Reykjavík University

Mechatronics I

T-411-MECH

Individual Assignment 5

14/09/2020

Students:
Giacomo Menchi

Teacher:
Joseph T. Foley

Contents

1 Tasks

As usual, the code for this assignment is available at the following Github repo:

`https://shorturl.at/uEUY2`

1.1 Task 1

Task: Get started with Arduino and Arduino IDE. Open blink example and put your name in the comment.

Screenshot: This was quite the easy task, so just check the screenshot below.

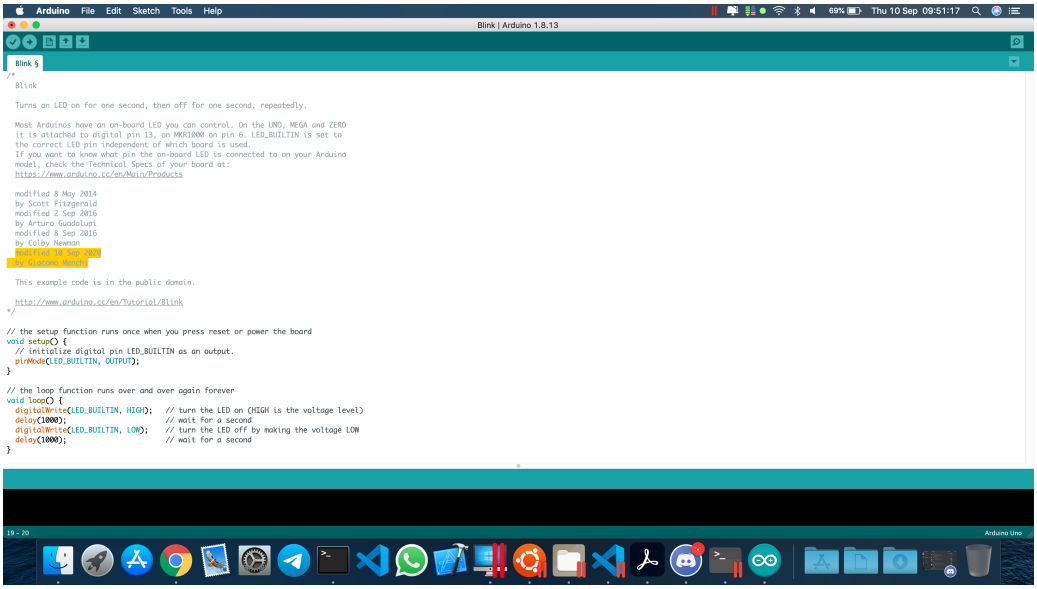

Figure 1: Blink example with my name on the comment.

1.2 Task 2

Task: Set Arduino Nano as the target board and set the com port to where the nano is. Blink a led on it with two different rates.

Video: Again, to complete this task I only needed to adjust the aforementioned settings and the Processor to ATmega328P (Old bootloader), and then edit some lines from the Blink sketch.

`https://shorturl.at/hCFT6`

1.3 Task 3

Task: Make echo code for I2C connection between the Arduino and the RPi so the Arduino is the slave device and the RPi is the Master.

NOTE; use Wire library for Arduino

Screenshot: As shown in the screenshots below, I just programmed the Raspberry Pi to send a "1" followed by a "0" to the Arduino in a looping (neverending) way, and the Arduino to receive those messages and print them on its serial console.

