Experiment No 12

Serial Communication Interface in STM32

Objectives:

- To learn the basics of USART and its usage by STM32F103 Board
- Interfacing USART by using the Polling & Interrupt method.
- Displaying the data received by the STM32 board on LCD.

12.1 USART Introduction:

The universal synchronous asynchronous receiver transmitter (USART) offers a flexible means of full-duplex data exchange with external equipment requiring an industry standard NRZ asynchronous serial data format. The USART offers a very wide range of baud rates using a fractional baud rate generator. It supports synchronous one-way communication and half-duplex single wire communication. It also supports the LIN (local interconnection network), Smartcard Protocol and IrDA (infrared data association) SIR ENDEC specifications, and modem operations (CTS/RTS). It allows multiprocessor communication. High speed data communication is possible by using the DMA for multi-buffer configuration.

12.2 USART main features:

- Full duplex, asynchronous communications
- NRZ standard format (Mark/Space)
- Fractional baud rate generator systems
  – A common programmable transmit and receive baud rates up to 4.5 MBits/s
- Programmable data word length (8 or 9 bits)
- Configurable stop bits - support for 1 or 2 stop bits
- LIN Master Synchronous Break send capability and LIN slave break detection capability
  – 13-bit break generation and 10/11-bit break detection
  when USART is hardwareconfigured for LIN
- Transmitter clock output for synchronous transmission
- IrDA SIR Encoder Decoder
  – Support for 3/16-bit duration for normal mode
- Smartcard Emulation Capability

  - The Smartcard interface supports the asynchronous protocol Smartcards as defined inISO 7816-3 standards
  - 0.5, 1.5 Stop Bits for Smartcard operation
- Single wire half duplex communication
- Configurable multi-buffer communication using DMA (direct memory access)
  - Buffering of received/transmitted bytes in reserved SRAM using centralized DMA
- Separate enable bits for Transmitter and Receiver
- Transfer detection flags:
  - Receive buffer full
  - Transmit buffer empty
  - End of Transmission flags
- Parity control:
  - Transmits parity bit
  - Checks parity of received data byte
- Four error detection flags:
  - Overrun error
  - Noise error
  - Frame error
  - Parity error
- Ten interrupt sources with flags:
  - CTS changes
  - LIN break detection
  - Transmit data register empty
  - Transmission complete
  - Receive data register full
  - Idle line received
  - Overrun error
  - Framing error
  - Noise error
  - Parity error
- Multiprocessor communication

  - enter into mute mode if address match does not occur
- Wake up from mute mode (by idle line detection or address mark detection)
- Two receiver wakeup modes: Address bit (MSB, 9th bit), Idle line

12.3 USART Functional Description:

The interface is externally connected to another device by three pins. Any USART bidirectional communication requires a minimum of two pins: Receive Data In (RX) and Transmit Data Out (TX):

RX: Receive Data Input is the serial data input. Oversampling techniques are used for data recovery by discriminating between valid incoming data and noise.

TX: Transmit Data Output. When the transmitter is disabled, the output pin returns to its IO port configuration. When the transmitter is enabled and nothing is to be transmitted, the TX pin is at high level. In single-wire and smartcard modes, this IO is used to transmit and receive the data(at USART level, data are then received on SW_RX).

Through these pins, serial data is transmitted and received in normal USART mode as frames comprising:

- An Idle Line prior to transmission or reception
- A start bit
- A data word (8 or 9 bits) least significant bit first
- 0.5,1, 1.5, 2 Stop bits indicating that the frame is complete
- This interface uses a fractional baud rate generator
  - with a 12-bit mantissa and 4-bit fraction
- A status register (USART_SR)
- Data register (USART_DR)
- A baud rate register (USART_BRR)
  - 12-bit mantissa and 4-bit fraction.
- A Guard time Register (USART_GTPR) in case of Smartcard mode.

