Garage Door Controller

Build and

Installation Instructions

By: Jim Schrempp and Bob Glicksman; v1, 7/23/2020

**NOTICE:** Use of this document is subject to the terms of use described in the document “Terms\_of\_Use\_License\_and\_Disclaimer” that is included in this release package. This document can be found at:

https://github.com/TeamPracticalProjects/Garage\_Door\_Controller/blob/master/Terms\_of\_Use\_License\_and\_Disclaimer.pdf

A wooden table

Description automatically generated

© 2020 Team Practical Projects, Bob Glicksman and Jim Schrempp. All rights reserved.

# 

# 

# 

# TABLE OF CONTENTS.

[TABLE OF CONTENTS. 1](#_Toc46340394)

[1. OVERVIEW. 2](#_Toc46340395)

[2. WIRELESS I/O BOARD ASSEMBLY INSTRUCTIONS. 5](#_Toc46340396)

[3. GARAGE DOOR REMOTE MODIFICATION INSTRUCTIONS. 7](#_Toc46340397)

[4. ENCLOSURE FABRICATION AND INSTALLATION INSTRUCTIONS. 12](#_Toc46340398)

[5. FIRMWARE INSTALLATION AND TESTING INSTRUCTIONS. 23](#_Toc46340399)

[6. ANDROID APP INSTALLATION INSTRUCTIONS. 29](#_Toc46340403)

# OVERVIEW.

This document provides step-by-step instructions for assembling and installing a Garage Door Controller. Before proceeding with this project, you must read and accept the “*Terms\_of\_Use\_License\_and\_Disclaimer*” document which is located in the root level of this repository.

The following parts are needed to build this project:

* Wireless I/O Board with components needed for the project. See section 2 of this document for details.
* (Optional) External WiFi Antenna for Photon. You might want to use an external antenna if your WiFi signal is weak in the garage where the project will be mounted. A candidate device is: <https://www.digikey.com/product-detail/en/pulselarsen-antennas/W3918B0100/1837-1007-ND/7667481>
* Garage Door Opener Remote Control Unit. Our garage door opener uses remote control unit “Liftmaster 371LM”, available at: <https://www.amazon.com/gp/product/B075MQCH2P/ref=ppx_yo_dt_b_asin_title_o02_s00?ie=UTF8&psc=1>. Your garage door opener may require a different remote control unit. The remote control unit needs to be modified and mounted in the project enclosure. See sections 3 and 4 of this document for details.
* Ultrasonic Distance Sensor, SR-HC04. These are very commonly found, for example: <https://www.amazon.com/ELEGOO-HC-SR04-Ultrasonic-Distance-MEGA2560/dp/B01COSN7O6/ref=pd_bxgy_3/137-6812788-8746123?_encoding=UTF8&pd_rd_i=B01COSN7O6&pd_rd_r=cd21824f-1050-44c8-8e2f-b261e16cf05a&pd_rd_w=oLoDZ&pd_rd_wg=nKuOW&pf_rd_p=ce6c479b-ef53-49a6-845b-bbbf35c28dd3&pf_rd_r=F9BXN7BAX9TP8ED9SY81&psc=1&refRID=F9BXN7BAX9TP8ED9SY81>
* Mounting bracket for HC-SR04 sensor. If your sensor did not come with a suitable mounting bracket, you can purchase these: <https://www.amazon.com/HC-SR04-Cartoon-Ultrasonic-Distance-Mounting/dp/B01FDGU0GY/ref=sxts_sxwds-bia-wc-nc-drs1_0?crid=1HS1W1659L4XW&cv_ct_cx=hc-sr04&dchild=1&keywords=hc-sr04&pd_rd_i=B01FDGU0GY&pd_rd_r=d8101316-87b4-4755-9f9f-a978dcfc9433&pd_rd_w=zuloG&pd_rd_wg=ulQPE&pf_rd_p=43f4b3f0-0b04-46ba-8a08-2e851d035e17&pf_rd_r=2DJESSE197W2F6GX3W0G&psc=1&qid=1595356387&sprefix=HC-SR04%2Caps%2C215&sr=1-1-f3947b35-9c59-4d7a-9603-b751e6eed25b>
* USB “Wall Wart” Power supply. A 5 volt, 1 amp (min) power supply. For example: <https://www.adafruit.com/product/501>
* USB A/B Cable. The length will depend upon where the project is mounted with respect to the nearest convenient AC power source. Here is a 10 foot cable that we used: <https://www.amazon.com/gp/product/B00NH13DV2/ref=ppx_yo_dt_b_asin_title_o02_s00?ie=UTF8&psc=1>
* Project enclosure. We used the following “pencil box”: <https://www.amazon.com/Really-Useful-Plastic-Storage-Liter/dp/B003H790JU/ref=sr_1_2?dchild=1&keywords=really+useful+boxes+pencil+box&qid=1595357131&sr=8-2>
* Female-Female Jumper Cables. If these didn’t come with the HC-SR04 ultrasonic sensors, you can purchase these: <https://www.adafruit.com/product/1950>
* Mounting Hardware.
  + 2 ea. ½ inch #4-40 nylon standoffs, female-female, threaded.
  + 4 ea. ¼ inch #4-40 nylon screws.
  + 4 ea. ½ inch #4-40 nylon screws.
  + 4 ea. #4-40 nylon nut
  + 6 ea. 1.25 inch, #6 wood screw.