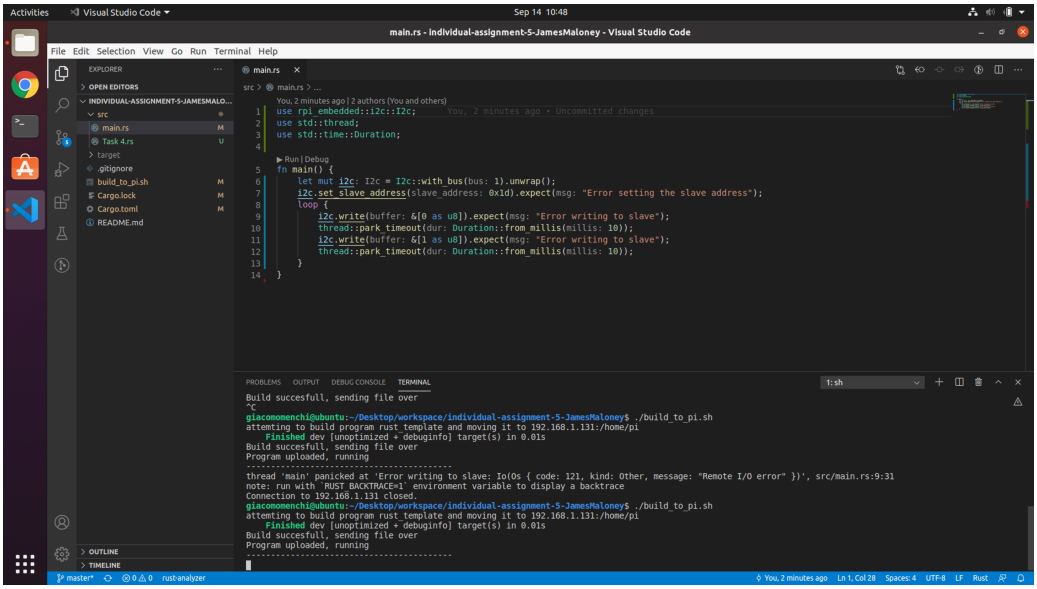

Figure 2: Echo code for Raspberry Pi in Rust.

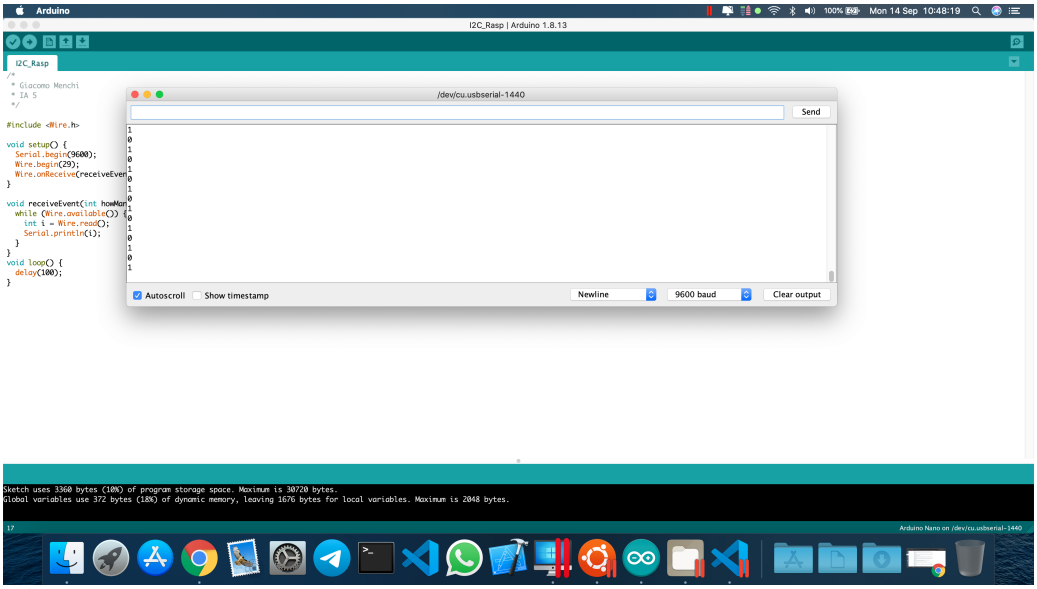

Figure 3: Echo code for Arduino in C.

1.4 Task 4

Task: Create code to receive and send any keyboard typed data to and from the I2C port on the Rpi in Rust.

