```c
/* ***************************************************************** */
/* File name:        UART.c                                          */
/* File description: Debugging through UART interface                */
/* Author name:      dloubach, rbacurau                              */
/* Creation date:    22out2015                                       */
/* Revision date:    01mai2020                                       */
/* ***************************************************************** */

/* definition include */
#include "UART.h"

/* system includes */
#include "fsl_clock_manager.h"
#include "fsl_device_registers.h"
#include "fsl_port_hal.h"
#include "fsl_smc_hal.h"
#include "fsl_debug_console.h"
#include "communicationStateMachine.h"

/* UART definitions */
#ifndef BOARD_DEBUG_UART_INSTANCE
   #define BOARD_DEBUG_UART_INSTANCE   0
   #define BOARD_DEBUG_UART_BASEADDR   UART0
#endif
#ifndef BOARD_DEBUG_UART_BAUD
   #define BOARD_DEBUG_UART_BAUD       115200
#endif


/* ************************************************ */
/* Method name:        UART0_init                   */
/* Method description: Initialize the UART0 as debug*/
/* Input params:       n/a                          */
/* Output params:      n/a                          */
/* ************************************************ */
void UART0_init (void)
{
   /* UART0 */
   /* UART0_RX */
   PORT_HAL_SetMuxMode(PORTA, 1u, kPortMuxAlt2);
   /* UART0_TX */
   PORT_HAL_SetMuxMode(PORTA, 2u, kPortMuxAlt2);

   /* Select the clock source for UART0 */
   SIM_SOPT2 |= 0x4000000;

   /* Init the debug console (UART) */
   DbgConsole_Init(BOARD_DEBUG_UART_INSTANCE, BOARD_DEBUG_UART_BAUD, kDebugConsoleLPSCI);
}

/* ************************************************ */
/* Method name:        UART0_enableIRQ              */
/* Method description: Enable the interruption for  */
/*                     serial port inputs and       */
/*                     prepare the buffer           */
/* Input params:       n/a                          */
/* Output params:      n/a                          */
/* ************************************************ */
void UART0_enableIRQ(void)
{
   /* Enable interruption in the NVIC */
   NVIC_EnableIRQ(UART0_IRQn);

   /* Enable receive interrupt (RIE) in the  UART module */
   UART0_C2 |= 0x20;
}
```

```
/* ************************************************ */
/* Method name:        UART0_IRQHandler             */
/* Method description: Serial port interruption     */
/*                     handler method. It Reads the */
/*                     new character and saves in   */
/*                     the buffer                   */
/* Input params:       n/a                          */
/* Output params:      n/a                          */
/* ************************************************ */
void UART0_IRQHandler(void)
{
 // Solicita o processo de tratamento de byte recebido pela UART
 processByteCommUART(debug_getchar());

}
```