The following tools and materials are needed to build this project. *Make sure that you know how to use these before undertaking this project*!

* Small tip soldering iron.
* Electrical solder.
* A few feet of #26 solid, insulated wire. It will be helpful to use three different colors – red, black, any other color.
* Wire stripper.
* Diagonal wire cutter.
* Needle nose pliers.
* Various screwdrivers.
* Electric drill, ¼” or larger, with a set of bits 1/32” to ¼” minimum
* Electrical tape.
* Hot glue gun with glue sticks.
* Nibbling tool (optional)
* Set of small files (optional).

Our recommended order of assembly is as follows.

* Assemble a *Wireless I/O Board* with the necessary components for this project – see section 2 of this document.
* Modify the remote control unit for use in this project – see section 3 of this document.
* Drill mounting holes in the remote control unit enclosure – see section 4 of this document.
* Drill/cut the project enclosure and HC-SR-4 mounting bracket – see section 4 of this document.
* Assemble parts in project enclosure – see section 4 of this document.
* Wire up all parts of the project enclosure – see section 4 of this document.
* Install Photon firmware and test that the parts work – see section 5 of this document.
* Mount the project in its final location – see section 4 of this document.
* Install the App on your smartphone – see section 6 of this document.

# WIRELESS I/O BOARD ASSEMBLY INSTRUCTIONS.

The *Wireless I/O Board* provides the basic electronics for this project, including the Particle[[1]](#footnote-1) Photon, a relay that is controlled by the Photon, a 3.3 volt power supply for the garage door remote, and supporting circuitry. The *Wireless I/O Board* is a separately documented project of ours and you can find complete details at:

<https://github.com/TeamPracticalProjects/Wireless_IO_Board>

The *Wireless I/O Board* is a general purpose circuit board that provides many different functions and external interfaces. Not all of its functionality is required for this project. You need only assemble the parts that are needed for this project. Specifically:

* 1 ea. Particle Photon
* 2 ea. 12 position female headers (to mount the Photon)
* 1 ea. Type B USB connector.
* 1 ea. 74AHCT125 level converter IC
* 1 ea. 14 pin DIP socket for the 74AHCT125
* 1 ea. 5 volt relay
* 1 ea. 2N2222 transistor
* 2 ea. 1N4004 diode
* 2 ea. Two position terminal block, 3.5 mm
* 1 ea. Three position terminal block, 3.5 mm
* 1 ea. LD1117-3.3v voltage regulator
* 1 ea. 3 position female header for mounting the LD1117
* 7 ea. capacitor, 0.1 uF
* 2 ea. Capacitor, 100 uF
* 6 ea. Resistor, 4.7 Kohms
* 4 ea. Male pin header
* 1 ea. *Wireless I/O Board* printed circuit board

A detailed parts list with ordering information can be found at:

<https://github.com/TeamPracticalProjects/Wireless_IO_Board/blob/master/Hardware/PCB/Wireless_IO_Board_Parts_List.pdf>

Detailed assembly instructions can be found in the document:

<https://github.com/TeamPracticalProjects/Wireless_IO_Board/blob/master/Docs/Wireless_IO_Board_Build_Instructions.pdf>