Screenshot: This is the proof demonstrating the echo server code in Arduino (and Raspberry Pi). Whenever a string is sent to the Arduino, it's received, saved, and also sent back to the Raspberry Pi itself.

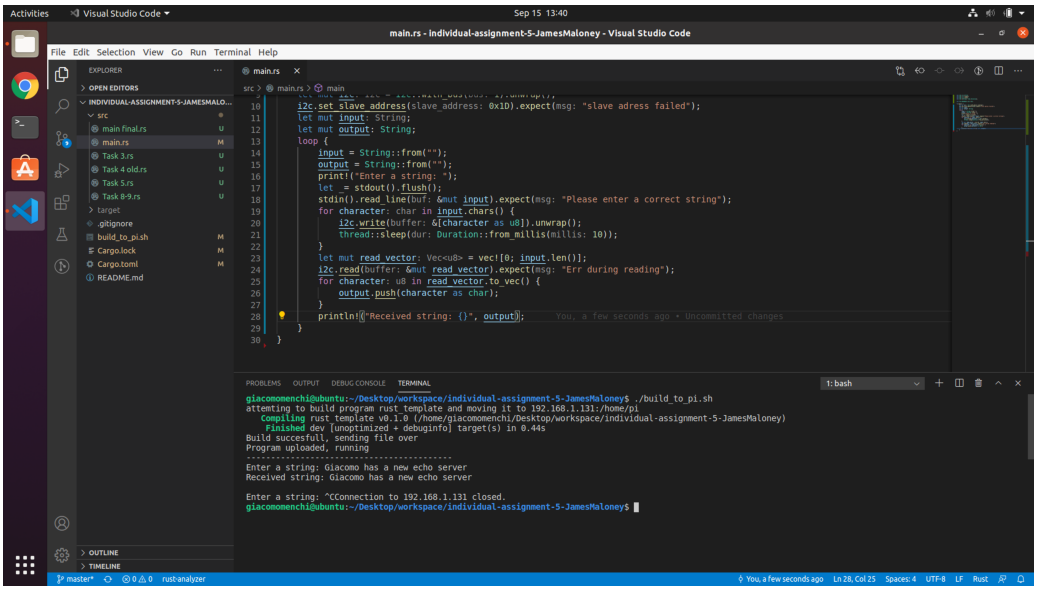

Figure 4: Echo code for Raspberry Pi in Rust.

