Experiment – 6

UART Port Based Serial Data Communication

(Read every task and sub-task and work accordingly. Put Tick mark (√) when done.)

Task – 6.1 Architecture of UART, SUART, and Serial Mnitor

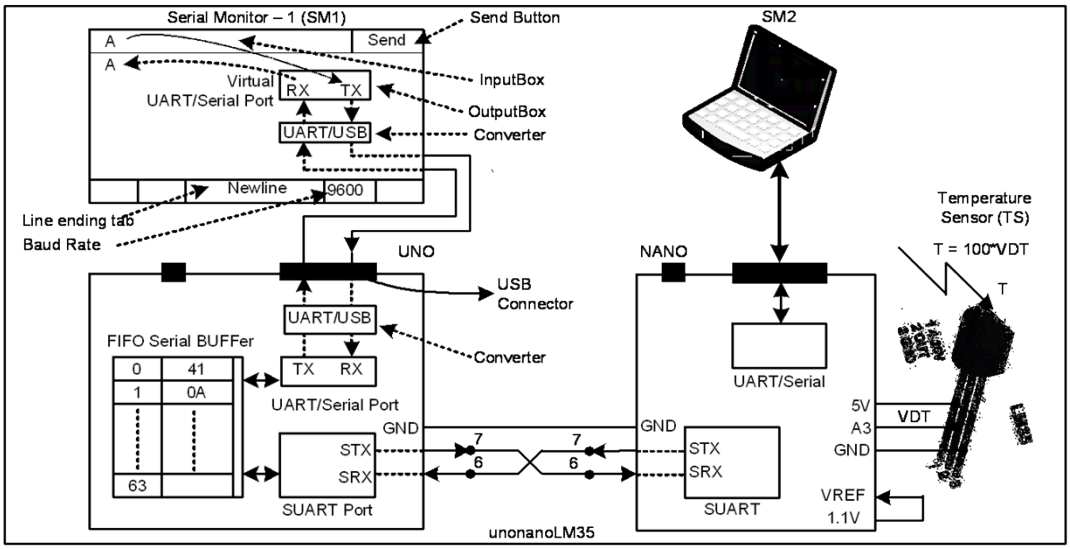

Figure-6.1: *UART and SUART Ports of ATmega328P MCU*

Task–6.2 Read the List of Functions for UART Port based Serial Data Communication

Sn	Function/Method	Description	Comment
1	serialEvent();	When a character arrives in receiver of UART, MCU is interrupted and then goes to serialEvent() subroutine to collect character from buffer. It does not work with Soft UART.	
2	Serial.available();	The *Serail.available()* function allows to know how many bytes of data are still there in the buffer.	
3	Serial.read();	This code reads a data byte from buffer..	
4	Serial.write(arg); Serial.write(array, size of array);	This function writes single byte/multi byte data into buffer for onward transmission.	
5	Serial.print(arg);	This function writes a character into buffer for onward transmission to OutputBox of Serial Monitor..	
6	Serial.println();	This code keeps two characters – CR (0x0D) first and line feed (0x0A) for onward transmission to OutputBox of SM.	
7	Serial.println(arg);	This code keeps three byte data – value **arg,** CR, and LF for onward transmission OutputBox of Serial Monitor..	
8	Serial,begin(arg);	Sets the Baud Rate (Bd) of the communication channel.	
9	Serial.end();	Ends the serial communication. The TX and RX lines could be used as general digital IO lines.	
10	SoftwareSeral.h	Header File which supports software UART)SURT) Port.	
11	SoftwareSerial SUART(5, 6);	Creates SUART Port with SRX Line attached at DPin-5 and STX Line attached at DPin-6.	

