VOL. 11, NO. 9, MAY 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

# FULL DUPLEX BIDIRECTIONAL UART COMMUNICATION BETWEEN PIC MICROCONTROLLERS

Dhineshkaarthi K., Sundar S., Vidhyapathi C. M. and Karthikeyan B.
M. Tech, School of Electronics Engineering, VIT University, Vellore, Tamilnadu, India
E-Mail: dhineshkaarthik@gmail.com

**ABSTRACT**

Universal Asynchronous Receiver Transmitter (UART) is a communication protocol which is mainly used for serial communication. In full duplex asynchronous UART mode, there is no need for clock synchronization between two devices. In this paper, we are analyzing and presenting the results of serially transmitting messages between two PIC16F877A Microcontrollers using UART protocol in full duplex mode. The simulation results show that, the full duplex bidirectional communication entirely depends upon the length of the characters transmitted and received.

**Keywords:** UART, PIC16F877A, full duplex communication, baud rate.

## INTRODUCTION

Depending upon the type of application used, the data communication between devices via several protocols have its own advantages and disadvantages based on parallel or serial modes, wired or wireless, speed or latency. Serial transmission between devices was the easiest low cost profile comparatively with parallel mode used. A large number of traditional peripheral communications in the early controllers and computing devices was done serially which required only two pins for transmitting and receiving. [1] It also reduces the material probes which often increased complexity and cost. [2] UART can be interfaced with Serial Peripheral Interface (SPI) for communication with the slave devices from the peripherals, which yielded optimum power constraints when used with System on Chip (SOC).[3-4] UART communication protocol can be implemented not only in general purpose Microcontrollers, but also in Digital signal processors (DSP) and Field programmable Gate arrays (FPGA) for communication to reduce circuit complexity and to increase flexibility. Universal Synchronous Asynchronous Receiver Transmitter (USART) protocol can be used in two modes, one is synchronous and the other is asynchronous. When there is synchronization of clock between two transceivers, it is called USRT protocol which offers half duplex communication. When the clock of the two transceivers is unsynchronized then it is called UART protocol which offers full duplex bidirectional way of communication. The two transceivers can send and receive data at once in full duplex mode in UART. [5] USART when used in asynchronous mode consumes less power than in synchronous mode. [6] Circuit synthesis of UART using Very high speed integrated circuit hardware description language (VHDL) for detection of error bits at behavioral level proves goods than at gate level modeling for different baud rates. [7] Tuning baud rate automatically to match the speed of communication between processor and peripherals improves reliability of transmission and reception. [8] Using UART enhanced capability of error correction and error detection is achieved by hamming code algorithm at the receiving end reducing the noise factor to an extent is implemented in VHDL. [9]

Computing at the same time with many core processors can be implemented with multiple UARTs. [10]Peripheral Interface Controller (PIC 16F877A) is one of the familiar Microcontroller that is of 8 bit word whose application is primarily embedded system and enormous in all domains of industries and in home safety and monitoring systems. It offers serial communication via Master Synchronous Serial Port (MSSP) and USART protocols. In this paper we are ensuring a communication between two PIC microcontrollers using UART.

## METHODOLOGY

The hardware channel interface between two Microcontrollers to ensure a valid communication through UART mode requires connecting Transmit pin (TX) of one Microcontroller to the Receive pin (RX) of the another one. The TX and RX pins in PIC16F877A Microcontrollers are 25$^{th}$ and 26$^{th}$ pin respectively and are connected as shown in Figure-1.


**Figure-1.** Block diagram of trapezoidal cross section.

The software used is MPLAB which is used to create the hex file of the software program that drives the PIC16F877A for the intended use. PIC downloader kit is used to dump the hex file into the microcontroller kit. The simulation of hardware is done in Proteus software.

## UART TRANSMITTER BLOCK

The main transmitter block of UART includes a Transmit Control register (TXSTA), a Transmit Data register (TXREG), a Transmit Shift register (TSR) and a buffer. In PIC16F877A, the Transmit shift register gets the data that is needed to be transmitted from the TXREG register and then sends it to the buffer where a byte of data is converted into bits before sending it to the transmitter

pin. The TXREG register can be accessed by the user for mentioning the character to be sent via UART while the Transmit Shift register is not accessible to the user. The flag bit TXIF and TRMT indicates the status of TXREG register and Transmit Shift Register respectively, whether those registers are empty or busy. While programming the next character to be sent, it is mandatory to check the status of Transmit Shift Register or Transmit Data Register. The flow of data is in form of bits from least significant bit to most significant bit with Non Return to Zero data format. In PIC16F877A, the Transmit Control register should be programmed to work in asynchronous mode clearing its bit4. RC6 pin of the Microcontroller should be cleared to declare it as output pin. The transmitter block is shown in Figure-2.

