1 Function Description

KMS relies on a serial communication protocol including an agent embedded on chip to transfer data from the chip to the PC-based user interface. This embedded agent comprises the Communication block.

The UART is used to communicate between KMS GUI and Motor Observer during motor commissioning. The embedded agent in the MCU project is called the RDA Agent. Once commissioning is completed the RDA client is not required and can be removed.

The communication agent does not need to be the UART. It could be the I2C, the SPI or a CAN interface. Changing the communication agent module from UART to something else will the topic of later posts.

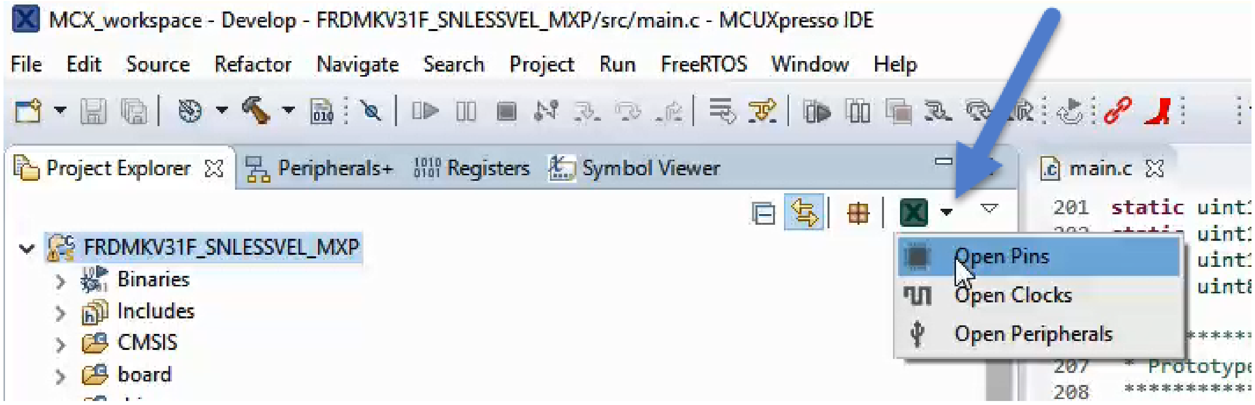
2 Using MCUXpresso IDE and Pins Config Tools to change UART

The default communication port UART0, is used to support all members of the KV3x, KV1x, and KV4x families. Typically, the default pins are PTB16 for RX and PTB17 for TX.

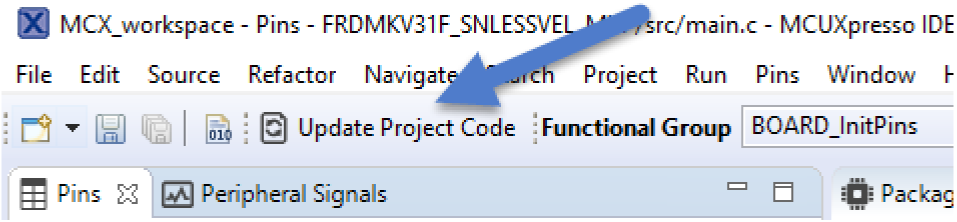
If you are staying with UART0 but are switching pin assignments then use the MCUXpresso pins tool to change from PTB16 and PTB17 to the alternative UART0 pins.

Open the MCUXpresso IDE and with the project selected click the drop-down button next the MCUXpresso tools icon.

2.1 Select Open Pins.



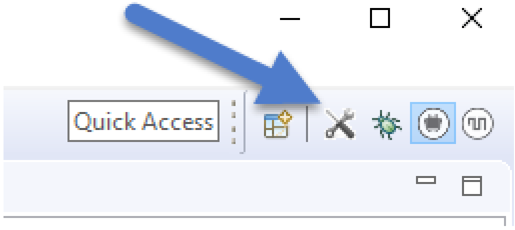
Make the changes needed to the pin assignments. When you have completed the changes to the pin assignments make sure to update the project code by clicking on the ***Update Project Code*** button.


MCX_workspace - Pins - FRDMKV31F_SNLESSVEL_M /src/main.c - MCUXpresso IDE
File Edit Source Refactor Navigate Search Project Run Pins Window H
Update Project Code
Functional Group
BOARD_InitPins
Pins
Peripheral Signals
Packag

This has the effect of updating the file "pin_mux.c" and "pin_mux.h" changing the UART pins definitions to correspond to the pins used on your custom board (for example: PTA14 and PTA15 for RX and TX or PTE0, PTE1 for UART1).

2.2 Return to IDE develop perspective

Move back to the develop perspective by clicking on the Develop icon in the top right hand corner



2.3 Using UART1 ?

If you are moving to UART1 then additional steps are needed.

2.3.1 Modify in the kms_hw.h file changing UART0 to UART1 and reduce the baud rate for serial communications (KMS_UART_BAUDRATE)

1. inside "main.c" find the calls and function declarations/definitions, explicitly referring to UART0:
   - FAST void UART0_RX_TX_IRQHandler(void) which overwrites the WEAK void UART0_RX_TX_IRQHandler(void)
   - void UART0_ERR_IRQHandler(void)
   - NVIC_EnableIRQ(UART0_RX_TX_IRQn);
   - NVIC_SetPriority(UART0_RX_TX_IRQn, 15);
   - NVIC_EnableIRQ(UART0_ERR_IRQn);
   - NVIC_SetPriority(UART0_ERR_IRQn, 15);
2. Changed 'UART0' with 'UART1' in all these instances.

> NOTE: There is a key difference between UART0 and UART1 on the KVx microcontrollers. UART0 has a much deeper FIFO as compared to the other UARTs. (Section 3.9.3.1 of the KV31F128 Ref Manual)

In summary: Change pin mux assignments with the MCUXpresso pin mux tool, then as necessary change the UART instance.

1. in kms_hw.h

```
/**
* @def KMS_UART
* @brief UART Instance used by KMS
*/
#define KMS_UART (UART1)
/**
* @def KMS_UART_BAUDRATE
* @brief Default baud rate for KMS UART communication
*/
#define KMS_UART_BAUDRATE (57600u)
```

2. In main: Change all of the UART0 references to UART1.

```
a. FAST void UART1_RX_TX_IRQHandler(void);
b. void UART1_ERR_IRQHandler(void);
c. void UART1_RX_TX_IRQHandler(void)
   {
   timestamp_t startCycleCount = GetProfilerCycles();

   RDA_UART_ISR();

   CpuUtilization.UartIsrCycles = GetProfilerCycles() -
   startCycleCount;
   }
d. void UART1_ERR_IRQHandler(void)
   {
   RDA_UART_ERR_ISR();
   }
e. NVIC_EnableIRQ(UART1_RX_TX_IRQn);
   NVIC_SetPriority(UART1_RX_TX_IRQn, 15);
   NVIC_EnableIRQ(UART1_ERR_IRQn);
   NVIC_SetPriority(UART1_ERR_IRQn, 15);
```