Task – 6.3 Exchanging Characters between Serial Monitor and UNO

(1) [Single Charcater] Open IDE and create the following sketch to receive character **A** from the InputBox of Serial Monitor (Fig-6.1) by UNO and send it back to the OutputBox of the Serial Monitor. Select 'No line ending" option for the "Line ending tab" of Serial Monitor. Save the program as P631.

```
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  byte n = Serial.available(); //n = 1
  if ( n != 0)  //Buffer contains at least one data byte
  {
     char ch = Serial.read();
     Serial.print(ch);
  }
}
```

(2) [Multiple Charcaters] Open IDE and create sketch to receive the message "AUST" from Serial Monitor by UNO , save it in an array named *myArray[]* and send it back to Serial Monitor. Choose *Newline* option for the Line ending tab of the Serial Monitor. Save the program as P632.

```
char myArray[10];
int arrayIndex = 0;

void setup()
{
  Serial.begin(9600);
}

void loop()
{
  byte n = Serial.available();
  if(n != 0)
  {
    char ch = Serial.read();
    if(ch != '\n') //not Newline charcater
    {
      myArray[arrayIndex] = ch;
      Serial.print(ch);
      arrayIndex++;
    }
    else
    {
      Serial.println();
      myArray[arrayIndex] = '\0'; //insert null-charcater
      Serial.println(myArray);
      arrayIndex = 0;  //reset
    }
  }
}
```

(3) **[***use of* **atoi()** (ASCII TO Integer) *function to extract decimal number from incoming ASCII coded data***]** Create sketch to receive 1234 (with Newline option) from Serial Monitor, receive it and save in an array, convert them into a numerical integer value to save in variable ***y***, and then send ***y*** to the OutputBox of the Serial Monitor. Save the program as P633.

```
char myData[10];

void setup()
{
  Serial.begin(9600);//UNO's UART is active
}

void loop()
{
  byte n = Serial.available(); //n = 1
  if ( n != 0)
  {
     byte m = Serial.readBytesUntil('\n', myData, 10);//arg3 = arraySize
     // m = number of charcaters stored in Buffer except Newline ;
     int x = atoi(myData);
     Serial.println(x, DEC);
   }
}
```

(4) **[***use of* **atof()**(ASCII TO Float) *function to extract floating point number from incoming ASCII coded data***]** Create sketch to receive 6712.56 (with Newline option) from Serial Monitor, receive it and save in an array, convert them into a numerical floating point number to save in variable ***fNum***n, and then send the ***fNum*** to the OutputBox of the Serial Monitor. Save the program as P634.

```
char myData[10];

void setup()
{
  Serial.begin(9600);
}

void loop()
{
  byte n = Serial.available(); //n = 1
  if ( n != 0)  //Buffer contains at least one data byte
  {
     //keep receiving data bytes until Newline is found
     byte m = Serial.readBytesUntil('\n', myData, 10);//arg3 = arraySize
     // m = number of charcaters stored in Buffer except Newline
     float x = atof(myData);
     Serial.println(x, 2);
  }
}
```

(5) Create sketch to start reading characters from Serial Buffer when it has finished accumulating only four characters. Save sketch as P635.

i. Open IDE and create sketch at Bd = 9600.
ii. Upload sketch.
iii. Open Serial Monitor (SM) at Bd = 9600.

v. Enter **Ahsanullah** in the InputBox of SM.
vi. Check that only **Ahsa** has appeared on the OutputBox of Serial Monitor.

Task - 6.4 Programming of the UART Port

Write sketch so that when you send message "**Turn ON L**" from SM of Fig-6.2 with Newline option, then **L** of UNO will be ON. The message **L is ON** will also appear SM. Save sketch as P64.

