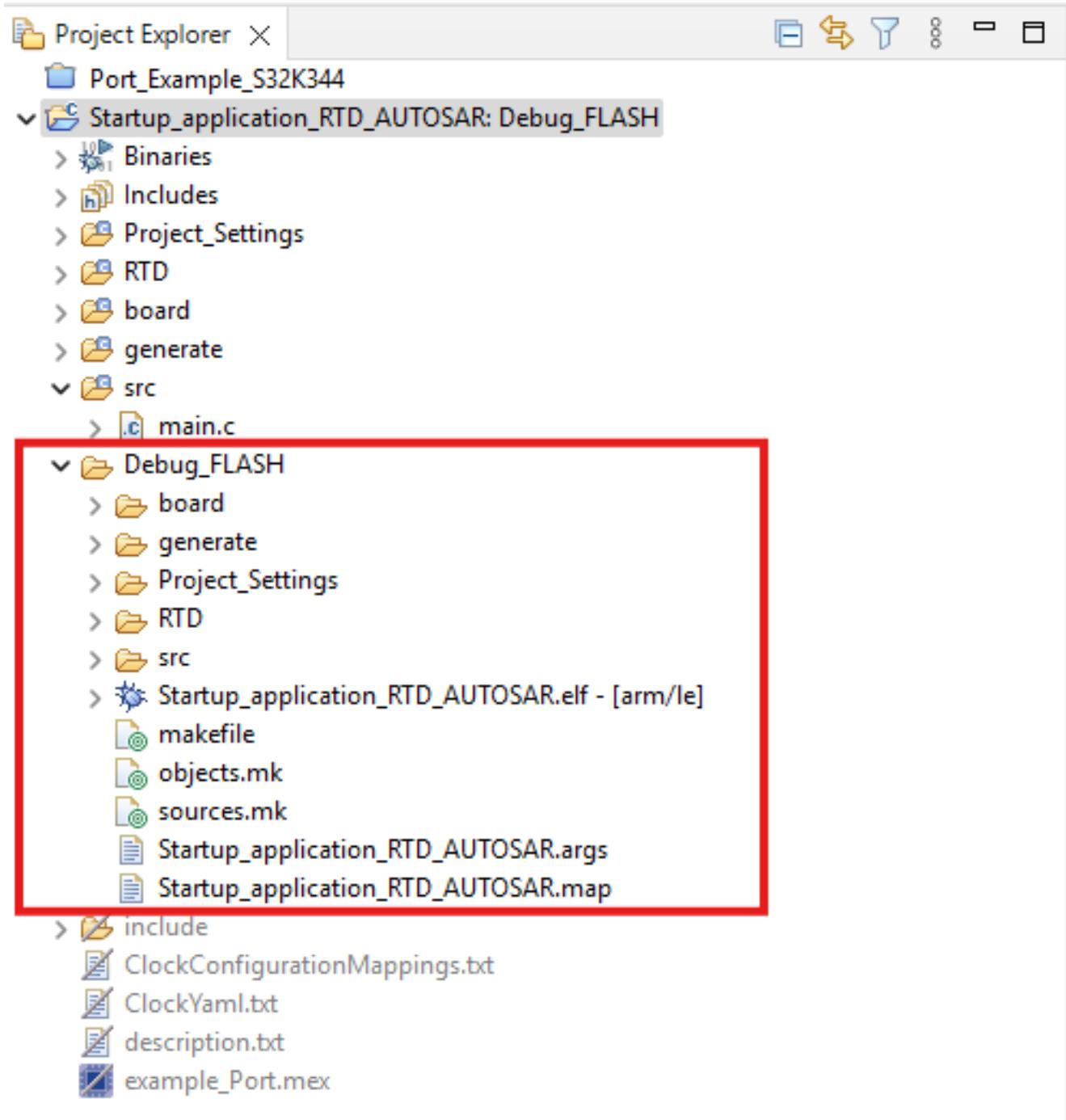


5. Right click On the **project name** in **Project Explorer Panel** then **Build Project**



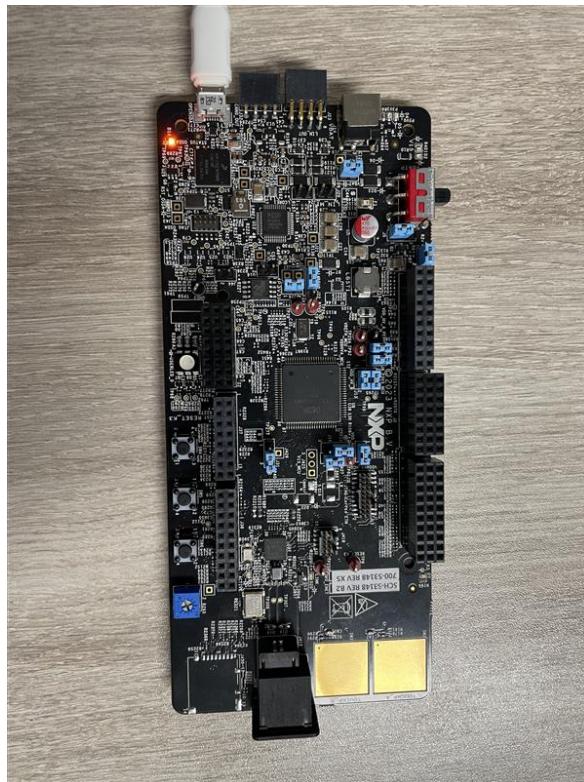
6. If Everything is Ok

## A Debug\_FLASH Should be Created

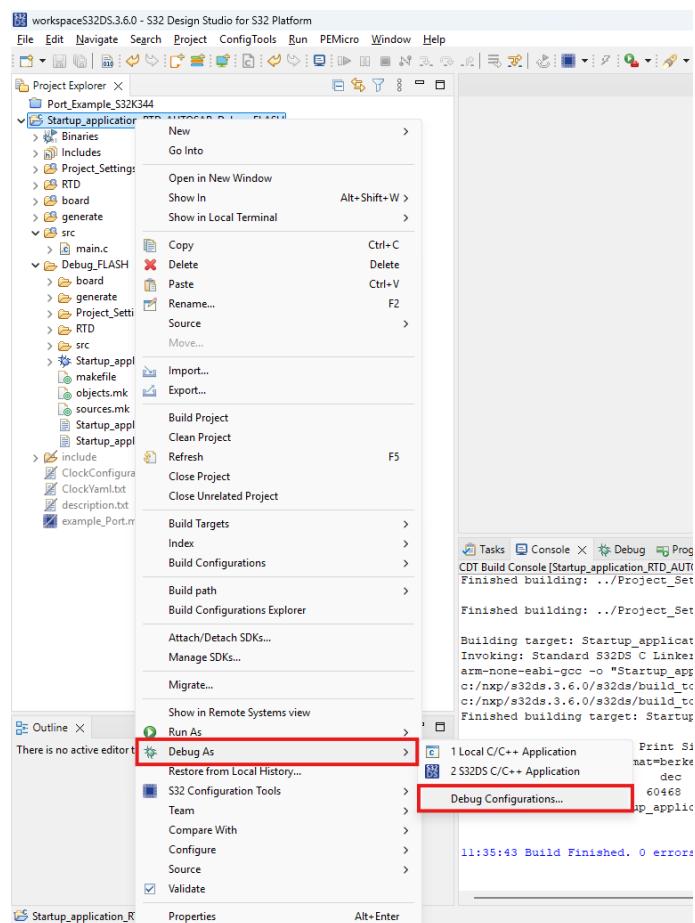


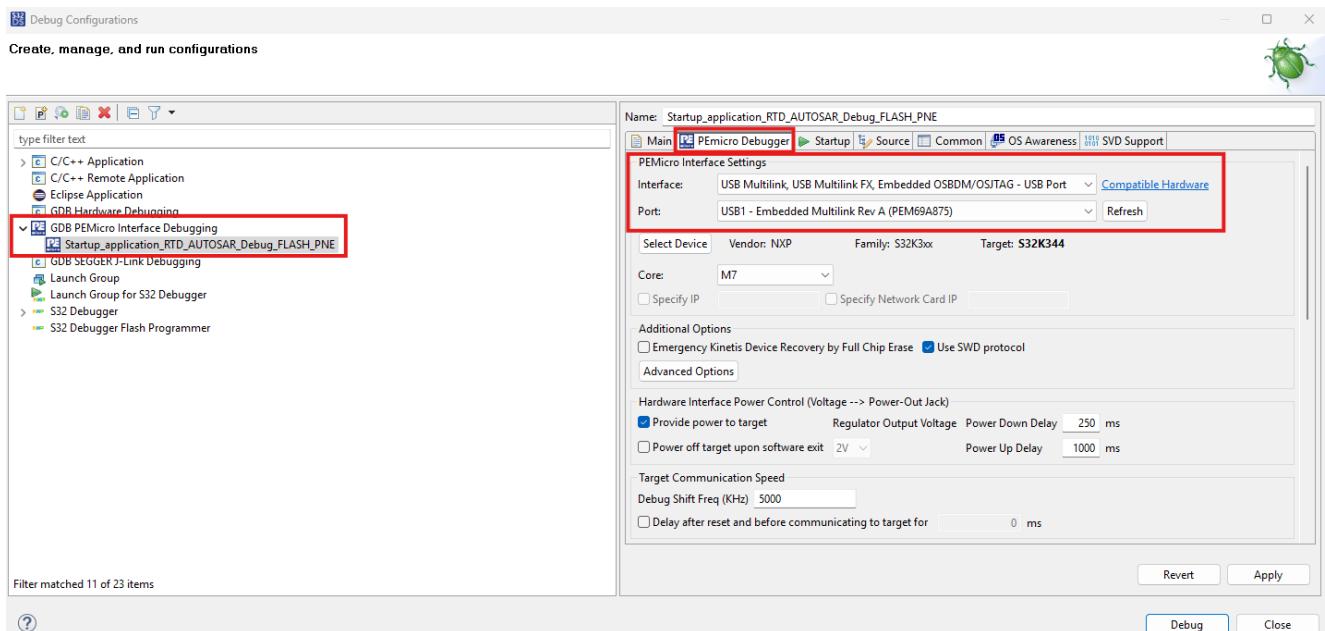
## 7. Now Let's prepare for debug

#### 8. Connect the USB to **S32K3x4EVB**

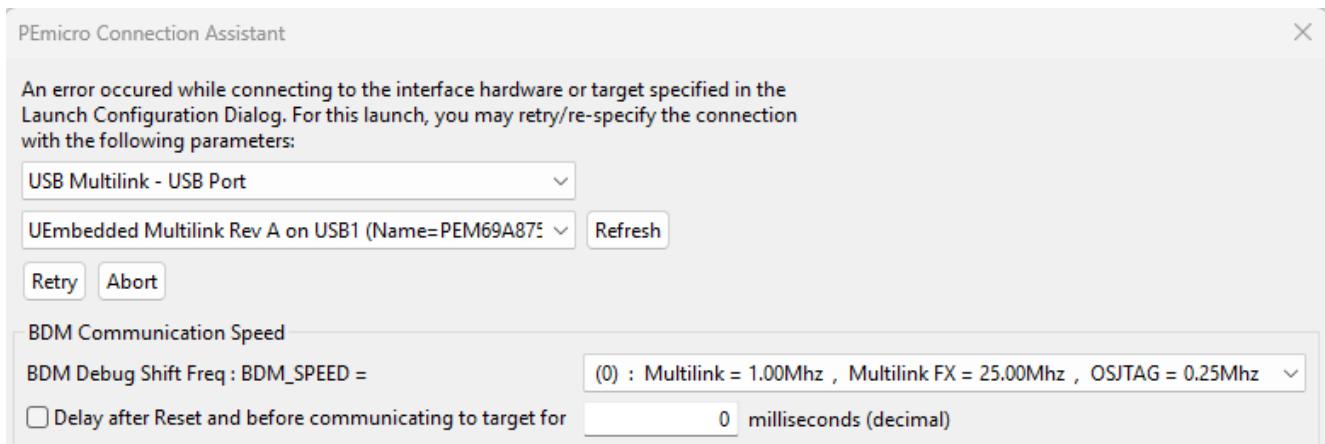