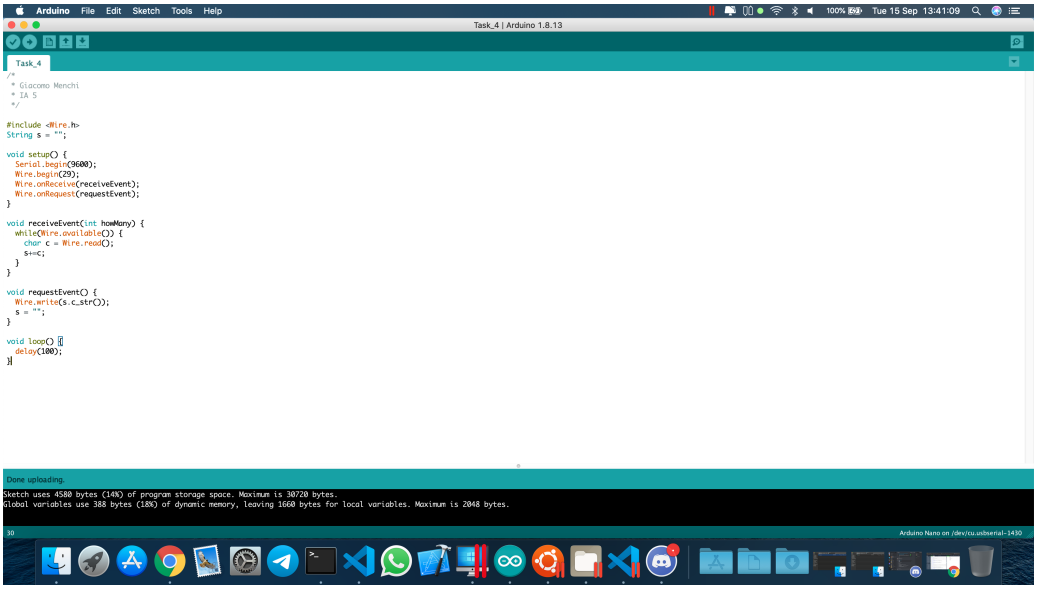

Figure 5: Echo code for Arduino in C.

1.5 Task 5

Task: Make the Arduino and the RPi communicate by sending numbers back and forth where the Rpi is adding 10 and the Arduino is adding 1 when they receive data and then sending it back. Print the incoming number on each device terminal.

Screenshot: Again, the screenshots below represent the implementation of the task 5 on both sides, the Raspberry Pi and the Arduino. The Raspberry starts by sending 0, which Arduino receives, prints and sends back adding 1, which is again sent back from Raspberry Pi as 11, and so on.

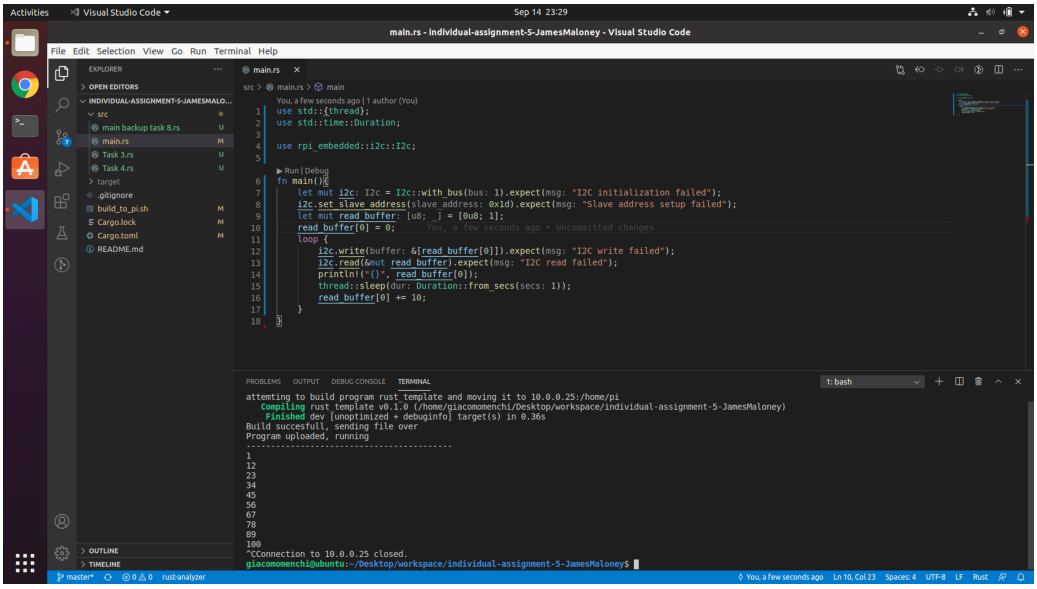

Figure 6: Number exchange code for Raspberry Pi in Rust.

