Lab Manual for Embedded System Design

Lab No. 11

Serial USART Interface Programming using Arduino UNO Board

Objectives

Understanding the basic concept of serial usart interface programming and implementation of concepts on Arduino UNO and IDE

LAB # 11

Serial USART Interface Programming using Arduino UNO Board

Introduction

Used for communication between the Arduino board and a computer or other devices. All Arduino boards have at least one serial port (also known as a UART or USART), and some have several.

BOARD	USB CDC NAME	SERIAL PINS	SERIAL1 PINS	SERIAL2 PINS	SERIAL3 PINS
Uno, Nano, Mini		0(RX), 1(TX)			
Mega		0(RX), 1(TX)	19(RX), 18(TX)	16(RX), 17(TX)	15(RX), 14(TX)
Leonardo, Micro, Yún	Serial	0(RX), 1(TX)			
Uno WiFi Rev.2		Connected to USB	0(RX), 1(TX)	Connected to NINA	
MKR boards	Serial		13(RX), 14(TX)		
Zero	SerialUSB (Native USB Port only)	Connected to Programming Port	0(RX), 1(TX)		
Due	SerialUSB (Native USB Port only)	0(RX), 1(TX)	19(RX), 18(TX)	16(RX), 17(TX)	15(RX), 14(TX)
101	Serial		0(RX), 1(TX)		

On Uno, Nano, Mini, and Mega, pins 0 and 1 are used for communication with the computer. Connecting anything to these pins can interfere with that communication, including causing failed uploads to the board.

You can use the Arduino environment's built-in serial monitor to communicate with an Arduino board. Click the serial monitor button in the toolbar and select the same baud rate used in the call to begin().

Serial communication on pins TX/RX uses TTL logic levels (5V or 3.3V depending on the board). Don't connect these pins directly to an RS232 serial port; they operate at +/- 12V and can damage your Arduino board.

To use these extra serial ports to communicate with your personal computer, you will need an additional USB-to-serial adaptor, as they are not connected to the Mega's USB-to-serial adaptor. To use them to communicate with an external TTL serial device, connect the TX pin to your device's RX pin, the RX to your device's TX pin, and the ground of your Mega to your device's ground.

Time Boxing

Activity Name	Activity Time	Total Time
Login Systems + Arduino & Proteus	3 mints + 5 mints	8 mints
Walk through Theory & Tasks	60 mints	60 mints
Implement Tasks	80 mints	80 mints
Evaluation Time	30 mints	30 mints
	Total Duration	178 mints

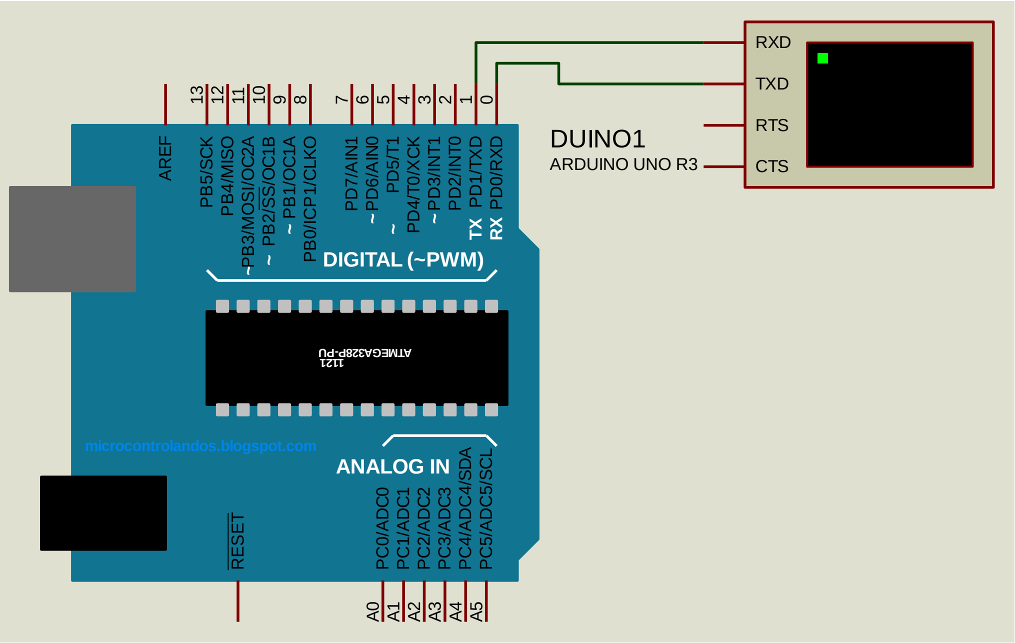
Objectives

This Lab exercise delivers the idea/concept of:

- Understand the purpose/ advantage of using Arduino UNO.
- Understanding Usart Interface peripheral programming.

Lab Tasks/Practical Work

1. In this task, we will write a program to print a string on serial monitor.



Code:

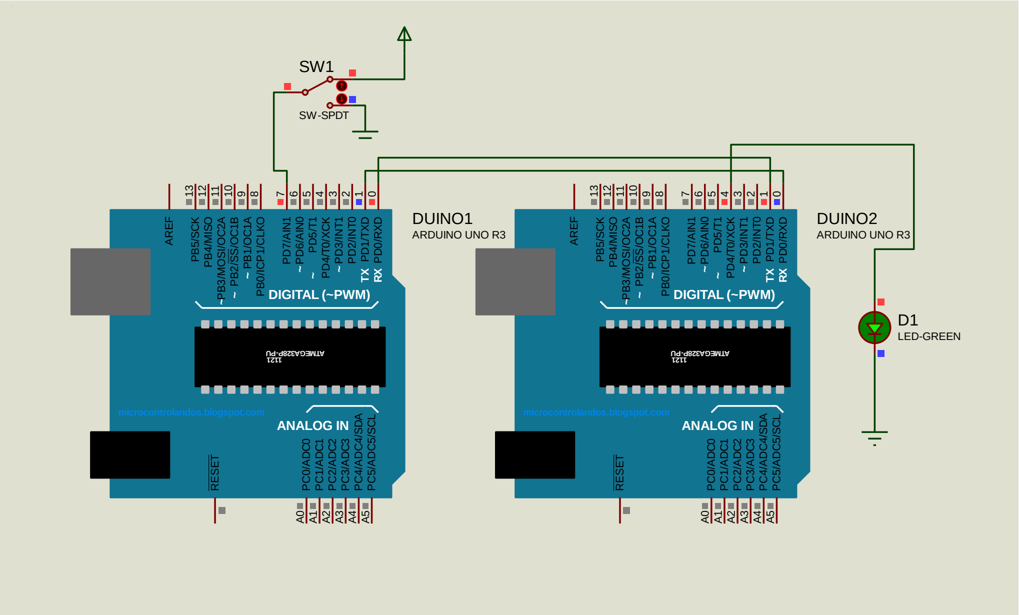
```
void setup() {
  // put your setup code here, to run once:
Serial.begin(9600); // begin serial communication , define baud rate.
}


void loop() {
  char string[8] =  "Embedded"  ;
int  index=0;


  for(index=0 ; index <9 ; index++)
  {
    Serial.print(string[index]);
  }
  // put your main code here, to run repeatedly:
Serial.println(" ");
delay(500);


}
```

2. In this task we will control the LED on “Arduino 2” using SPDT switch on “Arduino 1” through UART communication.



Code (Arduino 1):

```
int SPDT =7;
void setup() {

  pinMode(SPDT,INPUT);
  Serial.begin(9600);
  // put your setup code here, to run once:

}
void loop() {
  // put your main code here, to run repeatedly:
int button_status = digitalRead(7);

Serial.write(button_status);
//delay(500);
}
```

Code (Arduino 2):

```
int LED = 4;
void setup() {
  // put your setup code here, to run once:
pinMode(LED,OUTPUT);
Serial.begin(9600);
}

int x =0;
void loop() {
  if(Serial.available() > 0)
  {
  // put your main code here, to run repeatedly:
x = Serial.read();
if (x == 1)
{digitalWrite(LED, HIGH);
}
else
{ digitalWrite(LED,LOW);
}
}
}
```

2. Develop a Program which provide serial usart interfacing programming using Arduino UNO board.