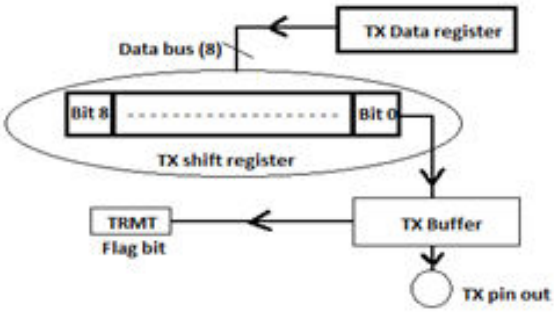


**Figure-2.** Block diagram of UART Transmitter.

## UART RECEIVER BLOCK

The main receiver block of UART includes a Receive Control register (RCSTA), a Receive Data register (RCREG), a Receive Shift register (RSR) and a buffer. In PIC16F877A the Receive shift register receives the data through receive buffer from Receive pin. As Receive shift register is not accessible to the user, the data is accessed after it gets into RCREG register as indicated in the Figure-3. The data can be moved to a declared variable for further processing. While receiving the next character that was sent by the transmitter, it is mandatory to check the status of Receive Shift register or Receive Data Register. In PIC16F877A, the Receive control register should be programmed to work in asynchronous mode clearing its bit4. RC7 pin of the microcontroller should be set to declare it as input pin. The receiver block diagram is shown in Figure-3.

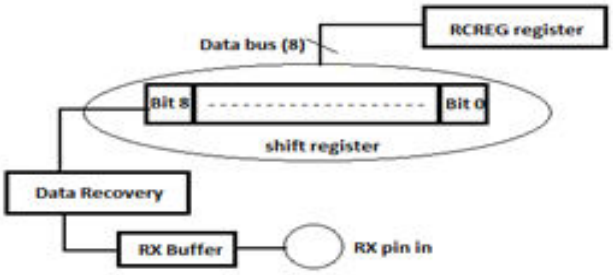


**Figure-3.** Block diagram of UART Receiver.

## REGISTER CONFIGURATIONS FOR UART MODE

### Setting SPBRG register value

To ensure a valid communication it is mandatory to set the same baud rate for both the transceivers. The high speed asynchronous communication is achieved by sending a value to Specify Baud rate value Register (SPBRG) and corresponding baud rate is calculated by the following formula,

$$Baud\ rate = \frac{F}{[(SPBRG\ reg\ value) + 1] * 16} \quad (1)$$

Where,

**F - Frequency of the oscillator**

### Configuring TXSTA Register

The bit configuration of TXSTA register is as shown in Figure-4, where green box indicates the particular bit is set and red box indicates the bit is cleared.

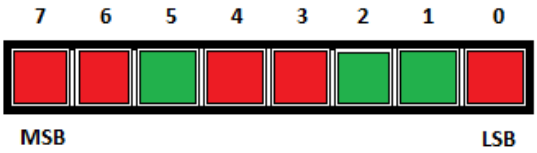


**Figure-4.** Bit configuration of TXSTA Register.

**Bit 7:** Clock source select bit, which can be ignored in asynchronous mode
**Bit 6:** This should be cleared since we are transmitting 8 bits of data
**Bit 5:** Setting this bit enables transmission
**Bit 4:** Clearing this bit ensures asynchronous mode of transmission
**Bit 3:** This bit is a reserved bit
**Bit 2:** Setting this bit enables baud rate selection for speedy transmission of data
**Bit 1:** Indicates the status of the flag bit
**Bit 0:** Clearing this bit does not include parity bit for transmission

### Configuring RCSTA Register

The bit configuration of RCSTA Register is as shown in Figure-5.

