Garage Door Controller
Build and
Installation Instructions

By: Jim Schrempp and Bob Glicksman; v3, 7/30/2020

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

TABLE OF CONTENTS.

1. 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.

A. Assemble a *Wireless I/O Board* with the necessary components for this project – see section 2 of this document.

B. Modify the remote control unit for use in this project – see section 3 of this document.
C. Drill mounting holes in the remote control unit enclosure – see section 4 of this document.
D. Drill/cut the project enclosure and HC-SR-4 mounting bracket – see section 4 of this document.
E. Assemble parts in the project enclosure – see section 4 of this document.
F. Wire up all parts of the project enclosure – see section 4 of this document.
G. Install Photon firmware and test that the parts work – see section 5 of this document.
H. Mount the project in its final location – see section 4 of this document.
I. Install the App on your smartphone – see section 6 of this document.

2. WIRELESS I/O BOARD ASSEMBLY INSTRUCTIONS.

The *Wireless I/O Board* provides the basic electronics for this project, including the Particle[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:

[1] https://www.particle.io

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 block (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.

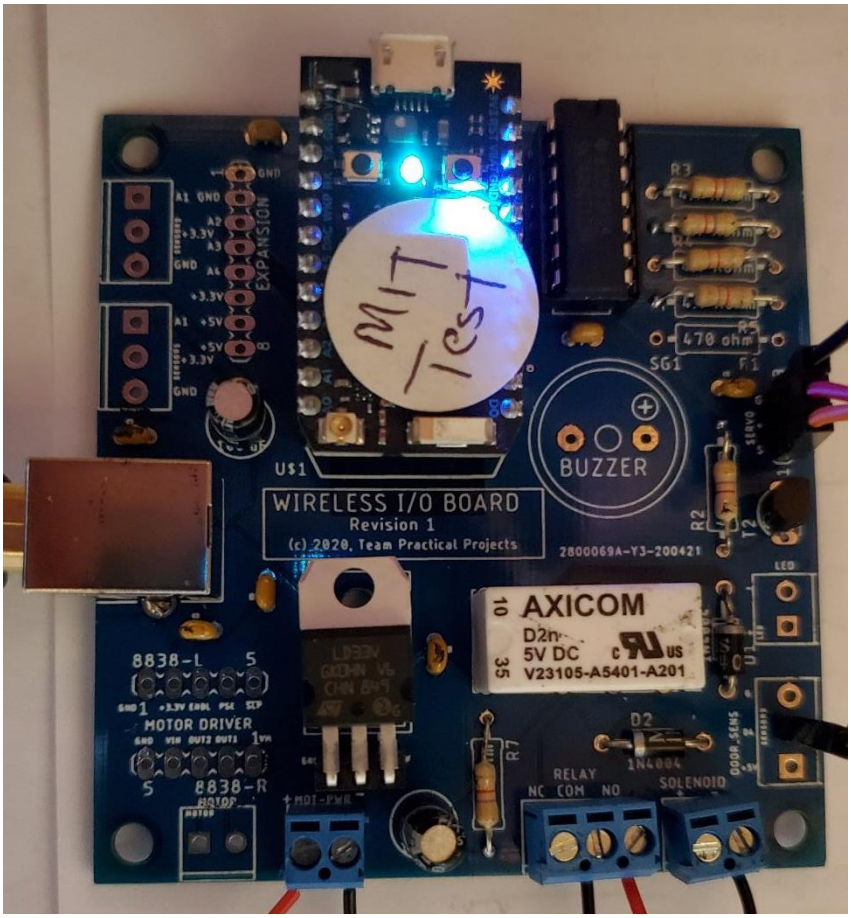

Figure 2-1. Wireless I/O Board Assembled for this Project.

3. 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.

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.

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.

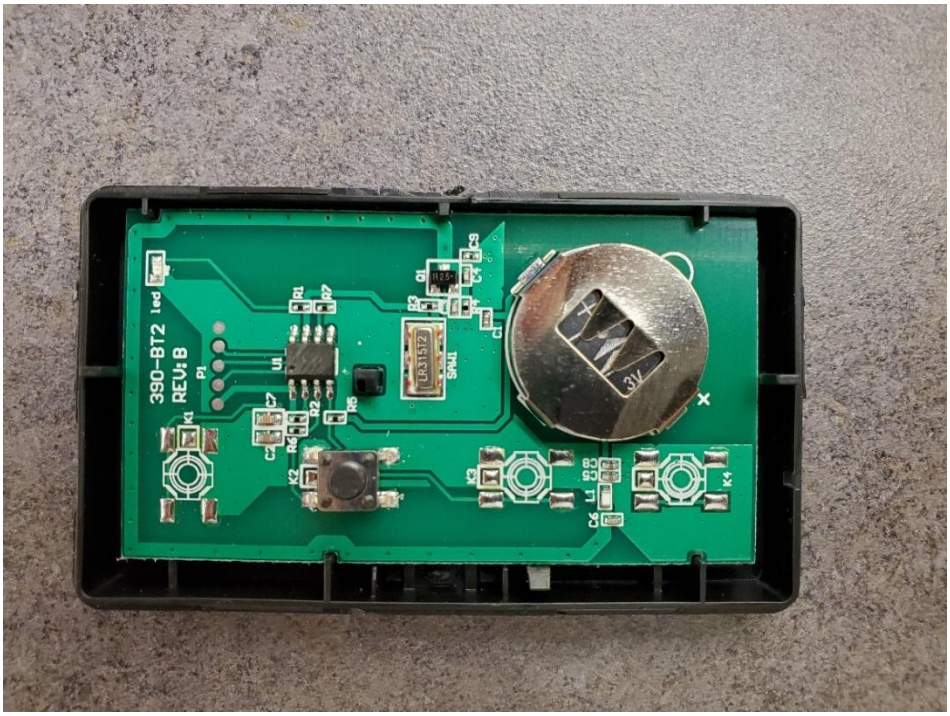

Figure 3-3. Remote Unit Opened Up.

