XT32H05x

XT32 microcontroller UART

Application notes

Rev 0.0.0

Original Release Date: 12-Sep-2023

Revised :

Revision History

Release	Date	Author	Summary of Change
V0.0.0	12/09/2023	Shirling Liu	Initial

Contents

List of Figures

List of Tables

1 Introduce

This application note serves as a comprehensive guide for software developers, offering essential information on Universal Asynchronous Receiver/Transmitter (UART). It covers fundamental concepts and provides guidelines to ensure proper utilization of UARTs in software development projects. Whether you're a beginner or an experienced developer, this document will equip you with the necessary knowledge and best practices to effectively configure and utilize UARTs in your applications.

1.1 Required peripherals

This application involves modules as table 1.

Table 1. Modules in example

Sub-module	Peripheral use	Note
PADI	2 ports as UART transmitter port and receiver port	Call HAL_PADI_Init() in code
UART	pin2 as UART1TX, pin6 as UART1RX	
GPIO	2 LED pins and 1 button pin	**only for result indication purposes.

DMA	DMA Handshaking Interface: INDEX 0 and INDEX 1 of CFG_0	**Only for DMA mode

1.2 Compatible devices

This example is compatible with the devices in Table 2.

Table 2. Device list

Product	EVB
XT32H050	XB002823

2 Design description

2.1 Feature overview

XT32H0xxx provides 4 UART peripherals: UART1, UART2, UART3, UART4.

The UART software provides the following features:

- Half-duplex operation
- Asynchronous operation
- Flexible data formats (5~ 8 data bits, 1 or 2 stop bits)
- Baud rate: 2400 to 1500000 baud

2.2 Design steps

1. Enable UART1 Baudrate source clock and set clock divider.

2. Configure pin alternate function as UART from Peripheral PADI through PADI_InitTypeDef structure. This example uses UART1 as serial communication.

   - PADI_IDX_IO1_UART1_TX, means select and enable the IO1(pin 2).
   - PADI_CFG_IO1_UART1_TX, means select UART0TX function for IO1.

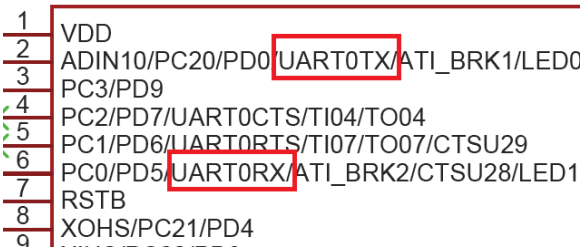

Figure 1. IO function selection

Note: please refer to XT32H0xxB—reference manual document to find the assignment relationship between pin 2 with IO1, pin6 with IO5.

3. Configure parameters for UART module.
4. Assign system DMA handshaking interface to UART1TX& UART1RX, link DMA handle with UART handle, and enable DMA1_IRQn interrupt configuration if the example under DMA mode.
5. Process to transfer serial data with external devices.