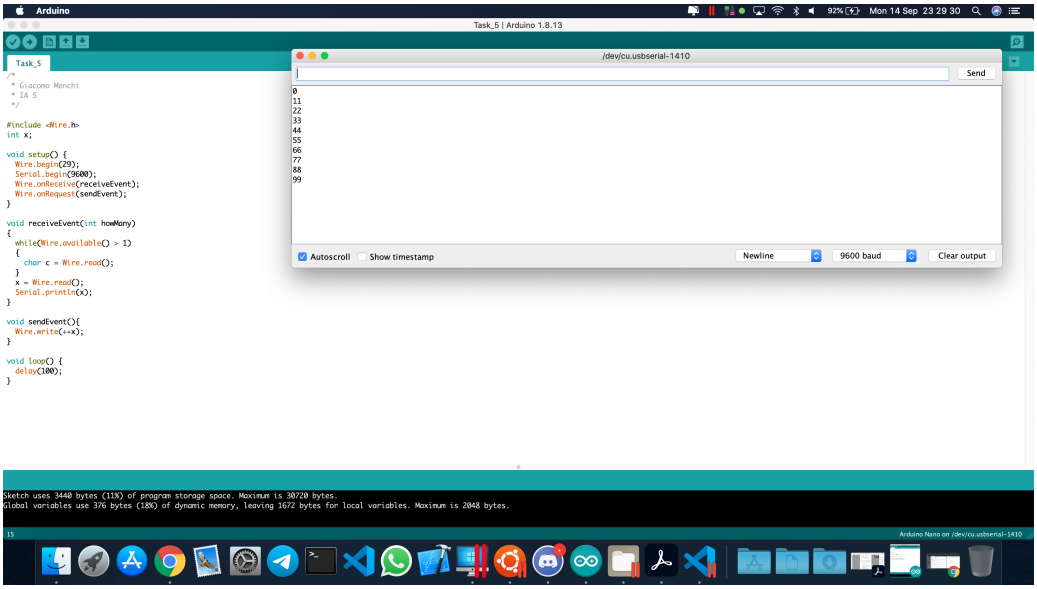

Figure 7: Number exchange code for Arduino in C.

1.6 Task 6

Task: Set Serial communication between the RPi and Arduino using a logic level converter (LLC).

NOTE; check the book page 347 for UART and 359 page for LLC

Schematic: The picture below illustrates the task 6 schematic, where the Raspberry Pi is communicating with the Arduino via logic level converter and the Raspberry Pi is also powering the Arduino itself.

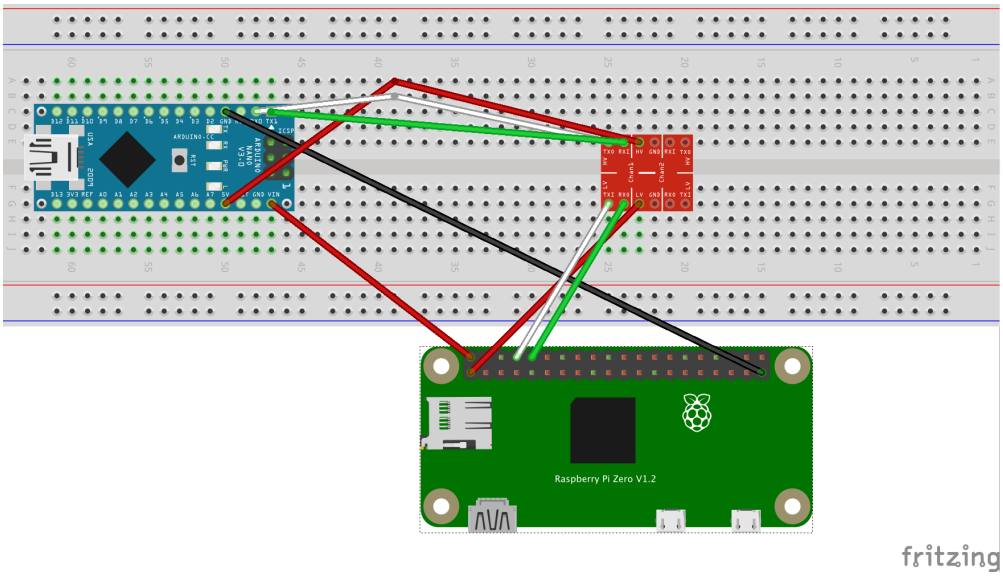

Figure 8: Raspberry Pi and Arduino Serial communication schematic.