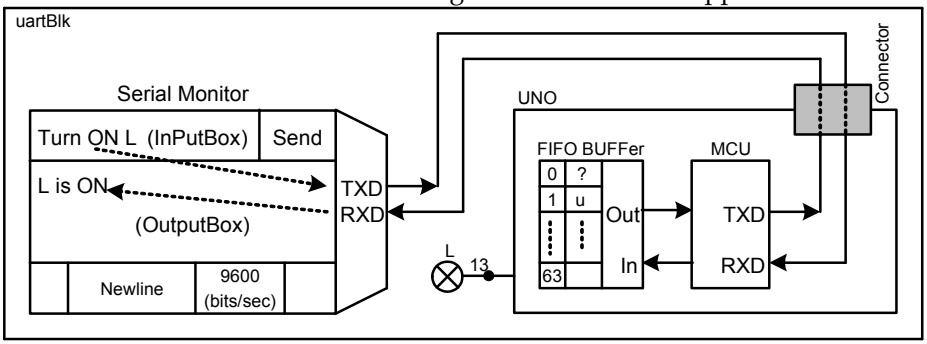

Figure-6.2:

```
char recString[10]; //array to store received characters
int i = 0;          //array index

void setup()
{
  Serial.begin(9600);
  pinMode(13, OUTPUT);
}

void loop()
{
  byte n = Serial.available();  //check that a character has arrived
  if(n !=0)                     //a character has arrived
  {
    char y = Serial.read();     //put the ASCII of character in y
    if(y == '\n')  //C code for Newline character is found;
    {
      recString[i] = '\0';                    //insert null-byte
      bool m = strcmp(recString, "Turn ON L"); //comparing two arrays
      if(m == 0x00)               //strng matches
      {
        digitalWrite(13, HIGH);   // L is ON
        Serial.print("L is ON");  //message on Serial Monitor
        while(1);                 //wait here
      }
    }
    else
    {
      recString[i] = y; //Newline character has not arrived; keep saving characters
      i++;
    }
  }
}
```

Task – 6.5 [use of **SUART** (Software UART Port) Port**]** to transmit 0x12 from UNO to NANO.NANO receives the data byte and shows on SM2.

1. Connect UNO and NANO using SUART Port as per Fig-6.1.
2. Include the following lines in the sketch to create and activate SUART Port.

```
#include<SoftwareSerial.h>
SoftwareSerial SUART(SRX, STX);
SUART.begin(9600);
```

3. Upload the following sketch in UNO/Sender. Save sketch as E653M.

```
#include<SoftwareSerial.h>
SoftwareSerial SUART(6, 7);  //SRX = DPin-6, STX = DPin-7

void setup()
{
  Serial.begin(9600);
  SUART.begin(9600);
}

void loop()
{
  SUART.write(0x12);
  delay(1000);
}
```

4. Upload the following sketch in NANO/Receiver. Save sketch as E654S.

```
#include<SoftwareSerial.h>
SoftwareSerial SUART(6, 7);  //SRX = DPin-6, STX = DPin-7

void setup()
{
  Serial.begin(9600);
  SUART.begin(9600);
}

void loop()
{
  byte n = SUART.available();
  if (n != 0)
  {
    byte y = SUART.read(); //y = 0x12
    //--converting 0x12 into 0x31 and 0x32 to display
    byte z = y;
    z = z >> 4;
    z = z + 0x30; //z = 0x31 = ASCII code of 1
    Serial.print(z); //shows: 1
    //--------
    y = y & 0b00001111;  //y = 0x02
    y = y + 0x30; // y = 0x32 = ASCII code of 2
    Serial. prinln(y); //shows: 2
  }
}
```

5. Check that 12 appear on SM2 at 1-sec interval.
6. Reduce codes of the loop() function of Step-4 by replacing by the two `Serial.print()` commands by a single `Serial.println()` command.

Task – 6.6 [use of **SUART** Port**]** to transmit a string from UNO to NANO

(1) UNO will send this message "Forum" to NANO using SUART Port (Fig-6.3) at 1-sec interval

(2) The Start-Marker (**STX** = 0x02) will be sent first to mark the beginning of message.

(3) Then the string "Forum" will be sent.

(4) Then a checksum (CKHSUM) byte will be sent. CHKSUM is computed by adding all the bytes of the message, discarding the carry, and then taking 2's complement of the remainder.

(5) Then send (**ETX** = 0x03) as the End-Marker of the message.

(6) At the receiver side, the NANO detects the **STX** first and then will collect data bytes of the arrived message including the CHKSUM. Reception will end when **ETX** is detected. After that all the bytes of the received message will be added, discard the carry, and then adds with the received CHKSUM. The result should be zero indicating valid message and then NANO will flash LED1 (Fig-1).