## 9. Right Click on the project





10. Then Press **Debug** This error should appear

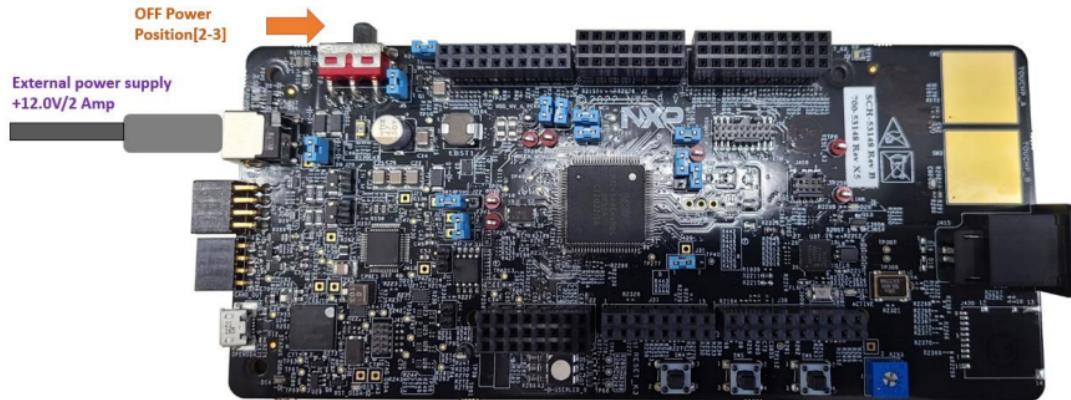


11. Plug the **External power Supply ( 12 V , >= 2.0 A )** to the Evaluation Board

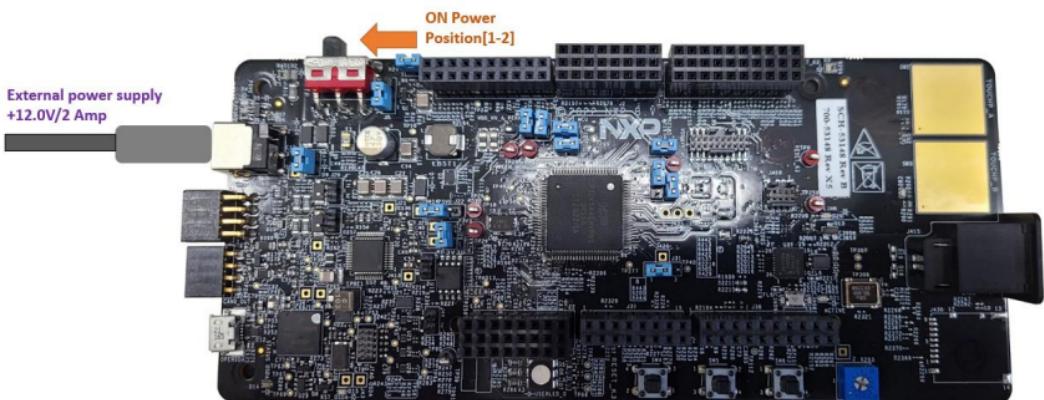


## 12. Turn on the Switcher Like this

Once the power switch SW1 is in OFF-position, then the EVB can be connected to an external power supply.