1.7 Task 7

Task: Write a code for Serial connection on the Arduino side to send data to RPi.

Explanation: Said code can be found in the Github repo, inside the file named "Task_7_8.ino". The screenshot of it working can also be found below in task 8 (the two tasks were very similar, so I merged their screenshots).

1.8 Task 8

Task: Create some code in Rust for RPi to get data from Arduino over Serial. Send your name and kennitala to your Raspberry Pi.

Screenshot: As always, the screenshots below illustrate both the Raspberry Pi and Arduino side of the task 8.

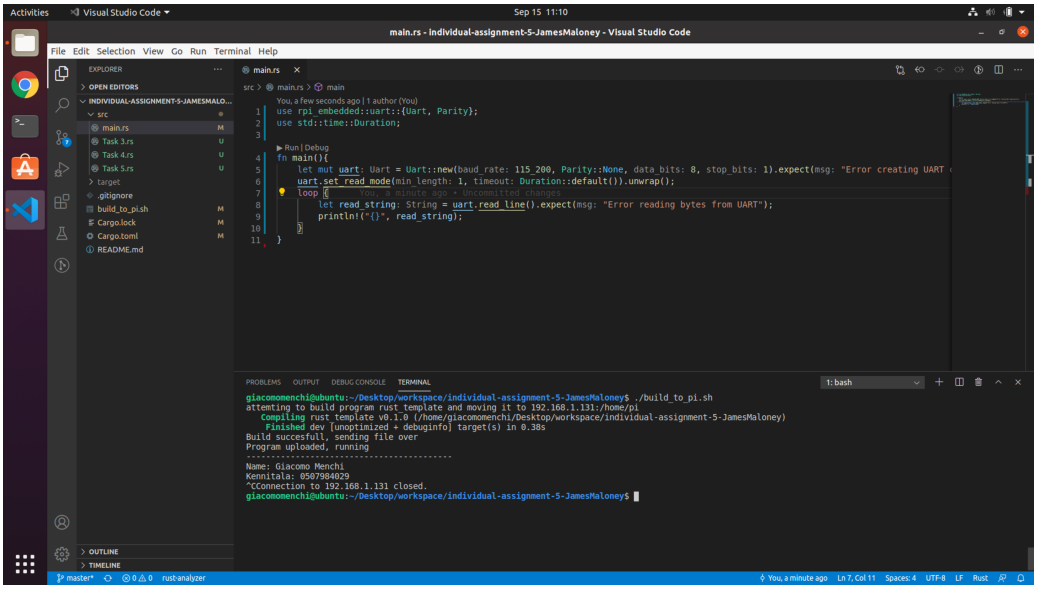

Figure 9: Name and Kennitala exchange code for Raspberry Pi in Rust.