2.3 Design considerations

2.4 Software flowchart

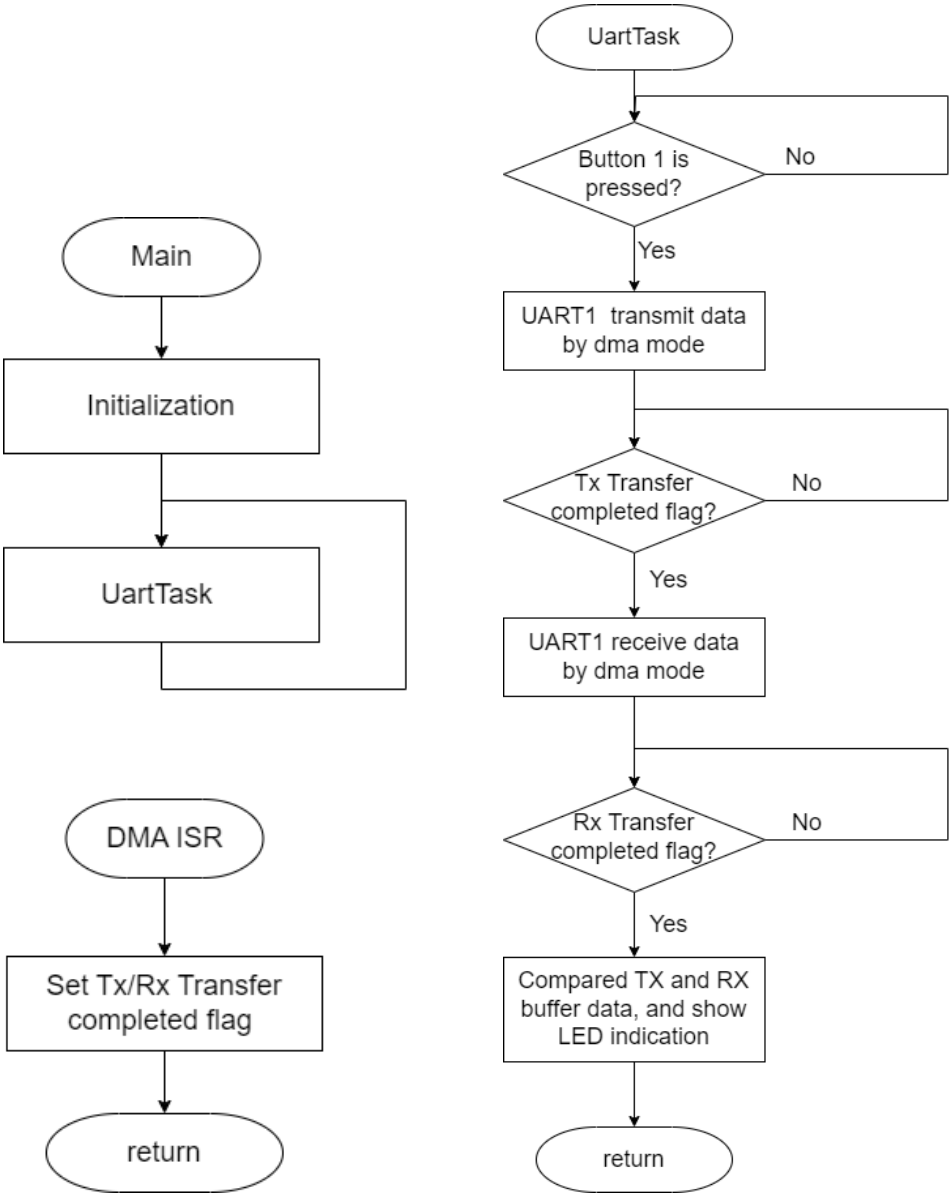

Figure 2. Application flow—DMA mode

2.5 Reference code

Configure Peripheral PADI through PADI_InitTypeDef structure to select alternate function as UART1 interface as bellowing code:

```
  if(huart->Instance == UART1)
  {
    sPadConfig.Pad       = PADI_IDX_IO5_UART1_RX ;
    sPadConfig.Alternate = PADI_CFG_IO5_UART1_RX ;
    HAL_PADI_Init(PADI, &sPadConfig);  /* Rx IO */
    sPadConfig.Pad       = PADI_IDX_IO1_UART1_TX ;
    sPadConfig.Alternate = PADI_CFG_IO1_UART1_TX ;
    HAL_PADI_Init(PADI, &sPadConfig);  /* Tx IO */
  }
```

If using DMA mode to transfer serial data, should assign system DMA handshaking interface to UART1 as below code:

```
    /* Enable uart1 dma hsifconfig index,*/
    HAL_UART_DMAHSIFConfig(huart,  &hdmamemRX, &hdmamemTX,
     LL_DMA_SHIF_CFG_0_INDEX_UART1_RX,
LL_DMA_SHIF_CFG_0_INDEX_UART1_TX,
     LL_DMA_SHIF_CFG_0_CFG_UART1_RX,   LL_DMA_SHIF_CFG_0_CFG_UART1_TX);
    HAL_UART_LinkDMA(huart, &hdmamemRX, &hdmamemTX);
```

Configure Peripheral UART1:

```
   /* -3- Configure uart module */
```

```
  huart1.Instance = UART1;
  huart1.Init.BaudRate = 9600;
  huart1.Init.WordLength = UART_WORDLENGTH_8;
  huart1.Init.StopBits = UART_STOPBITS_1;
  huart1.Init.Parity = UART_PARITY_NONE;
  huart1.Init.HwFlowCtl = UART_HWCONTROL_NONE;

  if (HAL_UART_Init(&huart1) != HAL_OK)
  {
    /* Error_Handler */
    Error_Handle();
  }
```

XT_Uart_Task handles the basic transfer process.

```
void XT_Uart_Task(void)
{
  if(EVB_Button1_State_Get()==PRESSED)
  {
    if (HAL_UART_Transmit_DMA(&huart1, (uint8_t *)aTxBuffer,
sizeof(aTxBuffer)) != HAL_OK)
    {
      if(HAL_UART_GetError(&huart1)!=HAL_UART_ERROR_NONE)
        Error_Handle();
    }
    while(!uTxFinished)
    {
    }
    uTxFinished = FALSE;
    /*receive data from slave device */
    if (HAL_UART_Receive_DMA(&huart1, (uint8_t *)aRxBuffer,
sizeof(aTxBuffer)-2) != HAL_OK)
    {
      if(HAL_UART_GetError(&huart1)!=HAL_UART_ERROR_NONE)
        Error_Handle();
    }
```

```
  while(!uRxFinished)
  {
  }
  uRxFinished = FALSE;
  /*receive data until the bus idle state*/

  if(Buffercmp(aRxBuffer,aTxBuffer,(sizeof(aTxBuffer))-2)==0){
    EVB_Led_Toggle(LED_GREEN);
  }else
    EVB_Led_Off(LED_GREEN);
}
```

2.6 Additional resources

- XT32H0xxB--reference manual
- XT32H0xxB--gpio-AN230200
- XT32H0xxB--dma-AN230800