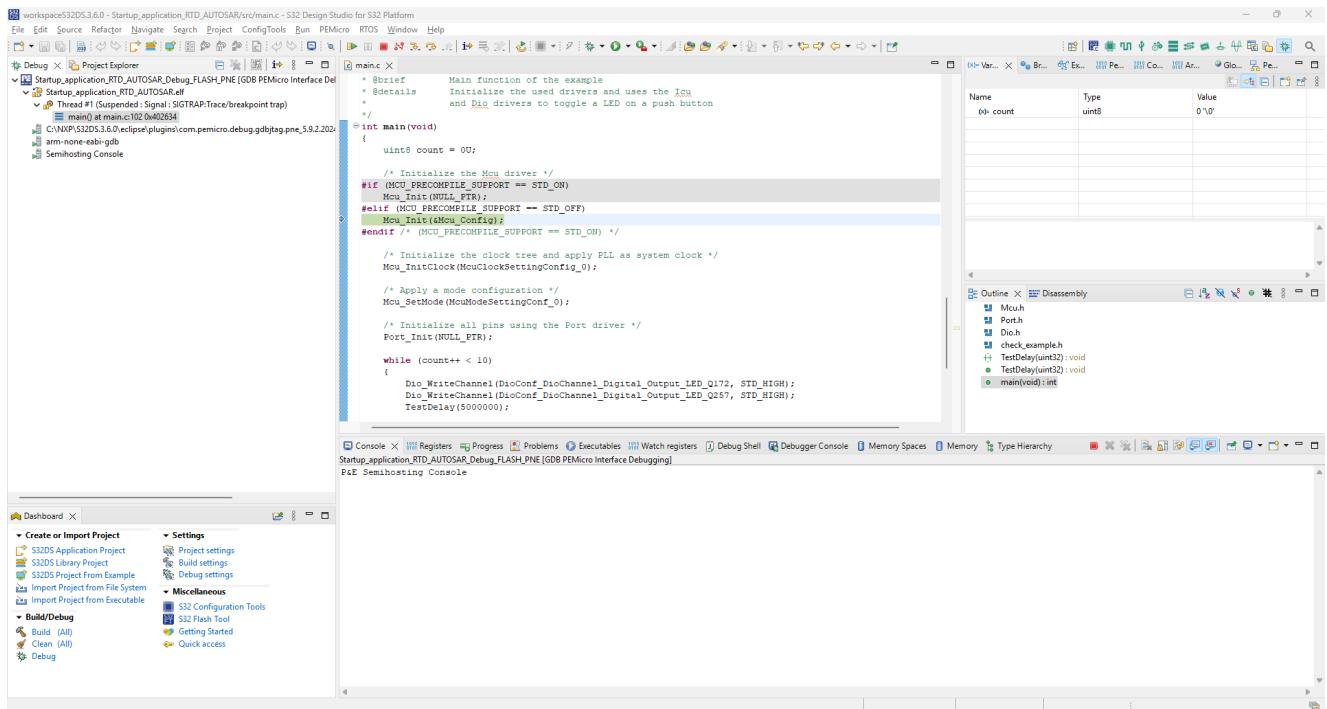
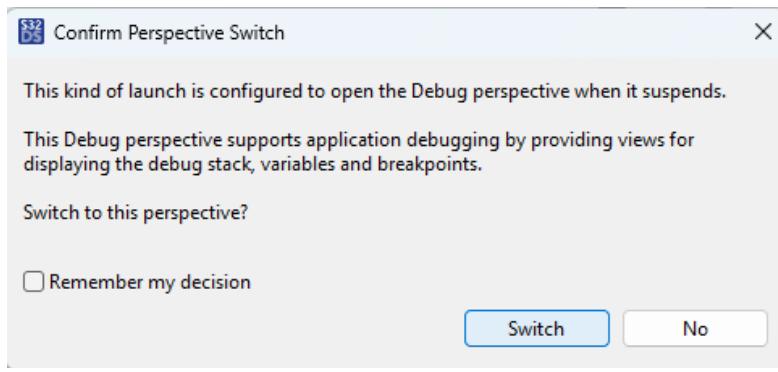
Now the power switch SW1 can be changed to ON-position.



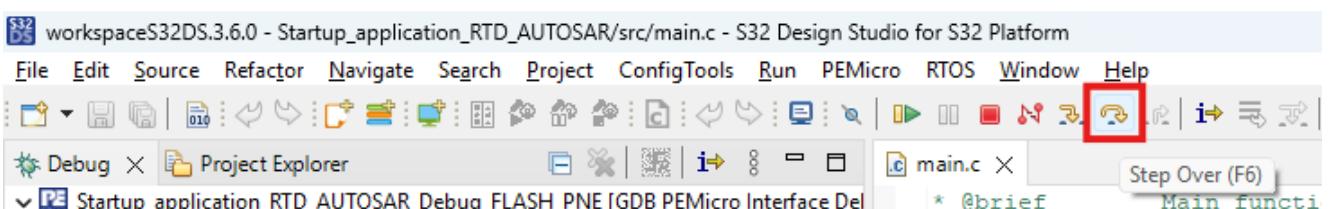
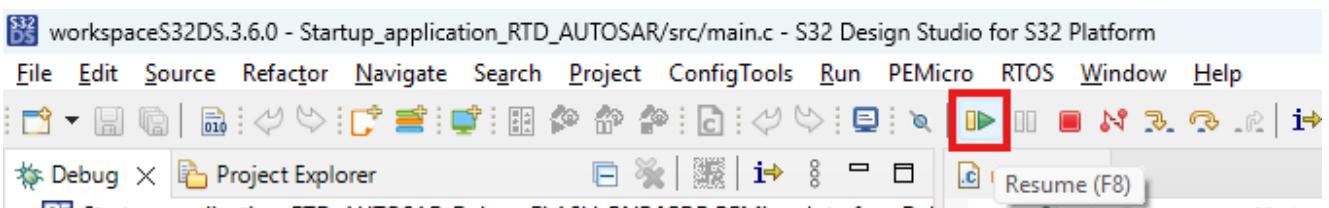
## 13. The LED on Bottom Left Corner should be ON