Specifically, refer to the following sections in the aforementioned document:

* Section 2.1, “Core Components”
* Section 2.3, “Relay/Solenoid Control Parts”
* Section 2.4, “Servo Control Parts”. Note: this project does not use a servo. The 3 pin servo control header provides 5 volt power, ground, and the trigger signal for the HC-SR04 ultrasonic sensor (mounted externally to the *Wireless I/O Board*). In addition, position 2 of the digital I/O terminal bock (section 2.6) does not have a terminal block soldered to it but rather has a male pin header soldered to it that provides *Wireless I/O Board* connection from the HC-SR04 ultrasonic sensor echo pin.

The completed circuit board for this project should look like figure 2-1, below.

A circuit board

Description automatically generated

*Figure 2-1. Wireless I/O Board Assembled for this Project.*

# GARAGE DOOR REMOTE MODIFICATION INSTRUCTIONS.

This section provides detailed instructions for modifying a Liftmaster 371LM garage door opener remote control unit for use in this project. This model remote control is compatible with Liftmaster and Chambertin garage door openers that have a purple learn button. You can purchase a spare remote at:

<https://www.amazon.com/gp/product/B075MQCH2P/ref=ppx_yo_dt_b_asin_image_o02_s00?ie=UTF8&psc=1>

If your garage door uses a different model remote, the instructions in this section should be useful in guiding your model-specific modifications.

A garage door opener remote unit is needed for this project, as it is the means by which the garage door opener is actually activated. The remote is wireless so no connections to the opener itself are needed. The basic idea is to open up the remote unit and find the pushbutton switch that activates the circuitry and then to solder wires to the switch so that the *Wireless I/O Board* relay contacts operate in parallel with the remote unit’s switch. Additionally, if the remote unit uses a 3 volt battery, the battery may be eliminated and the *Wireless I/O Board* 3.3 volt motor power supply may be used in its place. Eliminating the battery eliminates the need to periodically replace it. Replacing the battery may be inconvenient if the project is mounted up in the garage’s rafters.

Figure 3-1 shows the remote control unit. The unit comes with a visor clip which is not needed and may be discarded.

A picture containing indoor, sitting, keyboard

Description automatically generated

*Figure 3-1. Remote Control Unit.*

Open up the remote unit by inserting a small, flat screwdriver in the notch at the side of the unit, see figure 3-2.

*A picture containing indoor, sitting, table, desk

Description automatically generated*

*Figure 3-2. Pry open slot in the remote unit.*

The plastic cover pries off by twisting the screwdriver and then moving the screwdriver around the periphery of the unit, twisting it open as you go. The opened up unit looks like figure 3-3.

A close up of a box

Description automatically generated

*Figure 3-3. Remote Unit Opened Up.*

Note the battery holder in figure 3-3. If your unit comes with a battery, remove the battery by prying up the top lip that holds the battery in place while pushing the battery out of the holder. The result is shown in figure 3-3.

The modifications that need to be made are depicted in figure 3-4.

A circuit board

Description automatically generated

*Figure 3-4. Modification Points on the Remote Unit.*

In this particular unit, we found that the point labeled “Pushbutton active contact” is open when the switch is not depressed and is shorted to ground when the switch is depressed. The sizable copper plane in figure 3-4 is the ground plane.

The pushbutton switch has 4 contacts. The two contacts across the top are shorted together inside the switch, as are the two contacts on the bottom. Depressing the switch shorts the top contacts to the bottom contacts.

We found it easier to access the contact labeled “Alternative pushbutton active contact” in figure 3-4. We found an assessable connection to the remote unit’s ground plane at the point labeled “Circuit ground (battery -)” in figure 3-4. Any place on the top of the battery holder is the battery + terminal.

In order to modify the remote unit, solder a #26 awg solid copper wire to each of the following points, as shown in figure 3-5:

* Battery + (we suggest a red wire for this)
* Circuit ground (battery -) (we suggest a black wire for this)
* Alternative pushbutton active contact (we suggest any other color wire for this)

The result should look like figure 3-5.

A circuit board

Description automatically generated

*Figure 3-5. Modifications to the Remote Unit.*

We suggest that each wire be at least 6 inches long at this time. You will trim the wire for mounting in the project enclosure later.

