***Rapid Embedded Systems***

***Design and Programming Course***

**LAB 5**

**Serial Communication**

**Issue 1.0**

Contents

[1 Introduction 1](#_Toc35809041)

[1.1 Lab overview 1](#_Toc35809042)

[2 Requirements 1](#_Toc35809043)

[2.1 Software and Hardware 1](#_Toc35809044)

[3 Hardware Setup 2](#_Toc35809045)

[3.1 Pin Layout 2](#_Toc35809046)

[4 UART 3](#_Toc35809047)

[4.1 Serial Interface on Mbed Studio 3](#_Toc35809048)

[4.2 Your Application Code 3](#_Toc35809049)

[5 Integration using high-level API 4](#_Toc35809050)

[6 Additional references 4](#_Toc35809051)

# Introduction

## 1.1 Lab overview

In this lab, we will send a message using UART and display the temperature and humidity on the LCD using the Mbed API.

# Requirements

## 2.1 Software and Hardware

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

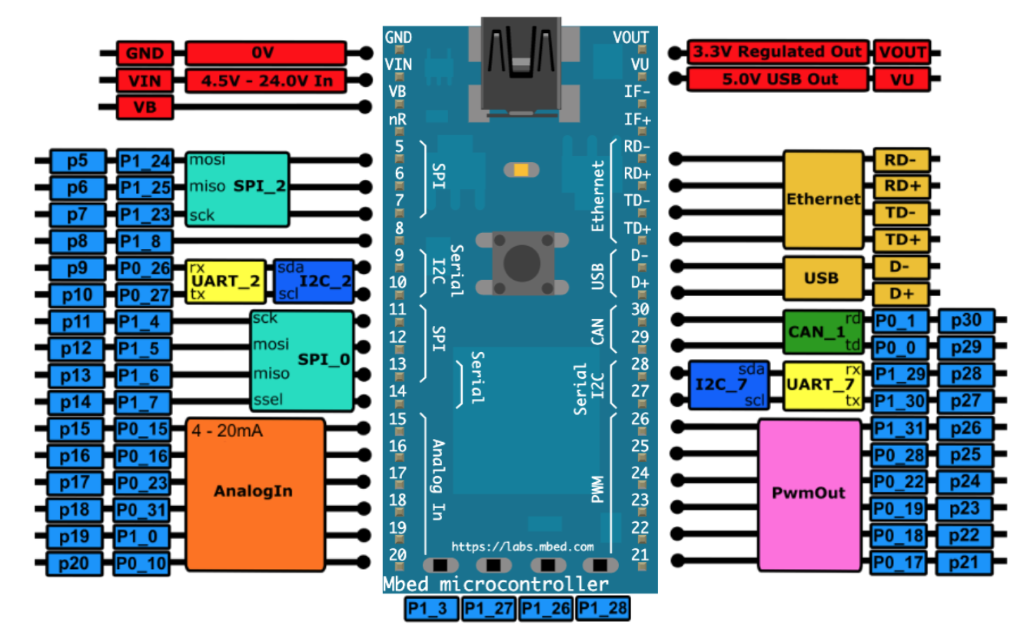
* **Visual Studio Code,**or any other text editor.
* **Mbed Simulator**, an offline simulator of an mbed microcontroller and hardware components.

The code skeletons, which includes some support for implementing the task in 5.2 and 7 should be found in the same folder as this manual

# Hardware Setup

## 3.1 Pin Layout

In this lab, we are going to use the Mbed Simulator target, for which pin descriptions can be found below:



*Figure 1: The Mbed Simulator board pin descriptions*

In the Simconfig.json file, connect LCD **C12832** and the temperature / humidity sensor **sht31** to their respective pin as defined in the table below.

Pin on the board as defined in the table below.

|  |  |
| --- | --- |
| Pin | Pin name in Mbed API |
| C12832 LCD display SPI COTI | p5 |
| C12832 LCD display SPI CITO | p6 |
| C12832 LCD display SPI SCK | p7 |
| Temperature/humidity sensor I2C SCL | p28 |
| Temperature/humidity sensor I2C SDA | p27 |
| USB UART TX | p13 |
| USB UART RX | p14 |

# UART

In this lab task, you will send text from the Mbed simulator to the “PC”.

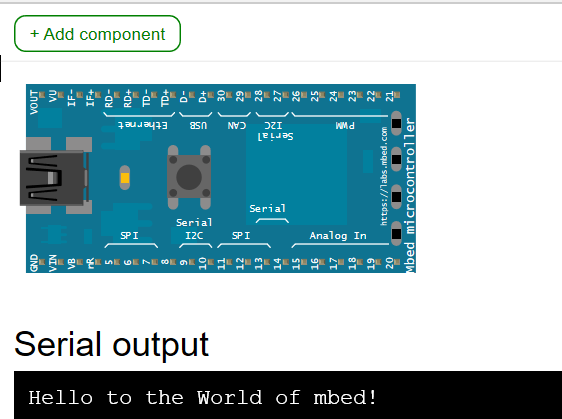
## 4.1 Serial Interface on Mbed Studio

The Serial interface provides UART functionality. The serial link has two unidirectional channels, one for sending and one for receiving. The link is asynchronous, and so both ends of the serial link must be configured to use the same settings.