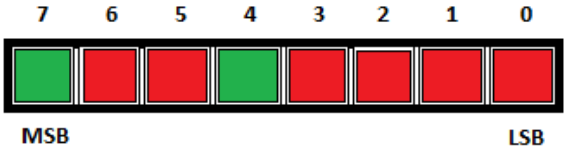


**Figure-5.** Bit configuration of RCSTA Register.

**Bit 7:** Setting this bit configures the TX/RX pin for serial communication
**Bit 6:** This bit should be cleared since we are receiving 8 bits of data
**Bit 5:** Master/slave select bit, which can be ignored in asynchronous mode
**Bit 4:** Setting this bit enables continuous reception
**Bit 3:** This bit is cleared since we are receiving 8 bit of data
**Bit 2:** Clearing this bit ignores framing error if it occurs
**Bit 1:** Clearing this bit ignores overrun error if it occurs
**Bit 0:** Clearing this does not receive parity bit

## SIMULATION RESULTS AND DISCUSSION

In this paper, Proteus software is used for simulating the results of hardware. The implementation of how characters are transmitted and received using full duplex mode in a single transceiver is shown in Figure-6.

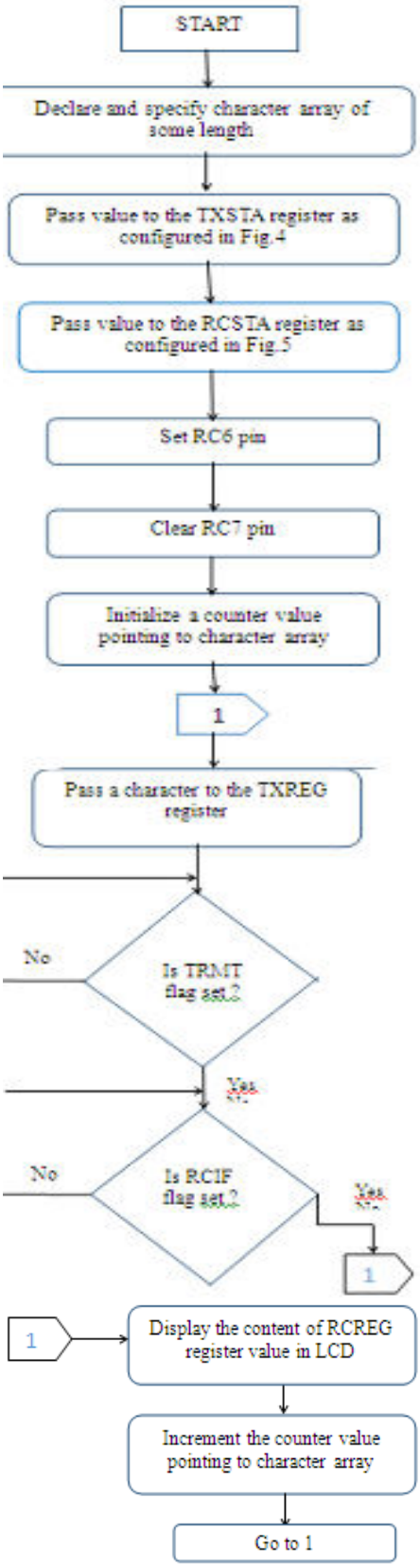


**Figure-6.** Flow chart of the program.

The virtual model of the hardware is created in Proteus software where, three different cases are studied by varying the length of the characters in two transceivers.

**Case-1**

In this case, the same word is transmitted by both transmitters as indicated in Figure-7.

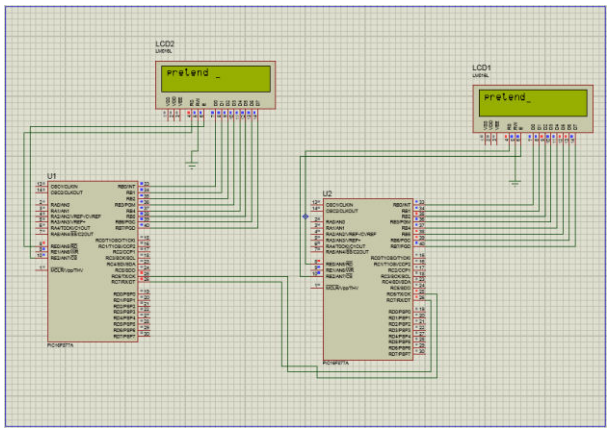


**Figure-7.** Two Transmitters transmit same word.

**Case-2**

In this case, different words but of same word size is transmitted by both transmitters as indicated in Figure-8.