Lastly, we strongly suggest that you bundle these wires somewhere near where they exit the remote unit and provide a strain relief by hot gluing them to the remote’s circuit board, see figure 3-6. Make sure to leave a little slack in the wires in the circuit board side so that the wires are not under any tension.

A circuit board

Description automatically generated

*Figure 3-6. Hot Glue Strain Relief.*

# ENCLOSURE FABRICATION AND INSTALLATION INSTRUCTIONS.

At this point in the project’s assembly you should have the following steps accomplished:

* A completed *Wireless I/O Board*, per section 2 of this document.
* A modified remote control unit, per section 3 of this document.

In this step, these components are mounted inside a plastic enclosure. We strongly suggest that you do this because:

* The enclosure will help keep dust and dirt away from the unit.
* The enclosure will hard mount the *Wireless I/O Board* and the remote control unit together so that they can be wired together safely and securely.

Note that we chose NOT to mount the ultrasonic sensor inside the project enclosure. This could be done, but the transducers on the unit need to be in free air. We chose to use an off the shelf mounting bracket that is designed for the ultrasonic unit, see figure 4-1.

**

*Figure 4-1. Ultrasonic Sensor Mounting Bracket.*

We suggest affixing the ultrasonic sensor to the bracket using hot glue. First, however, note that the bracket mounting holes on the angled side of the bracket (left side of the figure) are very small. We suggest drilling the two end holes out so that a #6 wood screw will fit through (an 11/64” drill bit will work fine here). This right-angle side will mount the bracket to the rafters so that the sensor looks down to where the door track holds the retracted garage door; see figure 4-2.

*A picture containing indoor, sitting, table, small

Description automatically generated*

*Figure 4-2. How the Project will be Mounted.*

After drilling out the mounting holes on the bracket, and before hot gluing the sensor to the bracket, take four female-female jumper wires and insert one end of each onto the Vcc, Trig, Echo, and Gnd pins on the sensor. Make note of which color wire goes to each of these contacts. We suggest (not mandatory) that you use the following color code for these jumpers:

* Vcc: red wire
* Trig: blue wire
* Echo: green wire
* Gnd: black wire

Put the sensor aside and let the glue harden.

The next step in the assembly is to prepare to mount the remote control unit in the plastic project box. We recommend doing this by drilling two holes in the back of the remote’s plastic case, as shown in figure 4-3. The location of the holes is not critical; the selected points were easily accessible and don’t interfere with remounting the circuit board into the remote housing. We suggest affixing two nylon ½” 4-40 threaded standoffs (female – female) to the remote housing using ¼” nylon 4-40 screws, see figures 4-3 and 4-4.

A close up of a device

Description automatically generated

*Figure 4-3. Modified remote housing, top view.*

A picture containing sitting, wooden, table, board

Description automatically generated

*Figure 4-4. Modified remote housing, bottom view.*

At this point, you should have the following steps completed:

* Ultrasonic sensor hot glued to a plastic bracket, wired and ready for mounting to a rafter.
* Wireless I/O Board with the necessary components soldered onto it.
* Garage opener remote unit back of case drilled and with nylon standoffs mounted on it.

The next step is to drill out the plastic project enclosure (pencil box) to accommodate these components. Use figure 4-5 as a guide:

* Drill 4 holes in the corners of the enclosure for mounting the enclosure in the garage (11/64” holes will suffice for mounting with #6 wood screws). These holes are clearly visible in figure 4-5.
* Locate the back of the garage remote unit in the enclosure as shown in figure 4-5. Mark where the open ends of the standoffs contact the enclosure through the transparent rear of the enclosure. Drill these two holes with a 1/8” bit.
* Place the *Wireless I/O Board* in the enclosure, approximately as shown in figure 4-5. Make sure that the project mounting holes are clear of the board and that there is sufficient space between the *Wireless I/O Board* and the garage remote unit to wire the two together. Mark the *Wireless I/O Board* mounting holes and also the center of the cutout for the USB connector. Additionally, mark a location to drill a ¼” hole in the side of the enclosure below the *Wireless I/O Board* to run the ultrasonic sensor wires through the enclosure. Locate this hole a little to the right of where the sensor wires connect to the *Wireless I/O Board*; see figure 4-5. Drill these two ¼” holes.
* At this point, you may wish to mark locations for 3 or 4 holes on the top side and the bottom side of the enclosure for ventilation. We suggest using a ¼” drill bit for these ventilation holes. Drill them now.