You can find some the member function of the Serial API in the table below:

|  |  |
| --- | --- |
| Function name | Description |
| Serial (PinName tx, PinName rx, const char \*name=NULL) | Create a Serial port, connected to the specified transmit and receive pins |
| void baud (int baudrate) | Set the baud rate of the serial port |
| void format (int bits=8, Parity parity=SerialBase::None, int stop\_bits=1) | Set the transmission format used by the serial port |
| int readable () | Determine if there is a character available to read |
| int writeable () | Determine if there is space available to write a character |
| void attach (void(\*fptr)(void), IrqType type=RxIrq) | Attach a function to call whenever a serial interrupt is generated |
| void send\_break () | Generate a break condition on the serial line |
| void set\_flow\_control (Flow type, PinName flow1=NC, PinName flow2=NC) | Set the flow control type on the serial port |
| int putc(int ch, FILE \*stream ) | Writes the character ch to stream. Function returns the character written, or EOF if an error happens |
| int getc(FILE \*stream ) | Read a character from the stream, an EOF indicates the end of file is reached |
| int printf(const char \*format, ... ) | Prints output both text string and data, according to format and other arguments passed to printf() |

## 4.2 Your Application Code

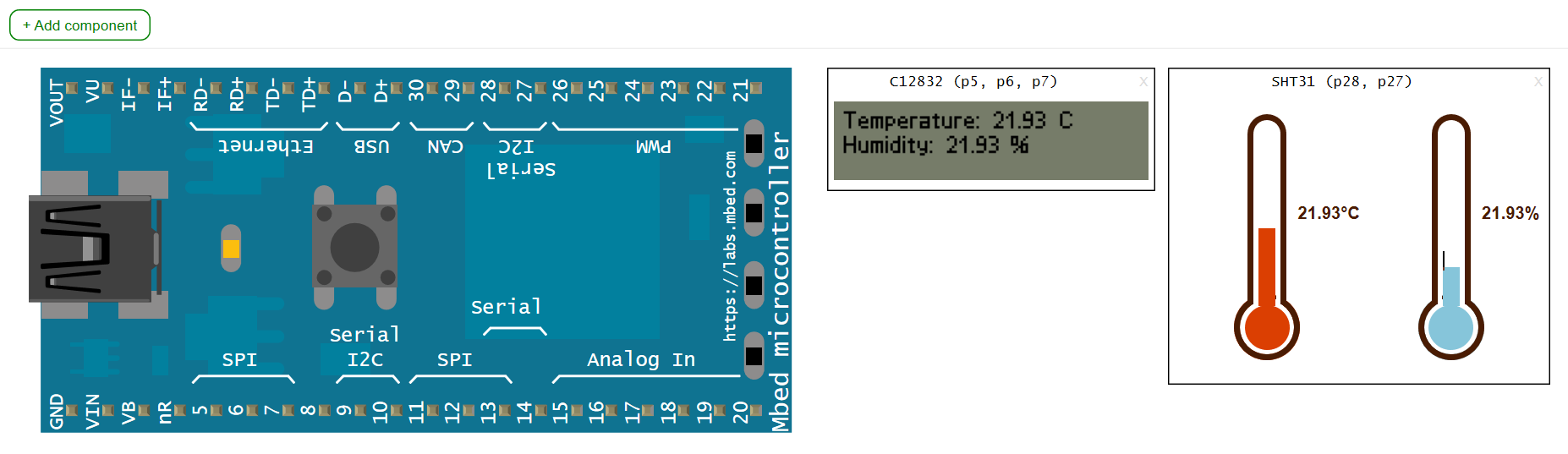
The aim of this task is to send “Hello to the world of mbed!” from the Mbed simulator with a baud rate of 9600 so that the message is printed in the serial output console.

If you get stuck or need help, refer to the solution code provided for this lab.

# Integration using high-level API

In this task, you are required to display the temperature, and humidity on the LCD using high-level APIs, including:

|  |  |
| --- | --- |
| Function name | Description |
| C12832 (PinName coti, PinName sck, PinName reset, PinName a0, PinName ncs) | Create a C12832 LCD interface |
| void cls () | Clear the screen |
| void locate (int x, int y) | setup cursor position |
| int printf(const char \*format, ... ) | Print to the LCD |
| Sht31 (PinName sda, PinName scl) | Create a SH31 temperature sensor interface |
| float readTemperature () | Read the temperature register and convert to the real temperature presentation |
| float readHumidity () | Read the humidity register and convert to the real humidity presentation |



# Additional references

**Documentation of the SPI API**

<https://os.mbed.com/docs/mbed-os/v5.13/apis/spi.html>

**Documentation of the Serial API**

<https://os.mbed.com/docs/mbed-os/v5.13/apis/serial.html>

**More about I2C**

<https://os.mbed.com/docs/mbed-os/v5.13/apis/i2c.html>

**More about shift registers**

<http://en.wikipedia.org/wiki/Shift_register>

**74HC595N shift register datasheet**

<http://www.nxp.com/documents/data_sheet/74HC_HCT595.pdf>

**ST7066U LCD driver datasheet**

[http://www.newhavendisplay.com/app\_notes/ST7066U.pdf](http://www.newhavendisplay.com/app_notes/ST7066U.pdf%20)

**DS1631 temperature sensor datasheet**

<http://datasheets.maximintegrated.com/en/ds/DS1631-DS1731.pdf>

**Download Tera Term VT**

<https://tera-term.en.lo4d.com/windows>