(7) Build the following circuit of Fig-1 using SUART Port (Software UART Port).

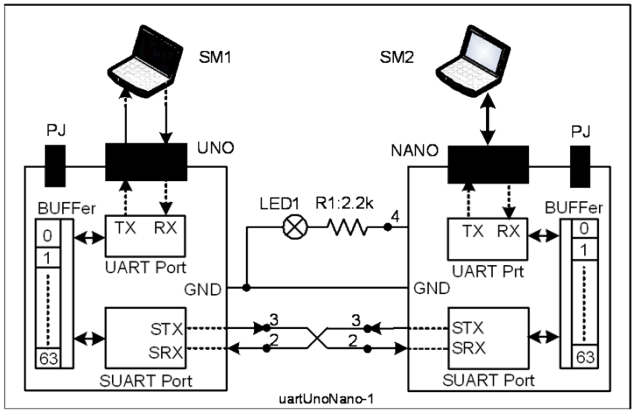

Figure-6.3:

(8) Upload the following sketch in UNO. Save the sketch as E668.

```
#include<SoftwareSerial.h> //Hedaer file that includes ready-made functons
SoftwareSerial SUART(2, 3); //SUART port is created with SRX = DPin-2, STX = DPin-3
#define STX 0x02  //STX = Start of Text
#define ETX 0x03  //ETX = End of Text

char myMsg[] = "Forum"; //Message/sting to be sent
byte CHKSUM = 0;

void setup()
{
  Serial.begin(9600); //UART Port is enabled
  SUART.begin(9600);  //SUART port is enabled
}

void loop()
```

```
{
  SUART.write(STX); //StartMarker is end
  SUART.write(myMsg, sizeof myMsg - 1); //bytes of the message are sent
  //-------------------------
  for (int i = 0; i < sizeof myMsg - 1; i++)
  {
    CHKSUM += myMsg[i];
  }
  CHKSUM = ~CHKSUM + 1;  //computing check sum = F7
  SUART.write(CHKSUM);  //sending CHKSUM to NANOCHKSUM = 0;
  CHKSUM = 0;
  //----------------------------
  SUART.write(ETX); //EndMarker is sent
  delay(1000);  //test interval
}
```

(9) Upload the following sketch in NANO. Save the sketch as E669.

```
#include<SoftwareSerial.h>
SoftwareSerial SUART(2, 3);
#define STX 0x02  //STX = Start of Text
#define ETX 0x03  //ETX = End of Text
char myMsg[10];
byte CHKSUM = 0;

void setup()
{
  Serial.begin(9600);
  SUART.begin(9600);
  pinMode(4, OUTPUT);
}

void loop()
{
  byte n = SUART.available();
  if ((n == 1) && (SUART.read() == STX))//StartMarker is detected
  {
    byte m = SUART.readBytesUntil(ETX, myMsg, 10); //save all except ETX
    //-----------------------
    for (int i = 0; i < m - 1; i++)
    {
      CHKSUM += myMsg[i]; //add all the bytes of received message
    }
    if (CHKSUM + myMsg[5] == 0) //the result should be zero
    {
      digitalWrite(4, HIGH);  //flash LED1 at DPin-4
      delay(200);
      digitalWrite(4, LOW);
      delay(200);
    }
    CHKSUM = 0;  //reset CHKSUM variable
  }
}
```

(10) Observe that LED1 of Fig-6.3 flashes at 1-sec interval indicating reception of valid string.

Task – 6.7 [use of **SUART** Port**]** to transmit signal from NANO to UNO

Connect LM35 temperature sensor with NANO as per Fig-6.1. Now, perform the following steps:

(1) Create sketch for NANO to acquire temp signal from LM35 sensor and show it on SM2 with 1-digit precision. NANO will also send the temp signal to UNO using SUART(6, 7) Port. Save program as P671nano.

(2) Create sketch for UNO to acquire temp signal from NANO using SUART(6, 7) Port and show it on SM1 with 1-digit precision. Save program as P672uno.