A circuit board

Description automatically generated

*Figure 4-5. Mounting electronic Parts in the Project Enclosure*.

We recommend the following drill sizes and cutout instructions for these holes:

* Enclosure mounting holes: 11/64”
* Remote mounting holes: 1/8”
* Wireless I/O Board mounting holes: 1/8”
* Hole for ultrasonic sensor wires: ¼”
* Ventilation holes: ¼”
* USB connector access: drill out a ¼” hole and then expand the opening using a file or nibbling tool so that the USB power cable fits comfortably through and into the USB type B connector on the Wireless I/O Board.

Mount the Wireless I/O Board into the enclosure using #4-40 x ½” nylon screws and #4-40 nylon nuts. Connect the external WiFi antenna to the Photon if you plan on using an external antenna and use the antenna’s adhesive back to stick it to the side of the enclosure; see figure 4-5.

Connect a short length of #26 awg solid wire to the COM (middle contact) connection on the 3 terminal “relay” block on the *Wireless I/O Board*. Connect the other end of this wire to the “-“ terminal of the 2 terminal “solenoid” block (this connects board ground to the COM contact of the relay). See figure 4-6 for a closeup view.

A circuit board

Description automatically generated

*Figure 4-6. Relay Common Wiring.*

Now take the modified remote unit circuit board and trim down the leads that you soldered onto it earlier so that you can wire this unit to the *Wireless I/O Board* without too much excess wire. Approximately 3” of lead length should suffice. Wire these leads to the screw terminals on the *Wireless I/O Board* as shown in figure 4-7:

* The Battery + wire connects to the “MOT\_PWR” positive (+) terminal.
* The Ground wire connects to the “MOT\_PWR” negative (+) terminal.
* The pushbutton activation wire connects to the “Relay NO” terminal.

Make sure that the leads are screwed down tightly on these terminal blocks.

A picture containing indoor, sitting, table, camera

Description automatically generated

*Figure 4-7. Wiring Up the Remote Unit.*

Next, place the remote’s circuit board into the back cover of the remote unit that you previously screwed down to the enclosure. The circuit board fits under a small lip at the top of the housing and down between two aligners at the bottom of the housing. This circuit board should hold fast into the housing, but you can further affix it in place with a dab of hot glue if you wish. Tuck the leads between the remote circuit board and the Wireless I/O Board out of the way, taking care to not break any of the connections. See figure 4-8.

A picture containing plastic, suitcase, luggage

Description automatically generated

*Figure 4-8. Completed Enclosure*.

At this point, you should have a completed project enclosure and the ultrasonic sensor wired and hot glued to its bracket. The remaining steps are to mount the ultrasonic sensor and the project enclosure in your garage at a location where:

* The ultrasonic sensor “looks” directly into the path of the garage door, and
* The project enclosure is just above and perhaps a little to the left of the ultrasonic sensor.

See figure 4-9 for a representative mounting of these parts to the rafters of a garage. We suggest using #6 x 1.25” wood screws to secure the project enclosure and the ultrasonic sensor bracket to the garage (rafter).

A picture containing wooden, indoor, table, sitting

Description automatically generated

*Figure 4-9. Representative mounting in a garage above the garage door track.*

Feed the four wires from the ultrasonic sensor through the ¼” hole in the bottom side of the enclosure. Connect the four wires as follows (refer to figure 4-10):

* Sensor “Vcc” goes to the middle pin (pin2) of the “servo” pin header.
* Sensor “Trig” goes to the bottom pin (pin 1) of the “servo” pin header.
* Sensor “Echo” goes to the middle pin of the “Door Sensor” terminal block field (note: you soldered a single male pin header here and not a terminal block)
* Sensor “GND” goes to the top pin (pin 3) of the “servo” pin header.

Double check your connections to make sure that they are correct. Now place the enclosure cover on the enclosure and secure it with the side snaps.

