***Efficient Embedded Course***

**LAB 8**

**SERIAL COMMUNICATIONS LAB EXERCISE:**

**PERFORMANCE ANALYSIS**

**Issue 1.0**

Contents

[1 Introduction 1](#_Toc87592065)

[1.1 Lab overview 1](#_Toc87592066)

[2 Learning Objectives 1](#_Toc87592067)

[3 Requirements 1](#_Toc87592068)

[4 Details 2](#_Toc87592069)

[4.1 Hardware 2](#_Toc87592070)

[4.1.1 Connections 2](#_Toc87592071)

[4.2 Software 2](#_Toc87592072)

[5 Procedure 2](#_Toc87592073)

# Introduction

## Lab overview

In this lab, you will use an oscilloscope or logic analyzer to see how serial communcations operate.

# Learning Objectives

* Use a logic analyser or oscilloscope to measure the response timing delay in serial communication.

# Requirements

In this lab, we will be using the following hardware and software:

* **Keil µVision5 MDK IDE**
  + Please see the included Getting Started with Keil guide on how to download and install Keil.
* **STM32 Nucleo-L552ZE-Q**
  + For more information, click [here](https://www.st.com/en/evaluation-tools/nucleo-l552ze-q.html).
* **Logic Analyzer or Oscilloscope**

# Details

## Hardware

For this lab you will need a USB to serial adapter, with the functionality shown in Figure 1 (e.g. <http://www.pololu.com/catalog/product/391>). It would also be useful to have an oscilloscope or logic analyzer.

PC

USB to UART bridge

MCU

D+ / D-

TX

RX

RX

TX

USB

UART

Figure . UART to USB to serial adapter

### Connections

Table . Serial signals and connections

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Signal Name | Description | Direction | USB to Serial adapter | | MCU | |
| UART\_RX | Data from PC to MCU | Input to MCU | TX | PD\_9 | |
| UART\_TX | Data from MCU to PC | Output from MCU | RX | PD\_8 | |
| VSS | Ground |  | GND |  | |

Connect the switch signal to the GPIO port input on the MCU as shown in table below. Connect the debug signals and the switch signal to a logic analyzer or oscilloscope. This matches the pins used in the supplied code.

## Software

# Procedure

Load the supplied lab code onto the Nucleo-L552ZE-Q board. Connect a USB to serial adapter. Load a terminal emulation program (e.g. Termite) onto your PC and configure it to support communications (at 57,600 Baud, 8 data bits, no parity, 1 stop bit) through the USB to serial adapter. Verify the system operates by evaluating the square roots of various numbers.

1. Connect an oscilloscope or logic analyzer to the transmit and receive data lines. What is the minimum pulse duration on each line? Is this what you expect? How is it related mathematically to the transmission rate of 57,600 Baud?
2. How long does it take to transmit a five-digit number from the PC, including the carriage return at the end
3. How long is the processing delay between when the end of when last character is received from the PC and the start of when the first character is transmitted back?
4. Modify the code to communicate at a higher Baud rate (e.g. 115,200 Baud, 234,000 Baud). What is the highest rate at which your system can communicate successfully?