Raspberry Pi and Arduino - Intercom Project

Time to start the final project of this course!

In this PDF, you will find the specs you need to start, as well as an overview of the steps I'm going to take for the solution videos.

For the specs, instead of an exhaustive technical list for Arduino and Raspberry Pi, I have made a recap following the user interaction. This should give you enough data to start. For the data you don't have, well, you wouldn't think that would be so easy, right? Try to come up with your own solutions for what I don't describe here. This is how things work in real life; you will never have all the variables of a particular problem when you start.

And then, remember that the design + code I propose is ONLY one possible solution. You might end up with a totally different code, but still working. And that's great too! The real goal here is to make what you saw on the demo video.

Note: At any time, feel free to come back to previous lessons on the course. Solving problems when you work on a project is one of the best ways to learn, because the lectures you will watch again will be directly linked to a practical problem.

Project Result - User Interaction

Make sure you have watched the introduction video where I show you a running demo of the project.

Here is a recap of the result.
The app will start with the
following:

- LED turned blue

- Message on the LCD screen: "Press button to call"

When you press on the push button, you request to open the door. The Arduino will send the request to the Raspberry Pi.

At this moment:

- Message on the LCD screen changes to "Please wait a few seconds"

- The Pi takes a photo with the camera

- The Pi sends a message + the photo to a Telegram chat, using the Telegram bot

When you receive the message + photo on Telegram, you have two options: you can choose to open the door or deny access.

If you choose to open the door, here is what happens:

- First:

    - Servo goes to the open position

    - LED turns green

    - Buzzer plays a sound

    - "Door is opened" message on the LCD screen

- After 10 seconds, the door will automatically be closed:

    - Servo goes back to the close position

    - LED turns blue again

    - Buzzer plays a sound

    - "Press button to call" message on the LCD screen

If you choose to deny the access:

- First:

    - LED turns red

    - Buzzer plays a sound

    - "Access denied" message on the LCD screen

- After five seconds:

    - LED turns blue again

    - "Press button to call" message on the LCD screen

At the end of the "open door" and "deny access" processes, the app comes back to the initial state and is ready to receive new requests when you press the button.

How to Start the Project

1. Make sure to watch the introduction video showing you the result you will have at the end.

2. Read the preceding description to get more details.

3. Try to define what functionalities you need to implement, and also:

   a. Which ones on the Arduino side

   b. Which ones on the Raspberry Pi side

4. Once you have the functionalities, define the steps you need to take to finish the project.

5. Read the project step orders to compare and/or get an idea of how to start.

6. Start working on the project! Iterate on the code, test, iterate, test, iterate, and so on.

7. While doing the project, keep this document on the side and don't hesitate to come back to previous course lessons for a refresher. If you are stuck, watch the next step solution.

Project Steps

Here's a quick overview of what I'm going to do in each step of the solution. If you are completely stuck, you can always watch the next solution step, and then work by yourself again to complete the next step.

- **Step 0-A:** as written before, try to design the steps by yourself! This is an additional practice for you, on how to design a project and break it down into smaller units of work.

- **Step 0-B:** Make sure that all the components on the circuit are correctly working, as well as the Pi camera and the Telegram bot.

- **Step 1 (Protocol):** Define the communication protocol (what commands/messages to send/receive) between the Arduino and the Raspberry Pi, and also between the Telegram chat and the Raspberry Pi. In this step, we don't write any code!

- **Step 2 (Arduino):** Init all the hardware components. Make the servo go to the close position and print a "Starting…" message on the LCD for one second, then clear the screen.

- **Step 3 (Arduino):** Detect when the push button has just been pressed, with a debounce mechanism. When the button is pressed, send a message to Serial.

- **Step 4 (Arduino):** Receive commands from Serial and process them for all the components: LED, buzzer, LCD screen, servo. For the servo, we don't receive an angle, but simply "open_door" and "close_door" commands, which we then translate into an angle.

- **Step 5 (Rpi):** Init Serial communication on a Python program to connect to the Arduino.

  Receive the button_pressed message, which corresponds to an "open door" request. When receiving the message, send a command back to change the text on the LCD.

- **Step 6 (Rpi):** Init the Camera. Take a photo when a request is received.

- **Step 7 (Rpi):** Init Telegram bot. After taking the photo, send a message and the photo to a Telegram chat. Also, make sure we don't send more than one request every five seconds, and that we don't send a request if one is still in process.

- **Step 8 (Rpi):** Init Telegram Updater. Receive and handle the "open" command. In the Telegram callback, open the door (servo, LED, buzzer, LCD commands). After 10

seconds, close the door (servo, LED, buzzer, LCD commands) and come back to the initial state.

- **Step 9 (Rpi):** Receive and handle the "deny" command. In the Telegram callback, send a deny message (LED, buzzer, LCD commands). After five seconds, come back to the initial state.

- **Step 10 (systemd):** Bonus step to make the Python program start on boot, so you don't have to manually launch it every time.

Get Started

All right, that's enough tips for now! Time for you to start working on this project. Again, if you need any help, feel free to:

- Re-watch lessons from the previous sections

- Search on Google

- Just take a break and come back later; this is often a good thing to do

- And if completely stuck, watch the next solution step

The solution will help you in three ways: first, if you are really stuck in one part, don't waste 10 hours being stuck. After some time, it's best to watch the solution and try to understand it. Second, after you finish the project, you can see how someone else is thinking about the same problem. And third, I will try to give you a solution with clean code and best practices (plus additional tips!), so you can use those tips to make your programs cleaner and more readable in the future.