## 14. Redebug the project



## 15. you can try for example Run ( The RED LED Will blink 10 Times )

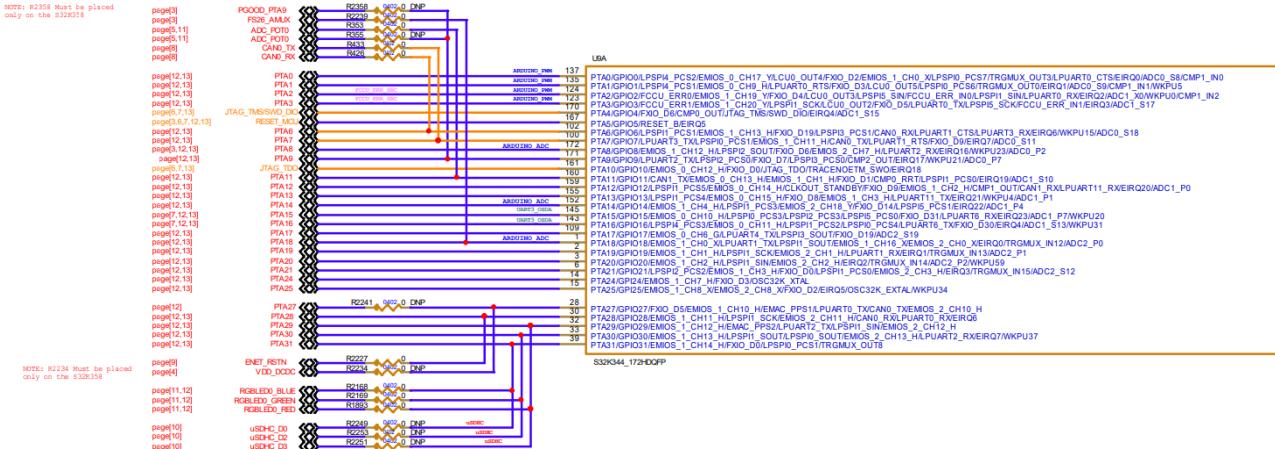


## Let's Toggle The Blue & Green Led Together

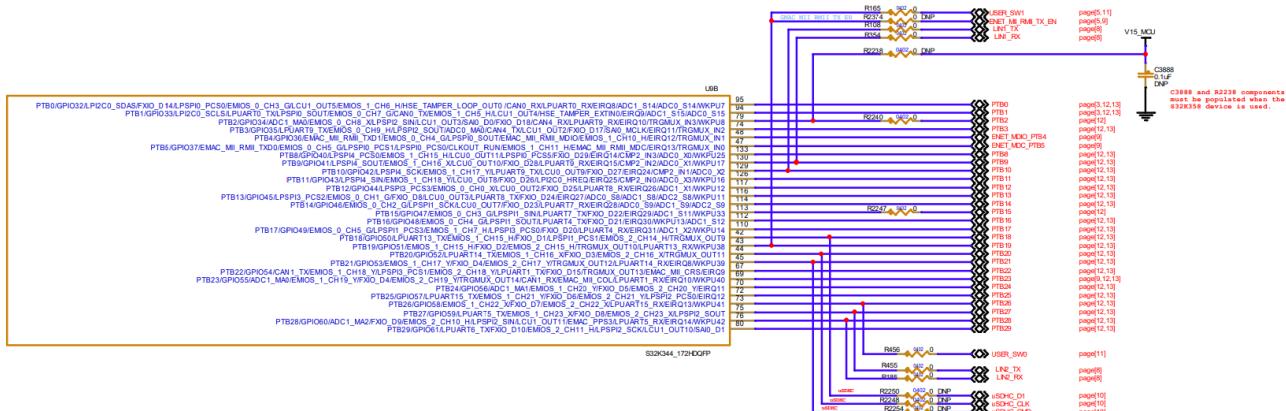
before we started we have some necessary docs :

1. S32K3X4EVB-T172\_Hardware User Manual
  2. S32K3X4EVB-T172\_Schematic

## S32K344 PORTA PINOUT :



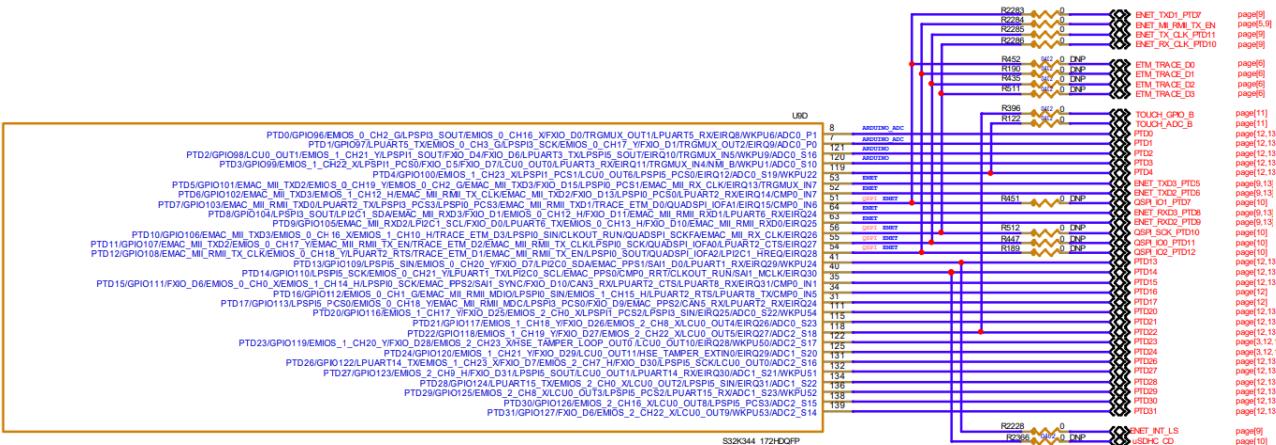
## S32K344 PORTB PINOUT :



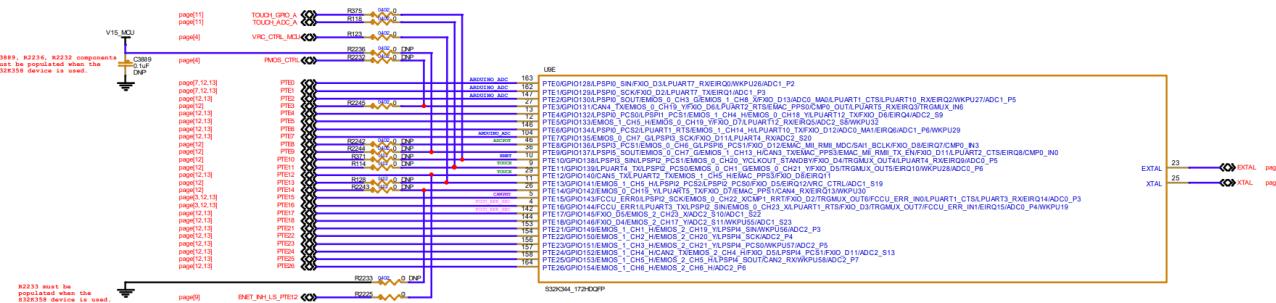
## S32K344 PORTC PINOUT :



