Indian Institute of Technology Bombay
CS684: Embedded Systems

# Serial Communication - UART

ERTS Lab, CSE Department

## 1 Lab Objective

1. Understand Serial Communication by UART protocol and its Interrupts
2. Get acquainted with using the UART protocol and its implementation

## 2 Pre-requisite

This lab assumes you have completed Lab-2 and Lab-3, which means you are aware of the working of the PWM, Timers, Hardware and Software Interrupts of ATmega2560.

## 3 Problem Statement

In this lab you have to use **Joystick module**, **Servo motor** and on-board **RGB LED** provided in the lab kit and interface them with the board.

**Note:** For this Lab, you can refer to the **7_UART** experiment from the **Quick Bytes** page of the website, but you are not allowed to use the **uart.c** and **uart.h** library for implementing the Problem Statements of this Lab. You can refer to the Atmega2560 datasheet https://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561_datasheet.pdf.

1. Write a program to configure the **UART0** in ATmega2560 with these settings:
   - Asynchronous USART Mode
   - No Parity
   - 1 Stop Bit
   - 8-bit Character Size

- Baud rate: 115200

You have to use any Serial Terminal application like **PuTTY**. Using this application in Serial mode, you have to send text: **Hello, we are Team - <Team_Name>** from the PC to the board. The board should reply the same text to the PC.

**Note:** Make sure the **S3** switch on board is towards **W_AVR** side.

2. Interface the **Joystick module**, **Servo motor** on the board as you did in Lab-2. You also have to use Timers as you did in Lab-3. Write a program such that when following message exchange happen between PC to board, the corresponding message from board to PC and actions to be taken is as described below.

   **Note:** The **Message from board to PC** and the **Action to be taken** should continue for infinity unless there is a change in **Message from PC to board**. You have to use UART Interrupt.

   (a) Send the ADC data of Joystick module to PC upon request
   - Message from PC to board: **'J'**
   - Message from board to PC:
     **"X = <Vx_ADC_value>;**
     **Y = <Vy_ADC_value>;**
     **SW = <ON_OR_OFF>"**
   - Action to be taken: – **(No action)**

   For example, Message from board to PC: **"X = 512; Y = 512; SW = ON"**

   (b) Set the angle of Servo motor by the data from PC
   - Message from PC to board: **"S|<angle_value>"**
   - Message from board to PC: **"Servo at <angle_value>"**
   - Action to be taken: **Servo motor should rotate to the <angle_value>**

   For example, if Message from PC to board: **"S|90"**, then Message from board to PC: **"Servo at 90"** and Action to be taken: Servo motor should rotate to the angle 90°.

   (c) Set the brightness level of RED LED by the data from PC
   - Message from PC to board: **"R|<brightness_level>"**
   - Message from board to PC: **"RED LED at brightness: <brightness_level>"**
   - Action to be taken: **RED LED should glow with brightness: <brightness_level>**

   For example, if Message from PC to board: **"R|127"**, then Message from board to PC: **"RED LED at brightness: 127"** and Action to be taken: RED LED should glow with brightness: 127.

   (d) Set the brightness level of GREEN LED by the data from PC

- Message from PC to board: **"G|<brightness_level>"**
- Message from board to PC: **"GREEN LED at brightness: <brightness_level>"**
- Action to be taken: **GREEN LED should glow with brightness: <brightness_level>**

For example, if Message from PC to board: **"G|0"**, then Message from board to PC: **"GREEN LED at brightness: 0"** and Action to be taken: GREEN LED should glow with brightness: 0.

(e) Set the brightness level of BLUE LED by the data from PC

- Message from PC to board: **"B|<brightness_level>"**
- Message from board to PC: **"BLUE LED at brightness: <brightness_level>"**
- Action to be taken: **BLUE LED should glow with brightness: <brightness_level>**

For example, if Message from PC to board: **"B|255"**, then Message from board to PC: **"BLUE LED at brightness: 255"** and Action to be taken: BLUE LED should glow with brightness: 255.

# 4 Demo and Submissions

- You have been issued a take-away kit to work on the experiment.
- You have to get your output verified by your TA on the lab day.
- There will be a folder **Lab-4** already created on that repo. Add Project folders of all the above **two** Problem Statements in that folder of Github repository shared with you.
- Name each folder as **Problem_Statement-1** and **Problem_Statement-2**.
- Deadline for completing **Lab 4** is **Wednesday**, **19th February 2020** by **5 PM**.
- Upload a well documented code with comments of the experiment every week after you have completed the experiment.