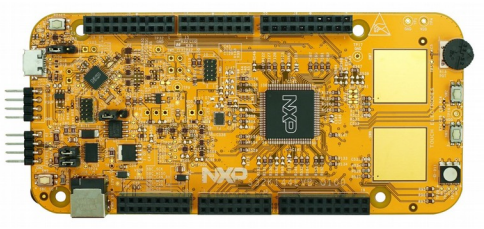
S32K144ZENKIT

How to test this kit?

If you want to test S32K144ZENKIT you need some things. It is:

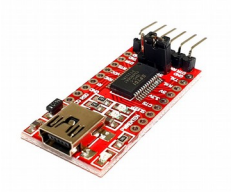
1 S32K144 EVB



2 USB-UART TTL Converter

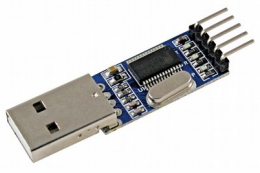
It may be device based FTDI, CH340G or PL2303 or something else. It must work with Linux via tty-driver.

FTDI

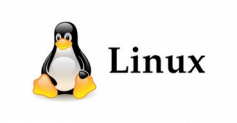


CH340G

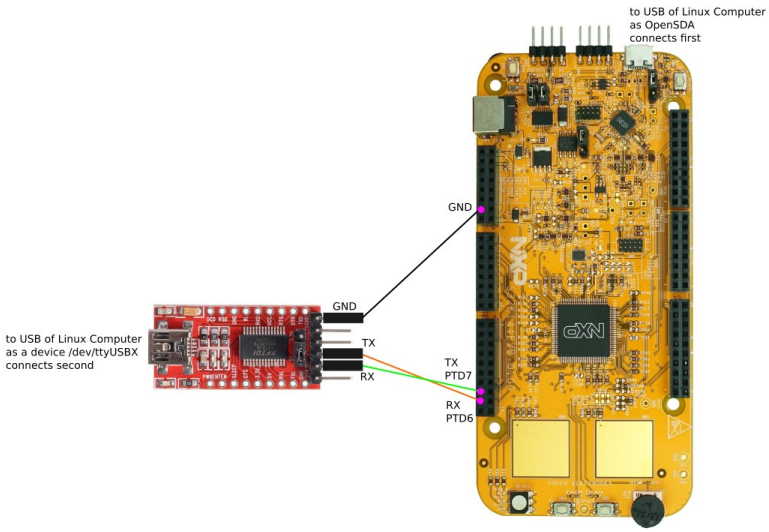
PL2303



3 Linux Computer and S32DS IDE. S32DS IDE may download from nxp.com



If you have all needed, connect those together.



When you install S32DS IDE you must set a directory as workspace, example **~/workspace/s32ds_arm_v2.2**

Create a project using S32DS IDE: File → New → S32DS Application Project

Choose Family S32Kxxx/S32K144

Enter “Project name”: bootloader

Next & Finish

The result will be project in the directory: **~/workspace/s32ds_arm_v2.2/bootloader**

Next step need starting terminal and entry execute follow commands:

> cd ~/workspace/s32ds_arm_v2.2

> git clone https://github.com/somebyte/S32K144ZENKIT

> cd bootloader/src

> ../../S32K144ZENKIT/s32ds_bootload.sh

Add to main.c follow lines:

```
#include "S32K144.h"
#include "bootload/bootload.h"
/* See ../tty/uart.h to choose needed uart & pins */
#define UARTCFG UART_IFC2|UART_PIN_RX1|UART_PIN_TX1|UART_B921600 /* UART2, RX: PTD6, TX: PTD7 */
// Other example: #define UARTCFG UART_IFC0|UART_PIN_RX1|UART_PIN_TX1|UART_B115200 /* UART0, RX: PTB0, TX: PTB1 */
int
main (void)
{
  return bootloadmain(UARTCFG);
}
```

You can run mcu application (**bootloader**) via S32DS IDE as Release_FLASH, now.

Come back to the terminal and try work with mcu:

> sudo dmesg # define what tty device is used by mcu, example it is */dev/ttyUSB1*

> cd ../../S32K144ZENKIT/linux

> make

> sudo ./uploader /dev/ttyUSB1 921600 ../evb/hello_clocks_blue__0x0000B000.srec

> sudo ./uploader /dev/ttyUSB1 921600 ../evb/hello_clocks_red___0x0000F000.srec

> sudo ./uploader /dev/ttyUSB1 921600 ../evb/hello_clocks_green_0x00013000.srec

> sudo ./ttydebug /dev/ttyUSB1 921600

jump 0x000B000 # blink blue

push reset button of mcu

jump 0x000F000 # blink red

push reset button of mcu

jump 0x0013000 # blink green

quit