*A circuit board

Description automatically generated*

*Figure 4-10. Wiring Diagram for the Ultrasonic Sensor.*

The final step in the project assembly is to supply power to the unit. Plug the 5 volt “wall wart” power supply into the nearest convenient power outlet and connect the “A” end of a USB cable to it. Run the cable through the garage to the top of the project enclosure and plug the “B” end of the USB cable into the USB connector on the *Wireless I/O Board*. The project should now be powered and the Photon’s multicolor USB light should be blinking, flashing, or “breathing”, depending upon the previous state of the device.

See figure 4-11 for a completed installation.

A picture containing wooden, building, table, wood

Description automatically generated

*Figure 4-11. Completed Installation in a Garage.*

Now that the hardware fabrication and installation is compete, it is time to install the Photon firmware and test out the completed project. See section 5, below.

# FIRMWARE INSTALLATION AND TESTING INSTRUCTIONS.

Firmware is the program that you need to install on your Photon. The firmware is provided in the form of source code. You will need to make small modifications to this source code, after which you need to compile the firmware and flash it to your Photon.

Before you can perform any if these steps, you need to get your Photon working on your WiFi network and claimed into your Particle account. If you don’t have a Particle account, you need to create one (it’s free). Particle provides complete documentation for this. Begin at:

<https://docs.particle.io/quickstart/photon/>

You can perform these steps either by removing the Photon from the *Wireless I/O Board* and powering it directly using a USB cable (as shown in Particle’s on-line documentation) or by performing these steps with the Photon in place in the installed project from section 4. We recommend that, however you do this, you do it in a location with a strong WiFi signal and in a place where you can observe the multi-color LED on the Photon.

If you are not familiar with Particle’s various integrated development environments, you can read up on the Web IDE and the Particle Workbench at:

<https://docs.particle.io/tutorials/developer-tools/build/>

The source file for the firmware for this project can be found at:

<https://github.com/TeamPracticalProjects/Garage_Door_Controller/blob/master/Software/Photon/src/GarageDoorController.ino>

This file is in Arduino source code format, which is just plain text and can be edited with any text editor or with either of Particle’s two supported integrated development environments (IDEs). If you are a novice Particle user, we suggest that you use the Web IDE. If you are familiar with the Particle Workbench, you may alternatively use it.

Whichever IDE that you use, you may need to make a few changes to the firmware that we have provided:

* Our default for how long the relay activated the garage door remote is 2 seconds. This value works reliably with our garage door opener. If you need to extend (or shorten) this time, you will need to edit the firmware line 29, which reads:

const int BUTTON\_TIME = 2000; // time to trip the relay to "press" the garage opener button

The value “2000” here is in milliseconds (2000 ms = 2 seconds). Change this value to whatever works best for your garage door system. If you are unsure, keep this default value, as you can always go back and change it after testing out your unit.

* Lines 15 and 16 are both commented out. The Photon has two WiFi antennas: an internal “patch” antenna and a uFL connector for an external WiFi antenna. You may have installed the external Wifi antenna while performing the steps of section 4. However, if you never changed the antenna since you first unboxed and claimed your Photon, the Photon’s default is the internal antenna. If you wish the keep the Photon running on the internal antenna, you don’t need to change anything – leave both of these lines commented out. However, if you wish to change to the external antenna, you need to uncomment the line:

STARTUP(WiFi.selectAntenna(ANT\_EXTERNAL)); // uncomment to use an external WiFi antenna on the Photon

If you ever selected the external antenna and wish to change back to the internal antenna, re-comment this line (line 16) and uncomment line 15, as follows:

STARTUP(WiFi.selectAntenna(ANT\_INTERNAL)); // uncomment to use the Photon's internal patch antenna

// STARTUP(WiFi.selectAntenna(ANT\_EXTERNAL)); // uncomment to use an external WiFi antenna on the Photon

After editing the firmware source code, above, you use whichever IDE your chose to save your changes, compile your changes, and flash the firmware to your Photon:



* Use the Particle Web IDE: Open a new project in the Web IDE and delete the default setup() and loop() templates. Open the file “GarageDoorController.ino” in this repository with any text editor or word processor program[[2]](#footnote-2). Make the changes described above and save the file to your computer. Now copy everything in the program and paste it into the blank template in the Web IDE. Be sure that all lines of “GarageDoorController.ino” are copied into the Web IDE window and save the program using any name that you wish. Next, use the Web IDE to select your target Photon and flash the code to it. For more information about the Particle Web IDE, see:

