Lab Manual

CYSL – Embedded System Security Lab

Topic: Asynchronous Serial Communication

Class: **BSCYS**

Semester: **IV**

Session: **Spring, 2022**

Instructor: Dr. Naveed Bhatti

Lab Instructor: Mr. Salman Ahmad

AIR UNIVERSITY

Air University Islamabad

FACULTY OF COMPUTING & ARTIFICAL INTELLIGENCE

Department of Cyber Security

Objectives

In this lab, you will understand:

- UART (Universal Asynchronous Receiver/Transmitter), Structure,
- UART Timing Diagram, (start, parity, data and stop bits), and its library
- Sending Text over Serial (From Microcontroller to Host PC)

1. Description:

Embedded electronics is all about interlinking circuits (processors or other integrated circuits) to create a symbiotic system. In order for those individual circuits to swap their information, they must share a common communication protocol. Hundreds of communication protocols have been defined to achieve this data exchange, and, in general, in serial communication, it can be divided into one of two categories: synchronous or asynchronous.

2. UART

A universal asynchronous receiver/transmitter (UART) is a block of circuitry responsible for implementing serial communication. Essentially, the UART acts as an intermediary between parallel and serial interfaces. On one end of the UART is a bus of eight-or-so data lines (plus some control pins), on the other is the two serial wires - RX and TX.

2.1 Simple UART Structure

Two sides to the communication, the transmitter Tx and the receiver Rx. Data is serialized by Tx, deserialized by Rx as shown below.

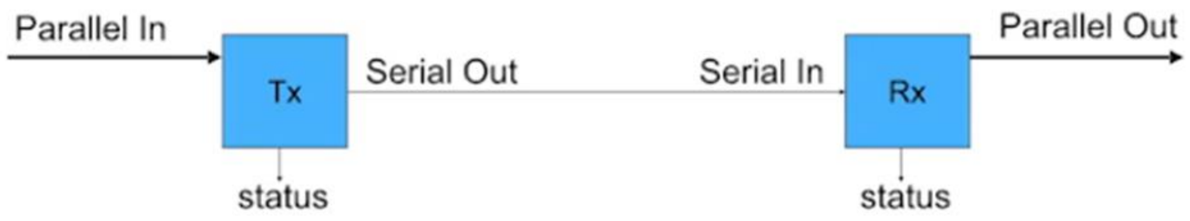

Fig 1: Simplified UART structure

2.2 UART Timing Diagram

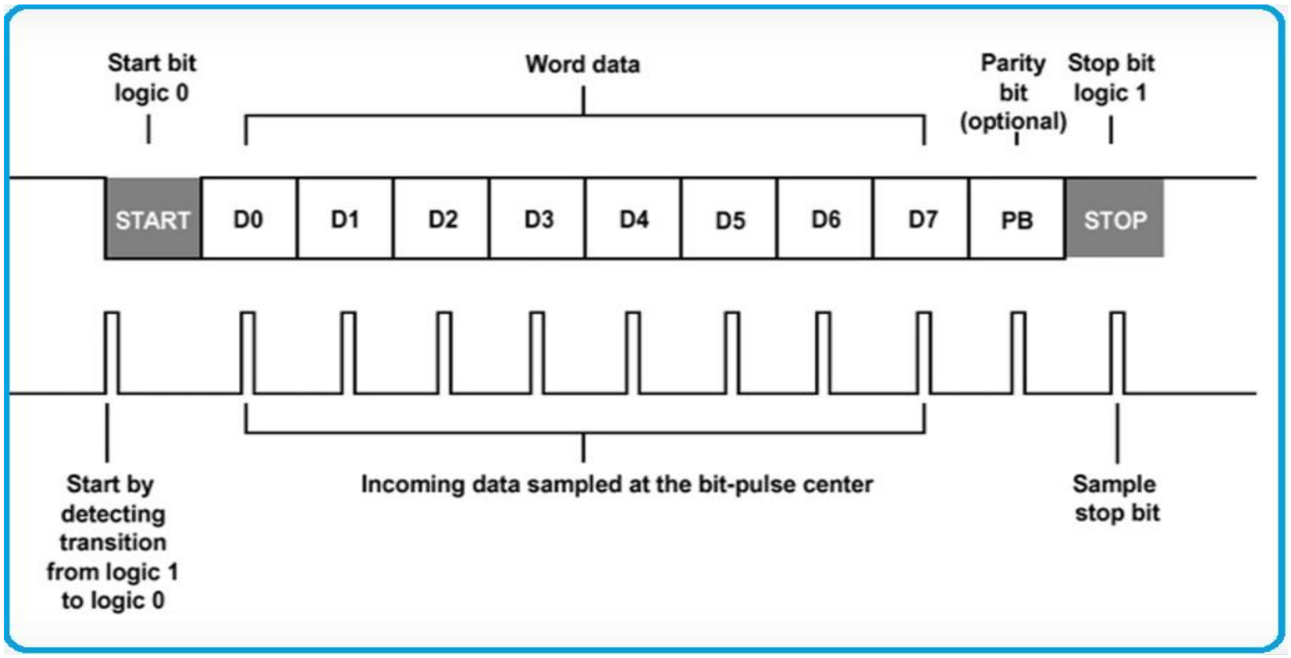

Fig 2: UART Timing Diagram

This is an approximate timing diagram, where the time is the X-axis. This is an example of sending one byte of data via UART. The first bit is the start bit and that initiates the transfer. The next bits are the data (chunk of 8-bits) which is common but you can change that. Then the last bit is the stop bit. This is how the signaling bits are needed by their receiver to know when communication is starting and when it is ending. Note that the parity bit is optional here, since the transmission medium is assumed to be error-prone. This bit is used to check for error.

2.3 UART library

Tx and Rx pins at 0 and 1 of the Arduino UNO are internally connected to the Tx and Rx pins of the USB port.

- **Serial.begin(speed) or Serial.begin(speed, config)**
  - Where speed is the baud rate
  - And config sets data, parity, and stop bits (customized)

3. Lab Experiments

3.1 Task 1: Sending Text over Serial

Write a program that allows the user to control any output pin of the Arduino. When the program is started, the pin should be low. The user should open the serial monitor to communicate with the Arduino. If the user sends the character '1' through the serial monitor then the pin goes to high. If the user sends the character '0' through the serial monitor then the pin again goes to low.

3.1 Components Required

1. Arduino UNO
2. Breadboard
3. Resistors 470, 10 k ohm
4. LED
5. Buzzer

3.2 Procedure:

1. Connect Arduino to the PC using the USB cable supplied. (Now the serial communication will be performed using this data cable)
2. Launch the Arduino IDE. Set the comm port

   Tools > Port > Comm___ (It can be verified from the device manager)
3. Set the corresponding microcontroller board

   Tools > Boards > Arduino AVR boards > Arduino UNO
4. Build the circuit using led and buzzer at the two digital output pins of Arduino
5. Compile the upload the program
6. Open the serial monitor and send ASCII characters from it to view the status of output devices.

3.3 Arduino Code:

```
// Task 1

int outRead = 8;

void setup()
{
  pinMode(outRead, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  if (Serial.available() > 0)
  {
    char state = Serial.read();
    if (state == '1')
     {
      digitalWrite(outRead, HIGH);
     }
    if (state == '0')
     {
      digitalWrite(outRead, LOW);
     }
  }
}
```

Assignment 7:

Write a program that do the following when user sends the

- character '1', it turns on the actuator 1
- character '2', it turns off the actuator 1
- character '3', it turns on the actuator 2
- character '4', it turns off the actuator 2

Note: If pending, complete the previous classes assignments today and submit on GCR.