The following pin is required to interface in synchronous mode:

- CK: Transmitter clock output. This pin outputs the transmitter data clock for synchronous transmission corresponding to SPI master mode (no clock pulses on start bit and stop bit, and a software option to send a clock pulse on the last data bit). In parallel data can be received synchronously on RX. This can be used to control peripherals that have shift registers (e.g. LCD drivers). The clock phase and polarity are software programmable. In Smartcard mode, CK can provide the clock to the smartcard. The following pins arerequired in Hardware flow control mode:
- CTS: Clear To Send blocks the data transmission at the end of the current transfer whenhigh
- RTS: Request to send indicates that the USART is ready to receive a data (when low).

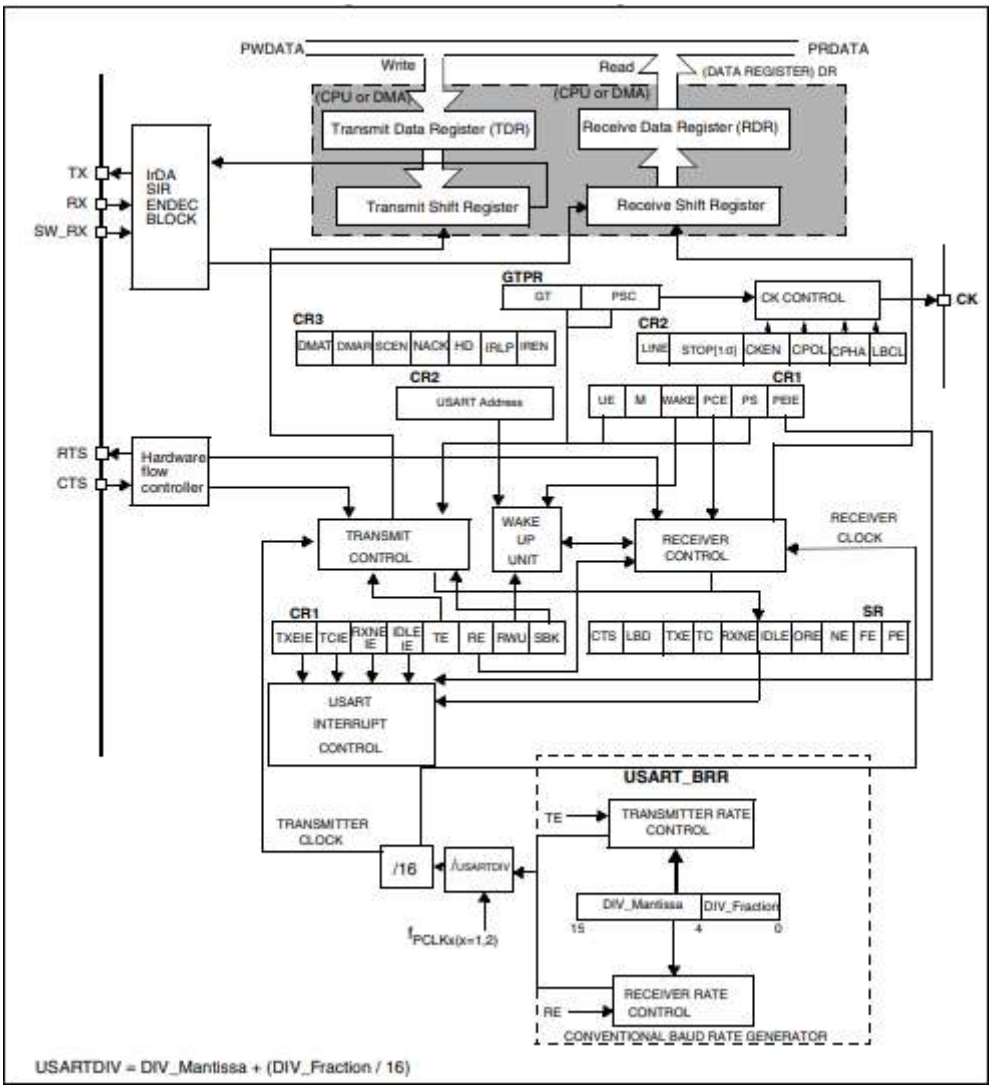


Figure 1: USART Block Diagram

12.4 Configuration for USART:

```
void USART2_Init()
{
        // Initialization struct
        USART_InitTypeDef
        USART_InitStruct;
        GPIO_InitTypeDef
        GPIO_InitStruct;

        // Step 1: USART2 initialization
        RCC_APB1PeriphClockCmd(RCC_APB1Periph_USART2, ENABLE);
        USART_InitStruct.USART_BaudRate = 9600;
        USART_InitStruct.USART_HardwareFlowControl =
        USART_HardwareFlowControl_None;USART_InitStruct.USART_Mode =
        USART_Mode_Rx | USART_Mode_Tx; USART_InitStruct.USART_Parity =
        USART_Parity_No; USART_InitStruct.USART_StopBits =
        USART_StopBits_1; USART_InitStruct.USART_WordLength =
        USART_WordLength_8b; USART_Init(USART2, &USART_InitStruct);
        USART_Cmd(USART2, ENABLE);

        // Step 2: GPIO initialization for Tx and Rx pin
        RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA, ENABLE);
        // Tx pin initialization as push-pull
        alternate function
        GPIO_InitStruct.GPIO_Pin =
        GPIO_Pin_2;
        GPIO_InitStruct.GPIO_Mode =
        GPIO_Mode_AF_PP;
        GPIO_InitStruct.GPIO_Speed =
        GPIO_Speed_2MHz;
        GPIO_Init(GPIOA, &GPIO_InitStruct);
        // Rx pin initialization as input floating
        GPIO_InitStruct.GPIO_Pin = GPIO_Pin_3;
        GPIO_InitStruct.GPIO_Mode =
        GPIO_Mode_IN_FLOATING;
        GPIO_InitStruct.GPIO_Speed =
        GPIO_Speed_2MHz; GPIO_Init(GPIOA,
        &GPIO_InitStruct);}
```