## S32K344 PORTD PINOUT :



## S32K344 PORTE PINOUT :



# Default Configuration

Interface	S32K3X4E VB-T172	Reference / Signal	Rev A Def. Config	Rev B Def. Config	Rev B1 Def Config	Description/Comment
S32K344 MCU	●	U9	V1.01	V1.01	V1.01	P32K344EHVPBS0P55A
MCU Power Supply	●	VDD_HV_A _MCU	+5.0V	+5.0V	+5.0V	The VDD_HV_A domain is connected to +5.0V– Switching Power Supply
	●	VDD_HV_B _MCU	+3.3V	+3.3V	+3.3V	The VDD_HV_B domain is connected to +3.3V– Switching Power Supply
	●	VDD_REFH _MCU	[VDD_H V_A]	[VDD_HV_ A]	[VDD_HV_A]	The VDD_REFH domain is connected to VDD_HV_A_MCU
	●	V15_MCU	External NPN Transisto r	External NPN Transistor	External NPN Transistor	The V15_MCU domain is routed to the VCORE from the FS26
Ethernet	●	Ethernet MII/RMII 100 Ethernet PHY	MII Enabled	RMII Enabled	RMII Enabled	The Ethernet PHY with TJA1103A which provides 100 Mbit/s transmit and receive capability
QSPI-A Memory	●	U16	Enabled	Disabled	Disabled	The MCU signals to the QSPI-A Memory Interface are disabled
OnBoard Debugg	●		PTA15	PTA15	PTA15	PTA15/LPUART6_RX is routed to OpenSDA for serial interface
			PTA16	PTA16	PTA16	PTA16/LPUART6_TX is routed to OpenSDA for serial interface
TRACE	●	J12	Enabled	Disabled	Disabled	The TRACE Signals are disabled as DEFAULT in the 20pin cortex Debug D ETM Connector
CAN Interface	●	CAN0	TJA1153 / PTA6	TJA1153/ PTA6	TJA1443/ PTA6	PTA26 is routed to the CAN0_RX signal. Depending the board revision the CAN PHY can vary
			TJA1153 / PTA7	TJA1153/ PTA7	TJA1443/ PTA7	PTA27 is routed to the CAN0_TX signal. Depending the board revision the CAN PHY can vary
			TJA1153 / PTC23	TJA1153/ PTC23	TJA1443/ PTC23	PTC23 is routed to the CAN0_ERRN. Depending the board revision the CAN PHY can vary
			TJA1153 / PTC21	TJA1153/ PTC21	TJA1443/ PTC21	PTC21 is routed to transceiver's ENABLE pin. Depending the board revision the CAN PHY can vary
			TJA1153 / PTC20	TJA1153/ PTC20	TJA1443/ PTC20	PTC20 is routed to transceiver's STB pin. Depending the board revision the CAN PHY can vary
LIN Interface	●	LIN1	TJA1022 / PTB9	TJA1022/ PTB9	TJA1022/ PTB9	LPUART9_RX is routed to LIN Phy0
			TJA1022 / PTB10	TJA1022/ PTB10	TJA1022/ PTB10	LPUART9_TX is routed to LIN Phy0
		LIN2	TJA1022 / PTB28	TJA1022/ PTB28	TJA1022/ PTB28	LPUART5_RX is routed to LIN Phy1
			TJA1022 / PTB27	TJA1022/ PTB27	TJA1022/ PTB27	LPUART5_TX is routed to LIN Phy1
User Push Buttons	●	SW0	PTB26	PTB26	PTB26	Active Low,
User LEDs	●	D13	SW1	PTB19	PTB19	Active Low,
			PTA29	PTA29	PTA29	RGB LED Red
			PTA30	PTA30	PTA30	RGB LED Green
			PTA31	PTA31	PTA31	RGB LED Blue
ADC Potentiometers	●	ADCPOT0	PTA11	PTA11	PTA11	ADCPOT0 [R293] is routed to PTA11 - ADC1_S10

Let's Start :

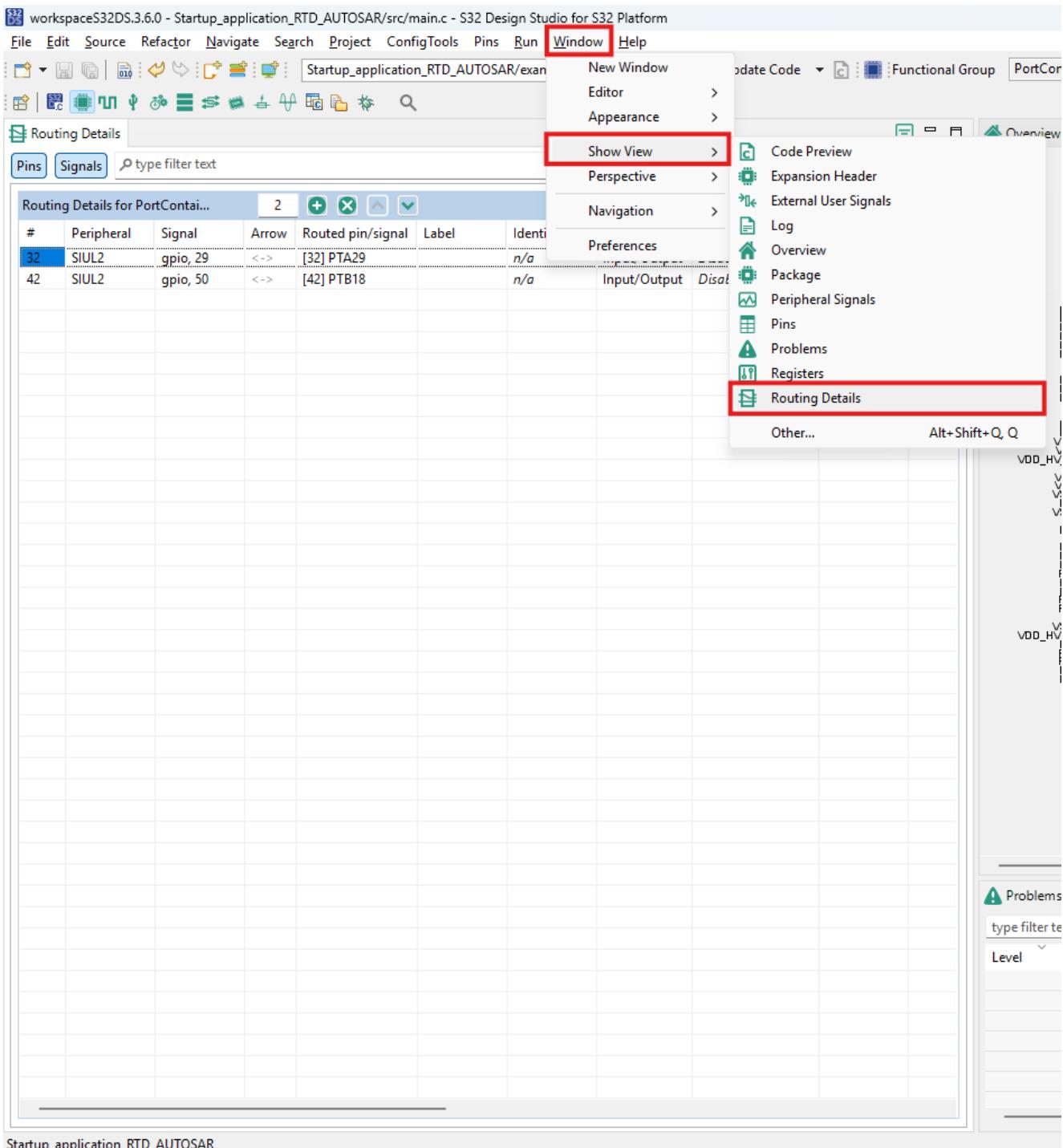
1. we are Going to use PTA30 and PTA31

User LEDs	●	D13	PTA29	PTA29	PTA29	RGB LED Red
			PTA30	PTA30	PTA30	RGB LED Green
			PTA31	PTA31	PTA31	RGB LED Blue

## 2. Open **example\_Port.mex** and then **Routing Detail**

Routing Details for PortContai...											
#	Peripheral	Signal	Arrow	Routed pin/signal	Label	Identifier	Direction	Safe Mode Control	Pull Select	Pullup	
32	SIUL2	gpio, 29	<->	[32] PTA29		n/a	Input/Output	Disable	Pulldown	Disable	
42	SIUL2	gpio, 50	<->	[42] PTB18		n/a	Input/Output	Disable	Pulldown	Disable	

If routing Details is not in your workspace follow this steps



### 3. Add New Row

Routing Details for PortContai...										
#	Peripheral	Signal	Arrow	Routed pin/signal	Label	Identifier	Direction	Safe Mode Control	Pull Select	Pullup
32	SIUL2	gpio, 29	<->	[32] PTA29		n/a	Input/Output	Disable	Pulldown	Disable
42	SIUL2	gpio, 50	<->	[42] PTB18		n/a	Input/Output	Disable	Pulldown	Disable

### 4. For Routed pin : PTA30

Routing Details for PortContai...										
#	Peripheral	Signal	Arrow	Routed pin/signal	Label	Identifier	Direction	Safe Mode Control	Pull Select	Pullup
32	SIUL2	gpio, 29	<->	[32] PTA29		n/a	Input/Output	Disable	Pulldown	Disable
42	SIUL2	gpio, 50	<->	[42] PTB18		n/a	Input/Output	Disable	Pulldown	Disable
33				[33] PTA30		n/a	n/a	Disable	Pulldown	Disable