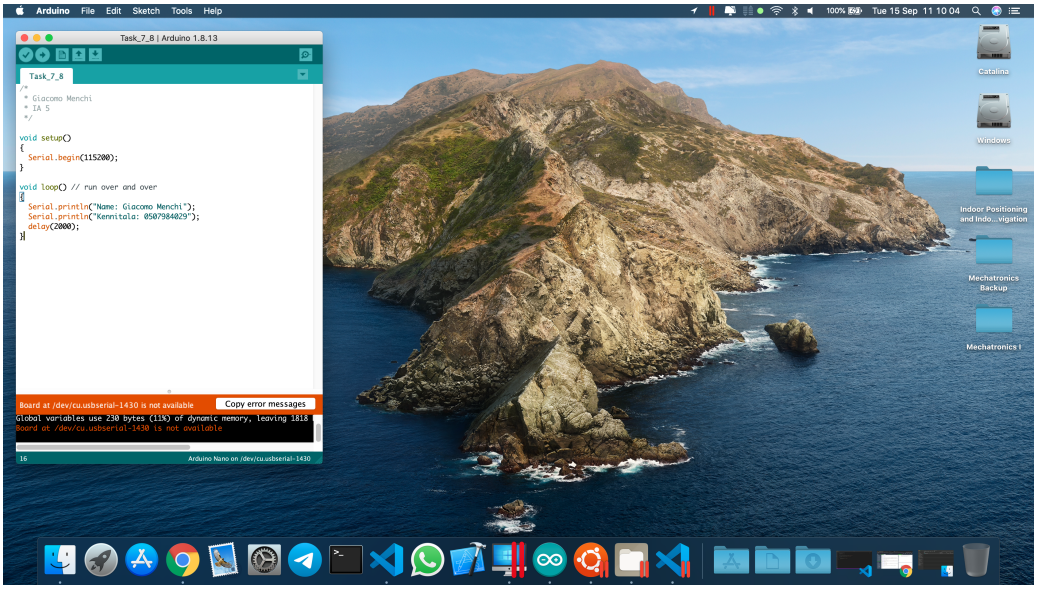

Figure 10: Name and Kennitala exchange code for Arduino in C.

1.9 Task 9

Task: Make Arduino send a string of numbers and RPi print the string. Let the new numbers increase with a new iteration.

Screenshot: No explanation required, just a numbers string sent from Arduino, printed by Raspberry Pi and sent again increased by one by Arduino each time.

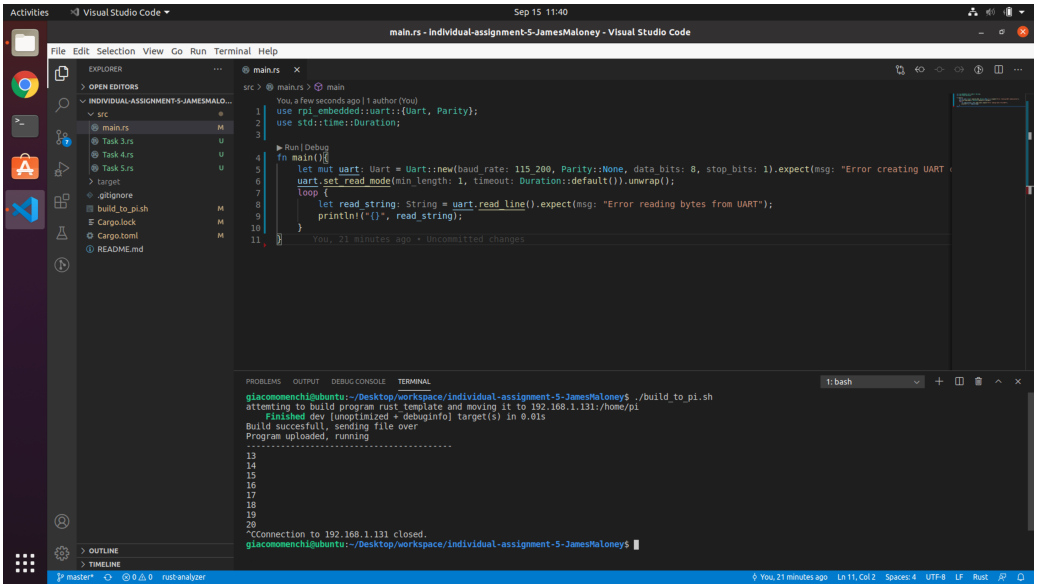

Figure 11: Increasing numbers string exchange code for Raspberry Pi in Rust.

1.10 Task 10

Task: Extract numbers from the string to integers and print them separately.

Screenshot: In this last task, I modified the previous code to make it print the single digits separately, as integers.