12.5 Libraries to include:

USART_Polling.c | lcd16x2.h

```
#include "stm32f10x.h"
#include "stm32f10x_rcc.h"
#include "stm32f10x_gpio.h"
#include "stm32f10x_usart.h"
#include "delay.h"
#include "lcd16x2.h"
#include <string.h>

void USART2_Init(void);
void USART2_PutChar(char c);
void USART2_PutString(char *s);
uint16_t USART2_GetChar(void);

// Buffer for store received chars
#define BUF_SIZE  16
char buf[BUF_SIZE];
int i = 0;

```

12.6 Main Function:

```
int main(void)
{
  DelayInit();
  lcd16x2_init(LCD16X2_DISPLAY_ON_CURSOR_OFF_BLINK_OFF);
  lcd16x2_puts("Soban");

  // Initialize USART
  USART2_Init();

  USART2_PutChar('W');
  USART2_PutString(" Test");

  while (1)
  {
    // Read received char

    char c = USART2_GetChar();
```

```
36      lcd16x2_gotoxy(0,1);
37
38      // Read chars until newline
39      if (c != '\n')
40      {
41        // Concat char to buffer
42        // If maximum buffer size is reached, then reset i to 0
43        if (i < BUF_SIZE - 1)
44        {
45          buf[i] = c;
46          i++;
47        }
48        else
49        {
50          buf[i] = c;
51          i = 0;
52        }
53      }
54      else
55      {
56          lcd16x2_puts("A = ");
57          lcd16x2_puts(buf);
58        // Display received string to LCD
59
60        // Echo received string to USART2
61          USART2_PutString(buf);
62          USART2_PutChar(c);
63
64        // Clear buffer
65          memset(&buf[0], 0, sizeof(buf));
66          i = 0;
67      }
68    }
69  }
```

12.7 Other Required Functions:

```
102  void USART2_PutChar(char c)
103  {
104    // Wait until transmit data register is empty
105    while (!USART_GetFlagStatus(USART2, USART_FLAG_TXE));
106    // Send a char using USART2
107    USART_SendData(USART2, c);
108  }
109
```

```
110  void USART2_PutString(char *s)
111  {
112    // Send a string
113    while (*s)
114    {
115      USART2_PutChar(*s++);
116    }
117  }
118
119  uint16_t USART2_GetChar()
120  {
121    // Wait until data is received
122    while (!USART_GetFlagStatus(USART2, USART_FLAG_RXNE));
123    // Read received char
124    return USART_ReceiveData(USART2);
125  }
```

12.8 Programming in KEIL:

To program the Timer in PWM mode in Keil, the first step is to create a new uVision Project andselecting the device which is STM32F103C8 just like we already did in the previous labs.