<https://docs.particle.io/tutorials/developer-tools/build/>

* Use the Particle Workbench: Copy the file “GarageDoorController.ino” onto the computer where you are running the Particle Workbench. Make the changes described above using the built-in editor in Particle Workbench and save your work. Use Workbench to compile and flash this code to your Photon. For more information about the Particle Workbench, see:

<https://docs.particle.io/tutorials/developer-tools/workbench/>

The Web IDE approach is probably easier for someone who is new to Particle firmware development, as it is all web based. However, if you plan to develop firmware for Particle devices, we recommend that you install and learn the Workbench.

If you removed the Photon from the Wireless I/O Board, now is the time to unpower it and place it back into he Wireless I/O Board in your project. Power up the project and make sure that the Photon re-connects to the Particle cloud via WiFi.

You can test that your project is working using the Particle Console. You can find out more about the Console from Particle’s on-line documentation. You can access the Particle Console by going to:

<https://console.particle.io/>

Log in to your Particle account to get to the Particle Console home screen. Select the “Devices” icon on the right hand side of the screen to see a list of the Particle devices that are in your account. Select the Photon device that is on your *Wireless I/O Board*. Make sure that the Board is powered up and that the Photon’s multicolor LED is breathing Cyan, indicating that it is connected to the Particle Cloud and running the firmware that you previously flashed to it.

After selecting the Photon device, you will get a Device screen. You should see two variables and one function at the lower left hand side of the screen, similar to figure 5-1.

\*\*\*LEFT OFF HERE\*\*\*

A screenshot of a cell phone

Description automatically generated

*Figure 5-1. Cloud Variables and Function.*

*A screenshot of a cell phone

Description automatically generated*

*Figure 5-2. Cloud Variable Data.*

*A screenshot of a cell phone

Description automatically generated*

*Figure 5-3. Console Health Check.*

You can then test these functions on your *Wireless I/O Board*, as follows:

* tripLock: The Cloud function “tripLock” requires no argument, so leave the “Argument” field blank. Selecting CALL activates this function on the Photon. The function exercises a small motor connected to the Motor Terminal Block. The function first pulses the motor to move in one direction for a total of 30 ms and then stop. The function waits for two seconds and then pulses the motor for 30 ms in the opposite direction. The motor that we are using for this test is part of the Tokatuker Cabinet Lock that we modified to be controlled via the Wireless I/O Board:

<https://www.amazon.com/Tokatuker-Electronic-Cabinet-Hidden-Drawer/dp/B075QF1VPR>

Pulsing this motor for 30 ms causes the shaft of this particular motor to turn approximately 180 degrees.

* testIO: The Cloud function “testIO” requires no argument, so leave the “Argument” field blank. Selecting CALL activates this function on the Photon. The function sounds the on-board buzzer, activates the external LED and activates the on-board relay; all for two seconds. After two seconds, all of these parts are deactivated. In order to test these parts of the Board, connect an external LED to the external LED terminal block and use a multimeter or other device to measure voltage across the Solenoid Terminal Block connectors and to measure resistance across the Relay Terminal Block connectors. The buzzer should sound for two seconds, the LED should light for the same two seconds, the solenoid voltage should go from zero to 3.3 volts (or whatever voltage is the Motor Power) for the same two seconds, the relay resistance between COM and NO should go from infinity to zero for the same two seconds, while the relay resistance between COM and NC should go from zero to infinity for the same two seconds.
* moveServo: The Cloud function “moveServo” requires an argument – an argument of “0” moves the servo to a position of 5 degrees, and any other argument (including no argument) moves the servo to a position of 85 degrees. Selecting CALL activates this function on the Photon. Make sure that you have a hobby servo properly plugged in to the Servo pin header connector when testing using this function.

You might also wish to flash the Particle “Tinker” firmware to your Photon and use the Particle Tinker App to exercise functions on the Board directly via the Photon’s pins.

# ANDROID APP INSTALLATION INSTRUCTIONS.

1. <https://www.particle.io> [↑](#footnote-ref-1)
2. When using a word processing program for this purpose, be sure that it is in plain text mode. [↑](#footnote-ref-2)