### 5. For Signal : GPIO 30

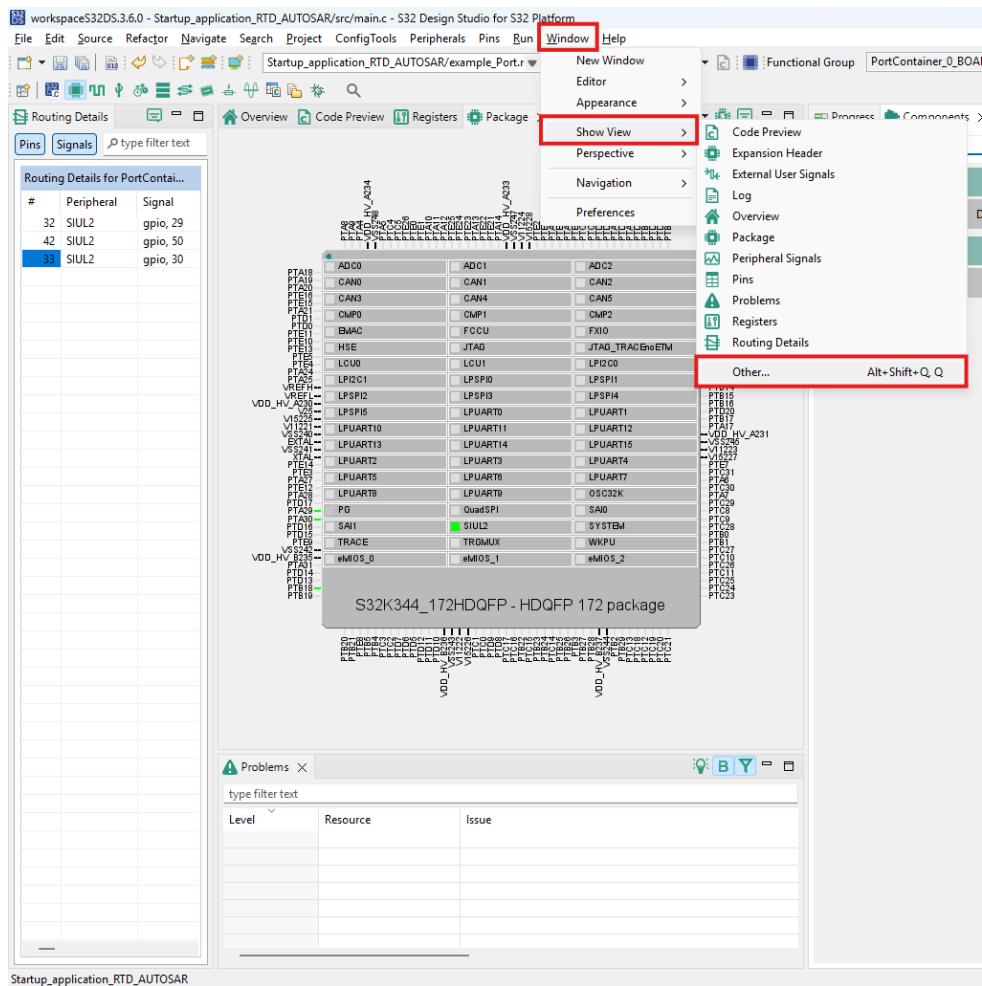
You can Find it by holding the cursor on the top of PTA30

Routing Details for PortContai...										
#	Peripheral	Signal	Arrow	Routed pin/signal	Label	Identifier	Direction	Safe Mode Control	Pull Select	Pullup
32	SIUL2	gpio, 29	<->	[32] PTA29		n/a	Input/Output	Disable	Pulldown	Disable
42	SIUL2	gpio, 50	<->	[42] PTB18		n/a	Input/Output	Disable	Pulldown	Disable
33				[33] PTA30	Pin No.: 33, PTA30 General Purpose I/O 30;eMIOS Channel 13;LPSPI1 Serial Data Output;LPSPI0 Serial Data Output;eMIOS Channel 13;Interrupt Source 37;External IRQ7;LPUART2 Receive data					

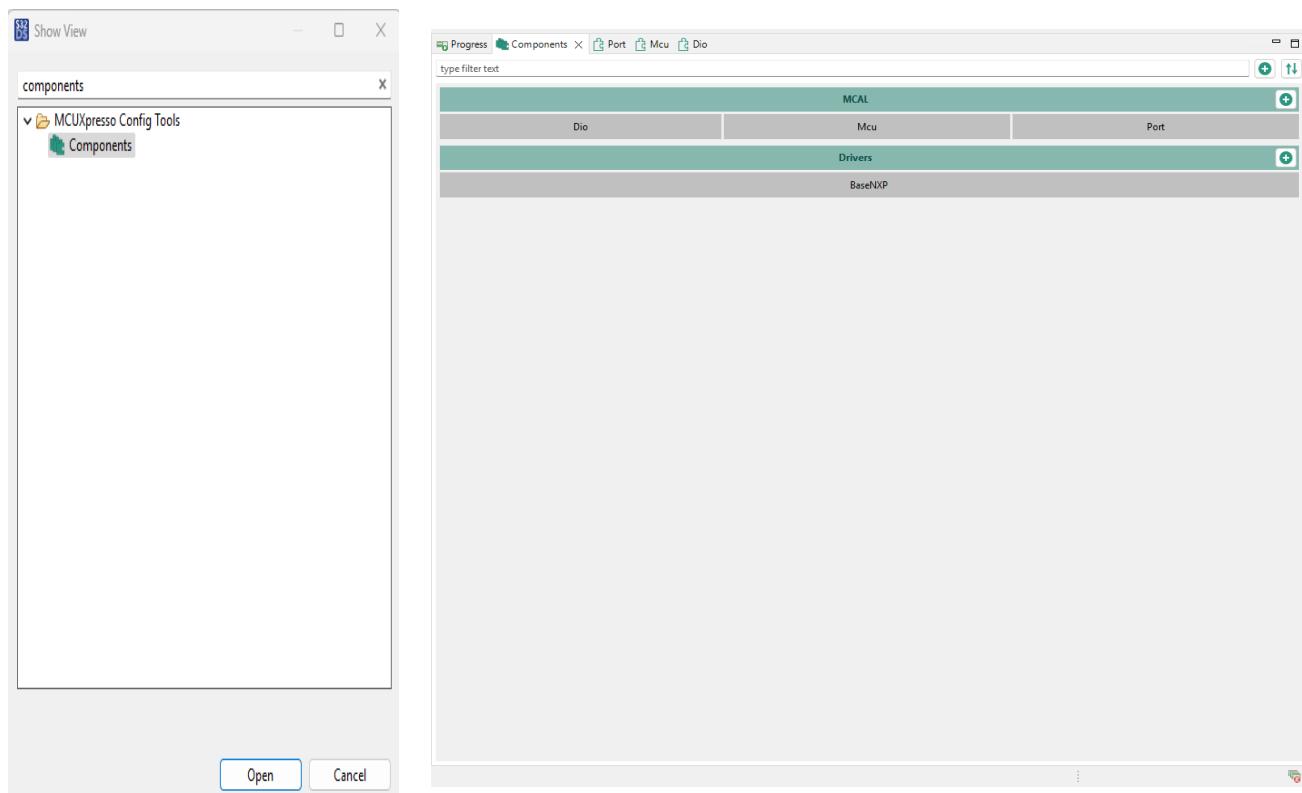
### 6. For Peripheral : SIUL2 and Direction : INPUT/OUTPUT

