

# Lab 1: Hello World

Taylor Okel

9/11/2018

## Contents

Problem Statement.....	1
Assumptions.....	1
Equipment.....	1
Architecture .....	1
Design.....	1
Implementation .....	2
Test Plan.....	2
Test Results .....	2
Suggestions .....	3
Future improvements of this implementation .....	3
Future improvements of Lab 1 description .....	3
References .....	A

## Problem Statement

Assess the functionality of the Olimexino board, install an IDE, and get a simple “Hello World” program running on the Olimexino board

## Assumptions

- Any IDE can be used
- The board is ready to be flashed – e.g. it has a bootloader already flashed on, and all peripherals and connections work as advertised

## Equipment

- Olimexino Computer Board
- USB Cable
- Desktop Computer running Windows 10 (Workstation)
- Arduino Studio 1.8.6 (IDE)

## Architecture

The architecture is quite simple; the Olimexino is connected to the Workstation via the provided USB cable. No further hardware was used.

## Design

To print Hello World, a serial connection must be made. Arduino provides a Serial library (Arduino Serial Reference, 2018); using this, at a baud rate of 9600, a serial connection is simple to setup. Since there will be a delay until the Serial port becomes active, use the loop function to print “Hello World” continuously. Figure 1 demonstrates the logic behind this design.

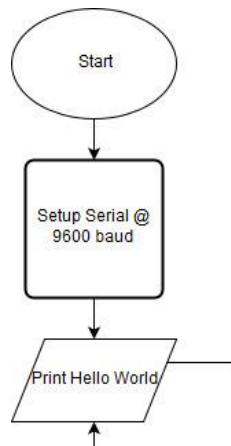


Figure 1: Lab 1 Flowchart

The following code implements this design:

```
/*
 * Lab 1 - Hello World
 * Taylor Okel
 * 9/9/2018
 * Opens a serial connection @rate bps and continuously
 * prints Hello, World! to the connected console.
 */
int rate = 9600;
void setup() {
  Serial.begin(rate);      // open the serial port at 9600 bps:
  while(!Serial){ ; } // Wait for serial to connect
}
void loop() {
  Serial.println("Hello, World!");
}
```

## Implementation

To implement this design, power the board with the USB cable and the workstation. Once powered, bring up the Arduino IDE. Configure the Arduino IDE as in Table 1, using the Tools Menu in the IDE:

Table 1: Arduino IDE configurations

Board	Maple (Rev 3)
CPU Speed	72 MHz (Normal)
Optimize	Debug (-g)
Port	<depends on workstation setup>

*Note that, for this application, the optimization does not matter; there is no need to save space, nor are there any timing constraints. The debug option may provide the Tester with more information in this application. Future applications should choose between smaller or faster optimizations.*

Use the Checkmark and Arrow buttons on the IDE to compile and upload the code, respectively.

## Test Plan

1. Connect the board as discussed in Implementation
2. Open the IDE
3. Open the Serial Monitor
4. Upload the sketch
5. Confirm that continuous “Hello, World!” messages appear on-screen after a short delay

## Test Results

The design worked as expected. The serial monitor output appeared as captured in Figure 2.

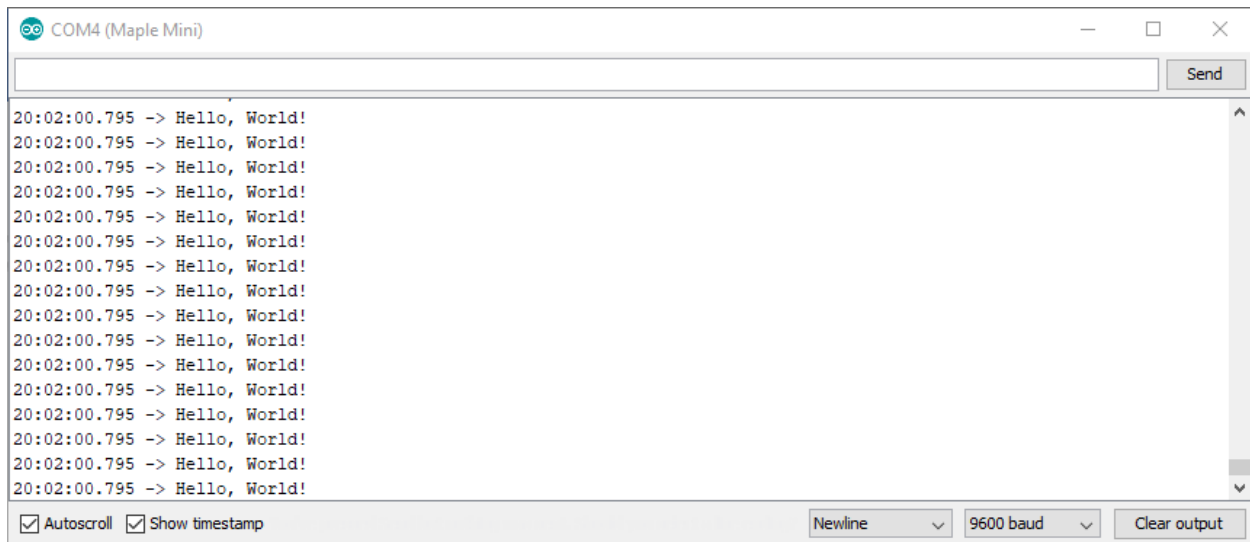


Figure 2: Serial Monitor Output

## Suggestions

### Future improvements of this implementation

The application is simple, the only future improvement that could be made is to print the “Hello, World!” message exactly once, after the serial connection is made.

### Future improvements of Lab 1 description

Several students, myself included, found themselves having some difficulty getting the board to work this lab. Of course, this was the exercise in this lab. However, it might be beneficial for future classes to have a list of proven steps, from a clean workstation, to get these boards running. Use of any other IDEs would therefore be encouraged, but not necessarily supported.

## References

*Arduino Serial Reference*. (2018, September 11). Retrieved from Arduino.cc:  
<https://www.arduino.cc/reference/en/language/functions/communication/serial/>