After saving the project in your desired directory, the device will be selected in the „SelectDevice for Target" dialogue box.

After that, include the libraries of Core, GPIO, Startup, Framework, RCC & USART as shown inthe figure below:

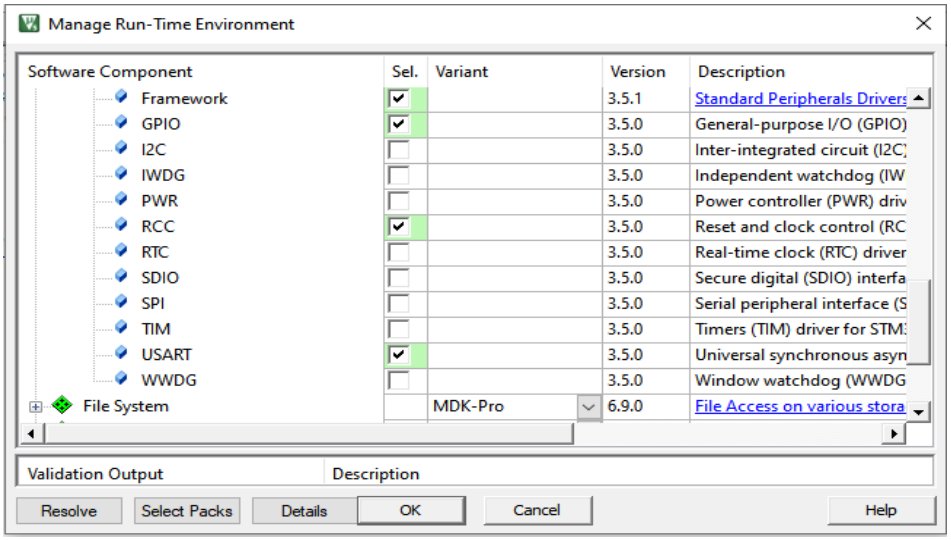


Figure 2: Libraries to Include

Then add a new item into your source group. Select the "C File (.c)" option, name your C fileand save it in your desired directory.

Lab Task (Psychomotor)

Activity 01:

Write a C-Program that can transmit and receive data using the Polling method of USART of STM32F103. The transmitted data should be displayed on Virtual Terminal of Proteus and the received data should be displayed on LCD. Show the simulation outputs to the lab instructor.

Activity 02:

Write a C-Program that can transmit and receive data using the Interrupt method of USART of STM32F103. The transmitted data should be displayed on Virtual Terminal of Proteus and the received data should be displayed on LCD. Show the simulation outputs to the lab instructor.

Lab Task (Cognitive)

Question 1:

A USART is configured for 9600 baud, 8 data bits, no parity and 1 stop bit (8N1).

a) Determine the number of bits per frame, the time taken to transmit one frame, and the maximum number of bytes that can be sent per second. **[4 Marks]**

b) Write the bit sequence (in transmission order) seen on the TX line when the ASCII character 'A' (0x41) is sent, including start and stop bits. Data is sent LSB first. **[3 Marks]**

c) Differentiate between polling and interrupt-based reception and name the status flag checked in the polling method. **[3 Marks]**

Lab Exercise and Summary

Summary should cover Introduction, Procedure, Data Analysis and Evaluation.

Student's Signature: _____ Date: _____

LABORATORY SKILLS ASSESSMENT (Cognitive)

Obtained Marks	**/10**

Instructor's Signature: _____ Date: _____

LABORATORY SKILLS ASSESSMENT (Psychomotor)