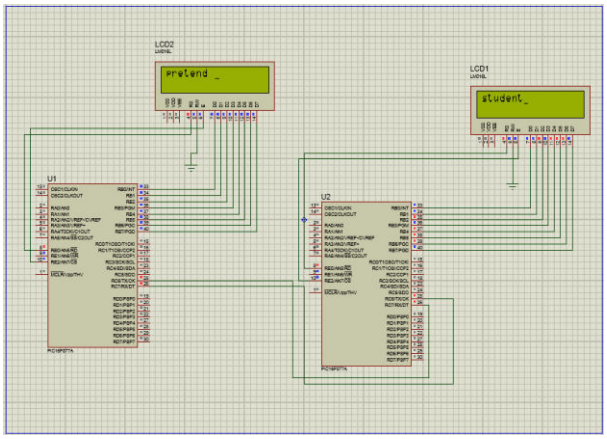


**Figure-8.** Two Transmitters transmit different words of same size.

**Case-3**

In this case, different words but of different character array size is transmitted by both transmitters as indicated in Figure-9.

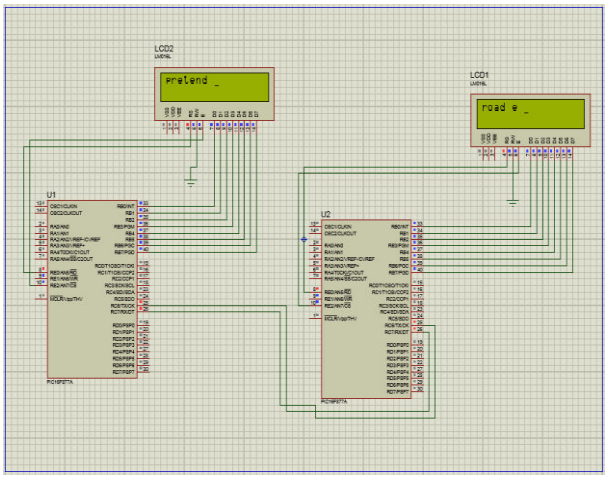


**Figure-9.** Two Transmitters transmit different words of different size.

## CONCLUSIONS

In this paper, a transceiver model is simulated in Proteus software for UART serial communication between two PIC Microcontrollers. The following are the conclusions drawn based on the simulation results,

- When the words of same size are transmitted, the received words will be same as transmitted by the other transceiver.
- When the words of different size are transmitted, the received words will not be same as transmitted by the other transceiver.

In PIC microcontroller, for a meaningful transmission, the length of the characters to be transmitted must be same when UART is used in full duplex mode.

## REFERENCES

[1] Yurcik, William J. 2015. Serial and Parallel Transmission. Computer Sciences 2002, <http://www.encyclopedia.com>.

[2] T. Praveen Blessingtoni, B. Bhanu Murthy, G.V. Ganesh, T.S.R Prasad. 2012. Optimal Implementation of UART-SPI Interface In: Soc. International Conference on Devices, Circuits and Systems.

[3] Yongcheng Wang, Kefei Song. 2011. A New Approach to Realize UART. International Conference on Electronic and Mechanical Engineering and Information Technology.

[4] Lawrence Wong. 1993. Hardware UART for the TMS320C3x. Texas instruments, 1993 <www.ti.com/lit/an/spra223.pdf>.

[5] Dipanjan Bhadra, Vikas S. Vij, Kenneth S. Stevens. 2013. A Low Power UART Design Based on

Asynchronous Techniques. International Midwest symposium on Circuits and Systems.

[6] Norhuzaimin J, Maimun H.H. 2005. The design of high speed UART. Asia -Pacific conference on Applied Electronics.

[7] He Chun-zhi, Xia Yin-shui, Wang Lun-yao. 2007. A Universal Asynchronous Receiver Transmitter Design. International conference on Electronics, Communications and Control.

[8] Muppalla S, Vaddempudi K.R. 2015. A novel VHDL implementation of UART with single error correction and double error detection capability. International conference on Signal Processing and Communication Engineering Systems.

[9] Sambamurthy N, Kamaraju M. 2015. Design and implementation of concurrent computing multiprocessor core architecture with multi UART. International conference on Electrical, Electronics, Signals, Communication and Optimization.

[10] Microchip Technology Inc. PIC 16F87XA. 2001-2013.