Routing Details for PortContai...										
#	Peripheral	Signal	Arrow	Routed pin/signal	Label	Identifier	Direction	Safe Mode Control	Pull Select	Pullup
32	SIUL2	gpio, 29	<->	[32] PTA29		n/a	Input/Output	Disable	Pulldown	Disable
42	SIUL2	gpio, 50	<->	[42] PTB18		n/a	Input/Output	Disable	Pulldown	Disable
33	SIUL2	gpio, 30	<->	[33] PTA30		n/a	Input/Output	Disable	Pulldown	Disable

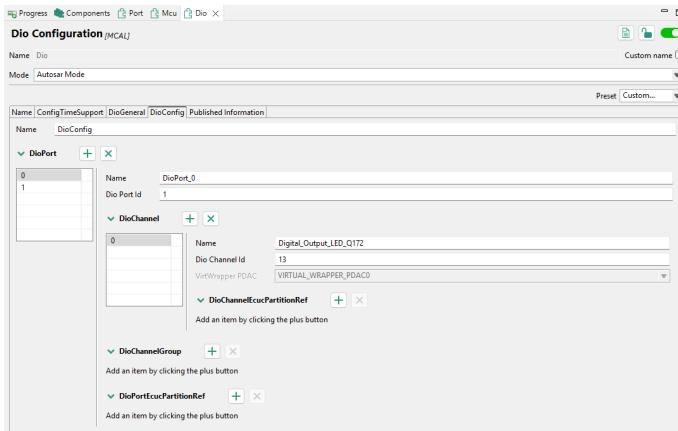
## 7. Open Components Panel



## Search for components



## 8. Open DIO > DioConfig



## 9. We need To see Which Port We are going to use

**Dio Port Id**  
Numeric identifier of the DIO port. Symbolic names will be generated for each port pin idfor the pins which being used for configuration.  
**NOTE:** Use the following values to configure different ports.

Port AL=0 - corresponds to PTA[15-0]

Port AH=1 - corresponds to PTA[31-16]

Port BL=2 - corresponds to PTB[15-0]

Port BH=3 - corresponds to PTB[31-16]

Port CL=4 - corresponds to PTC[15-0]

Port CH=5 - corresponds to PTC[31-16]

Port DL=6 - corresponds to PTD[15-0]

Port DH=7 - corresponds to PTD[31-16]

Port EL=8 - corresponds to PTE[15-0]

Port EH=9 - corresponds to PTE[31-16]

Port FL=10 - corresponds to PTF[15-0]

Port FH=11 - corresponds to PTF[31-16]

Port GL=12 - corresponds to PTG[15-0]

Port GH=13 - corresponds to PTG[31-16]

Use the following values to configure different ports on S32K396 or S32K394 or S32K366 or S32K364 derivative:

Port HL=14 - corresponds to PTH[15-0]

Supported input formats are: hexadecimal, decimal, octal, binary.

**VALUE:**  
1

**Setting value will be synchronized between different instances of the same component.**

You Can't use the port id Twice If you are going for example to Work PTA15 and PTA14

you need to define one Dio port with ID = 1 and define 2 DioChannel

For example : Port ID = 0

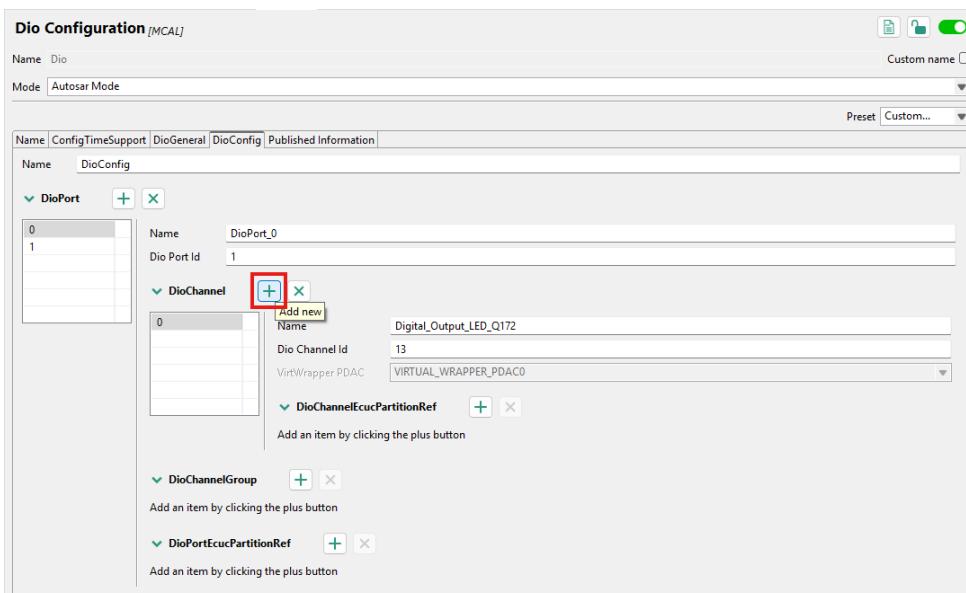
Contains : PTA1,PTA2,PTA3,PTA4,.....,PTA15

In Our Case : PTA30 in PORTID=1

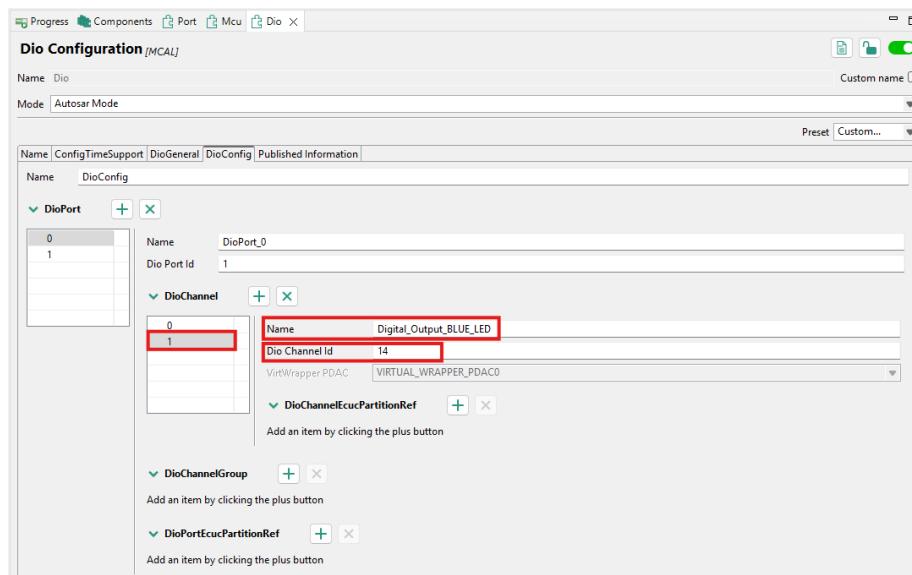
Then we should verify if any of existing Port Have the ID=1 and we define the DIOchannel underit