Psychomotor Activity 1: Experimental Setup

S. No	Criteria	Performance	Description	Marks
1	**USART Hardware & Terminal Setup (4)**	Level 4	Correctly prepares the required USART hardware and terminal setup for the experiment independently and according to the laboratory requirements.	4
		Level 3	Correctly prepares the required USART hardware and terminal setup with minor errors or assistance.	3
		Level 2	Partially prepares the required USART hardware and terminal setup and requires considerable assistance.	2
		Level 1	Shows limited ability to prepare the required USART hardware and terminal setup and requires substantial assistance.	1
2	**USART & Baud-Rate Configuration (4)**	Level 4	Correctly prepares the required USART and baud-rate configuration for the experiment independently and according to the laboratory requirements.	4
		Level 3	Correctly prepares the required USART and baud-rate configuration with minor errors or assistance.	3
		Level 2	Partially prepares the required USART and baud-rate configuration and requires considerable assistance.	2
		Level 1	Shows limited ability to prepare the required USART and baud-rate configuration and requires substantial assistance.	1
3	**Polling/Interrupt Parameters (2)**	Level 2	Correctly sets the required polling/interrupt parameters for the experiment.	2
		Level 1	Sets the basic polling/interrupt parameters with minor errors or assistance.	1
Total Obtained				**/10**

If a student is absent:

Criteria	Description	Marks
Absent	Student is absent from the laboratory activity.	**0**

Psychomotor Activity 2: Implementation + Results

S. No	Criteria	Performance	Description	Marks
1	**USART C Program (4)**	Level 4	Correctly implements the assigned USART C program independently while following appropriate laboratory practices.	4
		Level 3	Implements the assigned USART C program with minor errors or assistance and generally follows proper laboratory practices.	3
		Level 2	Partially implements the assigned USART C program and requires considerable assistance or repeated guidance.	2
		Level 1	Implementation of the assigned USART C program is largely incomplete, requires substantial assistance, or shows poor adherence to proper laboratory and safety practices.	1
2	**Execution & Debugging (2)**	Level 2	Correctly performs the required execution and debugging and verifies the expected operation.	2
		Level 1	Performs the basic execution and debugging with minor errors or assistance.	1
3	**Transmit/Receive Data Analysis (4)**	Level 4	Obtains correct transmit/receive data analysis and appropriately verifies and interprets the required experimental results.	4
		Level 3	Obtains mostly correct transmit/receive data analysis with minor errors in verification or interpretation.	3
		Level 2	Obtains partially correct transmit/receive data analysis with noticeable errors or omissions.	2
		Level 1	Transmit/receive data analysis is largely incorrect or incomplete and requires substantial assistance.	1
Total Obtained				**/10**

If a student is absent:

Criteria	Description	Marks
Absent	Student is absent from the laboratory activity.	**0**

Instructor's Signature: _____ Date: _____

LABORATORY SKILLS ASSESSMENT (Affective)

Affective Activity: Lab Report + Viva Voce

S. No	Criteria	Performance	Description	Marks
1	**Lab Report Completeness & Organization (4)**	Level 4	Report is complete, well-organized, properly presented, and demonstrates responsible completion of the assigned laboratory work.	4
		Level 3	Report is complete with minor omissions or organizational issues and demonstrates satisfactory responsibility for the assigned work.	3
		Level 2	Report is partially complete with several omissions and demonstrates limited responsibility for the assigned work.	2
		Level 1	Report has major omissions or poor organization and demonstrates minimal responsibility for the assigned work.	1
2	**USART Output & Conclusion (2)**	Level 2	The USART output and conclusion are correctly presented and discussed, demonstrating ownership of the assigned work.	2
		Level 1	The USART output and conclusion are presented with minor errors or omissions.	1
3	**Viva Voce (4)**	Level 4	Clearly explains the assigned experiment, USART frame format, polling and interrupt methods, program explanation, working, and results, demonstrating individual responsibility, participation, and effective communication.	4
		Level 3	Explains most of the assigned experiment and results correctly, with minor difficulty, and demonstrates good individual participation.	3
		Level 2	Explains basic aspects of the assigned experiment with some prompting and demonstrates basic participation.	2
		Level 1	Demonstrates limited understanding or involvement and requires substantial prompting to explain the assigned work.	1
Total Obtained				**/10**

The student has not submitted the report and has not appeared in the Viva Voce.

Criteria	Description	Marks
Lab Report and Results Not Submitted	Student has not submitted the report and results file.	**0**
Absent in Viva Voce	Student is absent from the viva voce.	**0**

Instructor's Signature: _____ Date: _____