

UART User Manual

1. Configure the GPIO pins for the UART you intend to use.

- Open `GPIO_Cfg.h` and alter the definition of `GPIO_GROUPS_NUMBER`.
- Open `GPIO_Cfg.c` and alter the configuration for each pin. You only need to change the `PortId`, `PortMask`, and `PortDirection` variables. The other variable can be left unchanged.

Example: Configuring the GPIO pins for UART2.

- Open `GPIO_Cfg.h` and alter the definition of `GPIO_GROUPS_NUMBER` to 2.
- Based on table 14-1 in the TM4c123GH6PM datasheet at page 892, PD6 and PD7 are the pins assigned for UART2.
So, the `PortID`, `PortMask`, and `PortDirection` for the GPIO pins will be assigned the values 0x03, 1<<6, 0x00 and 0x03, 1<<7, 0xff respectively.
- That makes the `groupID = 0` for the Rx pin 1 and makes the `groupID = 1` for the Tx pin. Pay attention to this as we will be using it in the following section.

2. Configure the UART you intend to use.

- Open UART_Cfg.h and alter the definition of UART_GROUPS_NUMBER.
- Open UART_Cfg.c and alter the configuration. The main variable that you may need to change are the UARTPortId, Rx_Pin_GPIO_groupID, Tx_Pin_GPIO_groupID, BaudRate, FIFOEN, TxDoneCallbackPtr, RxDoneCallbackPtr, Int_Enable, and FIFO_INT_Level variables. The other variable can be left unchanged.

Following our example:

- Open UART_Cfg.h and alter the definition of UART_GROUPS_NUMBER to 1.
- Open UART_Cfg.c and alter the configuration. So the previously mentioned variables will be assigned the following values: 0x02, 0, 1, 9600, Enabled/Disabled, &TxDone_Func, &RxDone_Func, Enabled/Disabled, and FIFO_8 respectively.
- That makes the groupID = 0 for UART2. Pay attention to this as we will be using it in the following sections.

3. Initializing the UART Driver.

The UART channel needs to be initialized to start working.

You shall do this by calling two functions; `GPIO_Init()`, and `UART_Init()`.

4. The UART Driver has 2 configurations; UART using interrupts, and UART using cyclic tasks. We will look into both of them in the next two sections.

5. Using the interrupts configuration.

- First, we need to make sure that the `Int_Enable` variable in the UART configuration is configured as `Enabled`.
- To trigger the interrupt, you need to call a special function. This function is: `EN_INT()`.
This function takes an input argument based on table 2-9 in the TM4c123GH6PM datasheet at page 102.
- You need to call the transmitter initialization function: `UART_Tx_Init`, or the receiver initialization function: `UART_Rx_Init`.
- Those 2 functions nearly take the same argument. If we take the transmitter initialization function as an example, it takes 3 arguments: the array of data to be sent, the length of the array, and the `groupID` of the UART.

Following our example: Sending 25 bytes of data using UART2.

- First of all, you need to call `EN_INT(33)`.
- Then, you call `UART_Tx_Init(array_name, 25, 0)`.

6. Using the cyclic configuration with enabled FIFO.

- First, you need to make sure that the `Int_Enable` variable in the UART configuration is configured as `Disabled`.
- Also, you need to make sure that the `FIFOEN` variable in the UART configuration is configured as `Enabled`.
- You need to call the transmitter initialization function: `UART_Tx_Init`.
- The transmitter initialization function takes 3 arguments: the array of data to be sent, the length of the array, and the `groupID` of the UART.
- At last, you need to call the transmitter managing function.

Following our example: Sending 25 bytes of data using UART2.

- First of all, you need to call `UART_Tx_Init(array_name, 25, 0)`. This function sends the first 16 bytes of the data.
- Then, you call `UART_Tx(0)`. This function sends the rest of the data, which are 9 bytes in this example.
- Then, you call `UART_Tx(0)`. This call will result in calling the callback function.

Another example: Sending 5 bytes of data using UART2.

- First of all, you need to call `UART_Tx_Init(array_name, 5, 0)`. This function sends all data.
- Then, you call `UART_Tx(0)`. This call will result in calling the callback function.

7. Using the cyclic configuration with disabled FIFO.

- First, you need to make sure that the `Int_Enable` variable in the UART configuration is configured as `Disabled`.
- Also, you need to make sure that the `FIFOEN` variable in the UART configuration is configured as `Disabled`.
- You need to call the transmitter initialization function: `UART_Tx_Init`.
- The transmitter initialization function takes 3 arguments: the array of data to be sent, the length of the array, and the `groupID` of the UART.
- At last, you need to call the transmitter managing function.

Following our example: Sending 25 bytes of data using UART2.

- First of all, you need to call `UART_Tx_Init(array_name, 25, 0)`. This function only sends the **first byte** of the data.
- Then, you call `UART_Tx(0)` 24 times. This function sends the rest of the data, which are 24 bytes in this example.
- The last called function will transmit the 24th byte and then will call the callback function. So, you don't need to call `UART_Tx(0)` again.

Another example: Sending 5 bytes of data using UART2.

- First of all, you need to call `UART_Tx_Init(array_name, 5, 0)`. This function only sends the **first byte** of the data.
- Then, you call `UART_Tx(0)` 4 times. This function sends the rest of the data, which are 4 bytes in this example.
- The last called function will transmit the 4th byte and then will call the callback function. So, you don't need to call `UART_Tx(0)` again.