Dio\_Port\_0 ID = 0

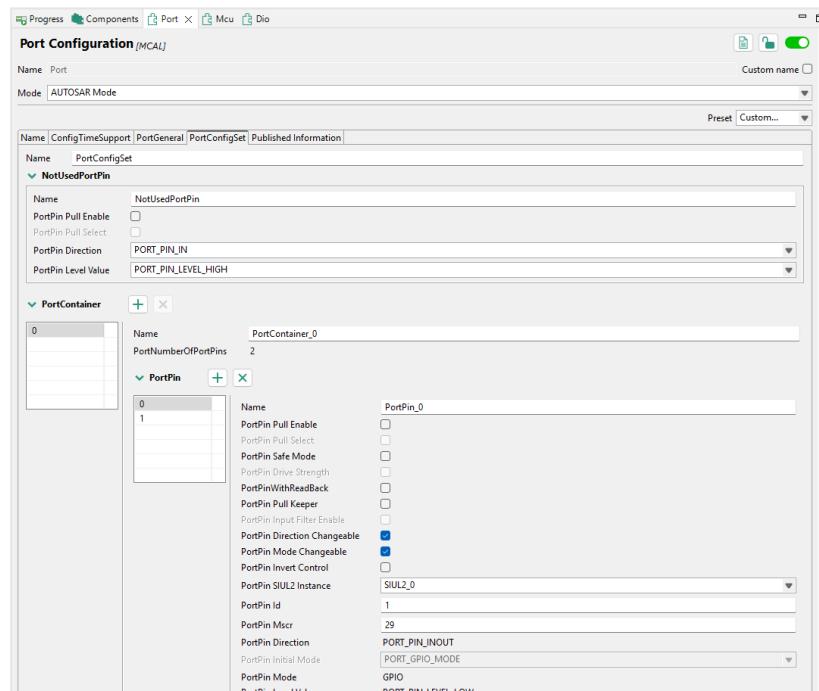
#### 10. We Add New DioChannel



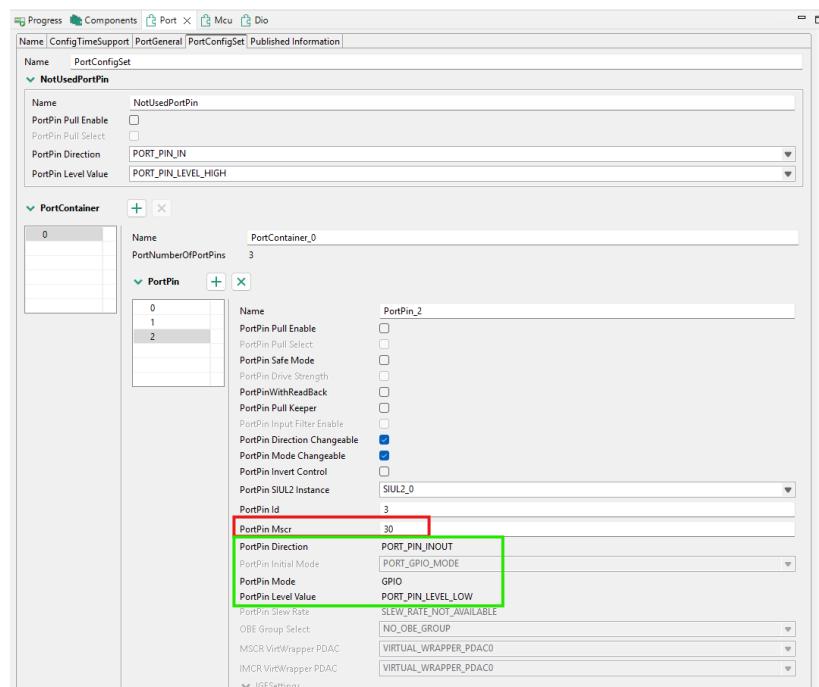
11. For The name : Digital\_Output\_BLUE\_LED and the id = 30 - 16 = 14 ( because the pin PTA16 will have Diochannel id = 0 , PTA17 will have id = 1 , and viseversa )



## 12. Know We are going to configure the Port , Open **PortConfigSet** in Port Panel

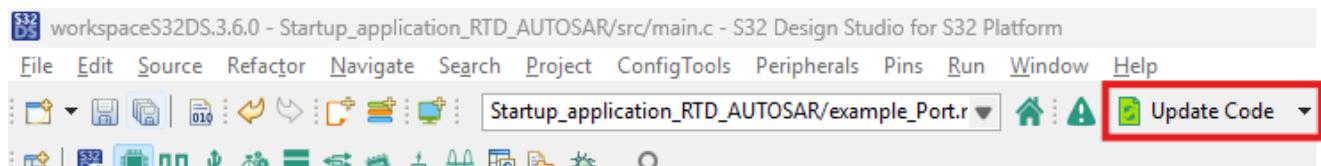


## 13. Add New Port Pin and Put The **PortPinMscr = 30**(GPIO Pin Number)



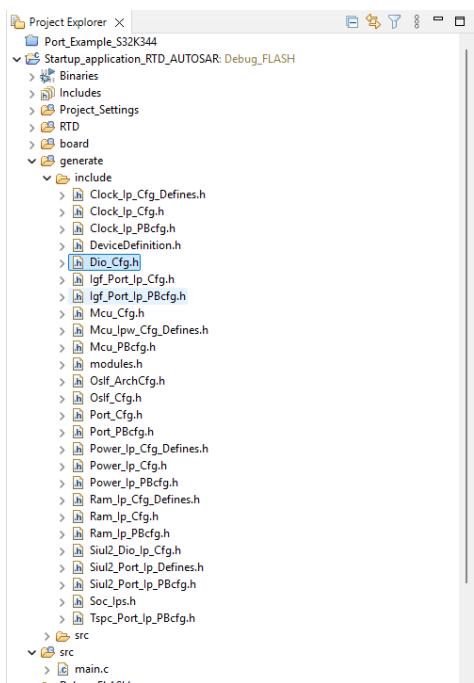
The Green Box should Automaticly updateit

## 14. Now We are Ready to **Update OUR Code**



## 15. Click On **Resource**

## 16. Open Dio\_Cfg.h



You will have a New Block Like this :

```
#define Dio_ConfigPC      (Dio_Config)

/* ===== DioConfig ===== */

/* ----- DioPort_0 ----- */

/**
 * @brief          Symbolic name for the port DioPort_0.
 */
#define DioConf_DioPort_DioPort_0  ((uint8)0x01U)
/** 
 * @brief          Symbolic name for the channel Digital_Output_LED_Q172.
 */
#define DioConf_DioChannel_Digital_Output_LED_Q172  ((uint16)0x001dU)
/** 
 * @brief          Symbolic name for the channel Digital_Output_BLUE_LED.
 */
#define DioConf_DioChannel_Digital_Output_BLUE_LED  ((uint16)0x001eU)
/* ----- DioPort_1 ----- */
/** 
 * @brief          Symbolic name for the port DioPort_1.
 */
#define DioConf_DioPort_DioPort_1  ((uint8)0x03U)
/** 
 * @brief          Symbolic name for the channel Digital_Output_LED_Q257.
 */
#define DioConf_DioChannel_Digital_Output_LED_Q257  ((uint16)0x0032U)
```

17. Our Blue LED is Ready To Be Used

18. in The main.c We add this in the **main()**

```
int main(void)
{
    uint8 count = 0U;

    /* Initialize the Mcu driver */
#if (MCU_PRECOMPILE_SUPPORT == STD_ON)
    Mcu_Init(NULL_PTR);
#elif (MCU_PRECOMPILE_SUPPORT == STD_OFF)
    Mcu_Init(&Mcu_Config);
#endif /* (MCU_PRECOMPILE_SUPPORT == STD_ON) */

    /* Initialize the clock tree and apply PLL as system clock */
    Mcu_InitClock(McuClockSettingConfig_0);

    /* Apply a mode configuration */
    Mcu_SetMode(McuModeSettingConf_0);

    /* Initialize all pins using the Port driver */
    Port_Init(NULL_PTR);

    while (count++ < 10)
    {
        Dio_WriteChannel(DioConf_DioChannel_Digital_Output_BLUE_LED, STD_HIGH);
        TestDelay(5000000);

        Dio_WriteChannel(DioConf_DioChannel_Digital_Output_BLUE_LED, STD_LOW);
        TestDelay(5000000);
    }

    Exit_Example(TRUE);

    return (0U);
}
```

19. Debugg it

Demo