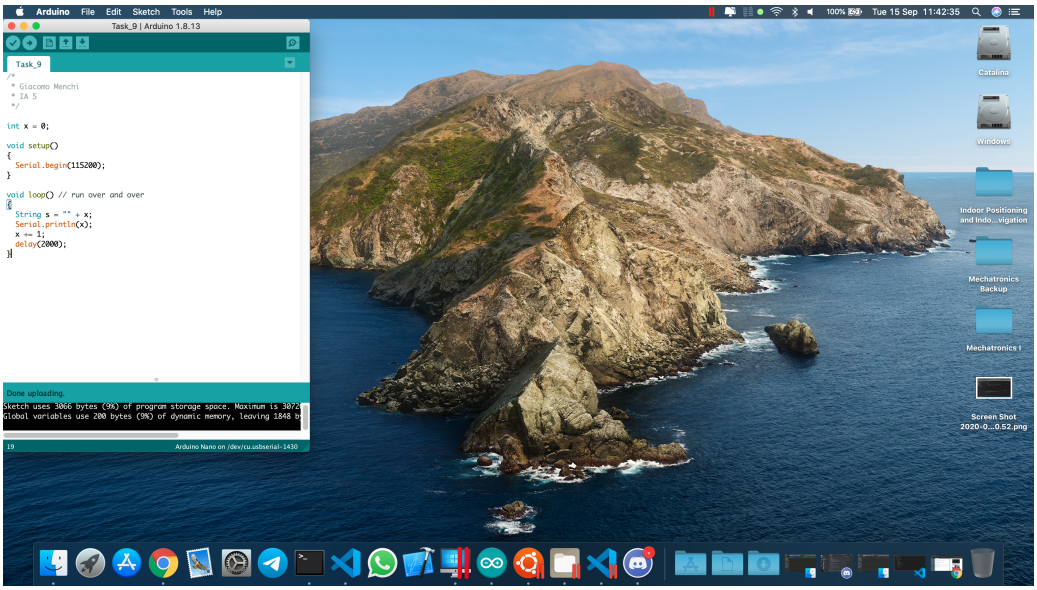

Figure 12: Increasing numbers string exchange code for Arduino in C.

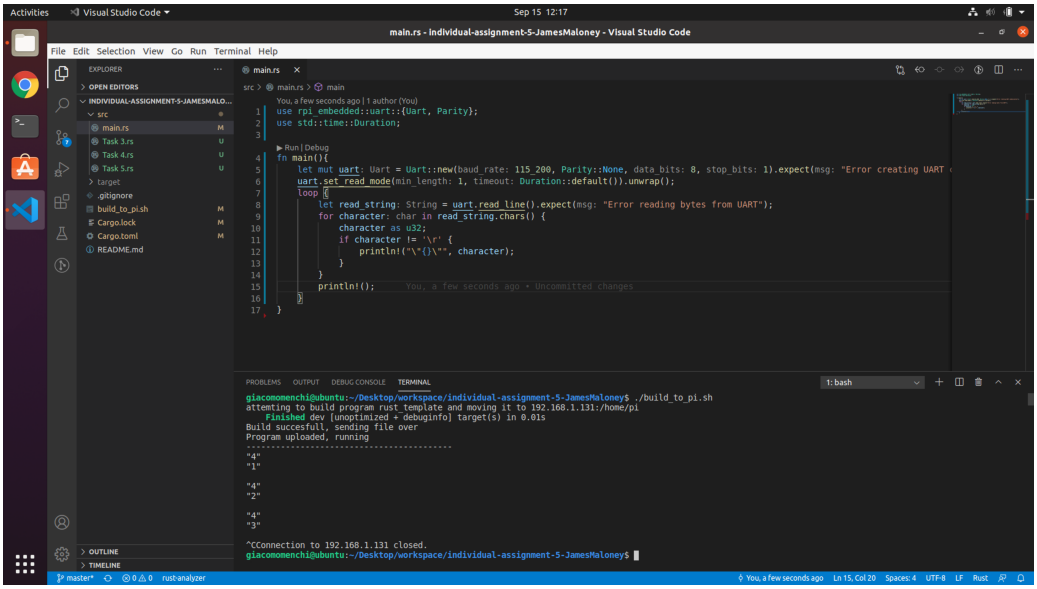

Figure 13: Printing numbers strings as single integers.