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.

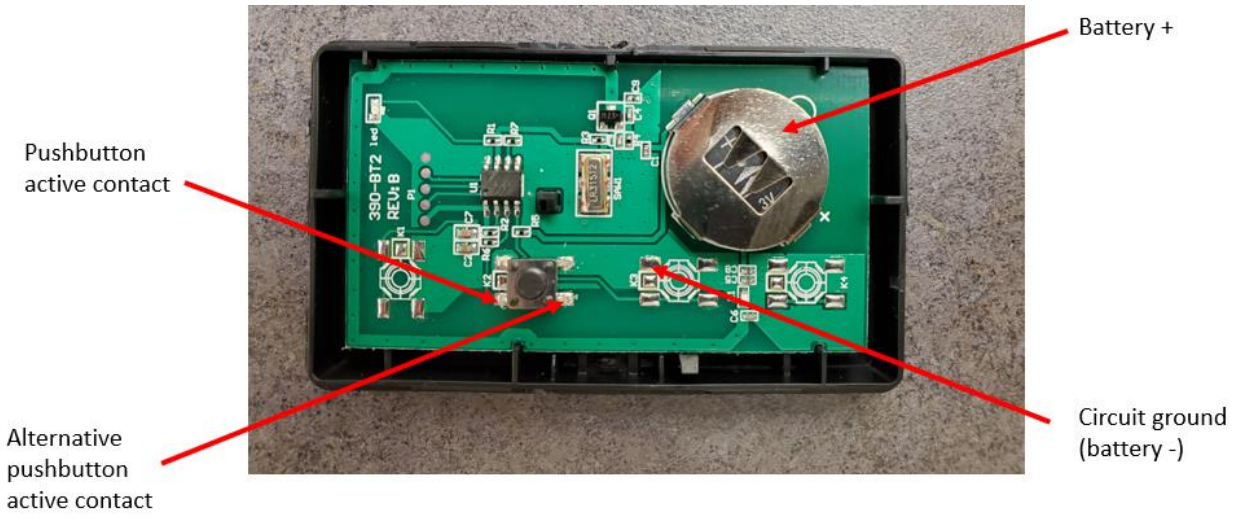

Figure 3-4. Modification Points on the Remote Unit.

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 accessible 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.

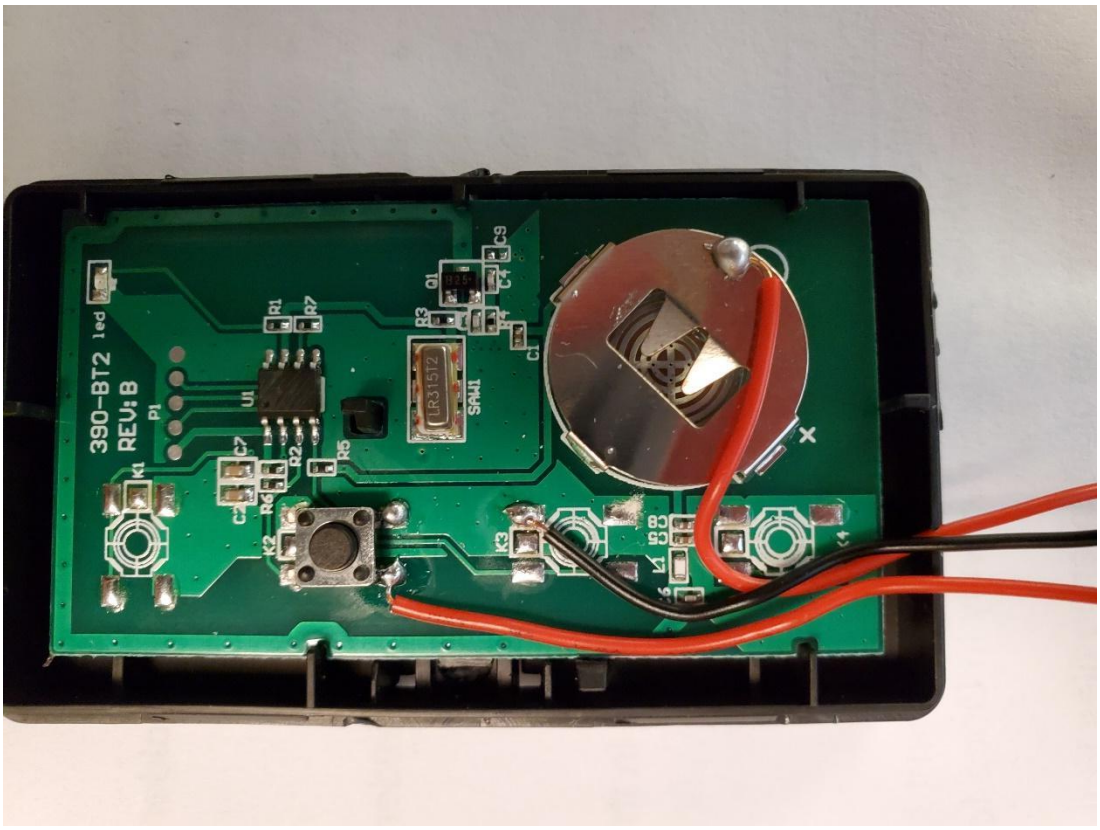

Figure 3-5. Modifications to the Remote Unit.

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.

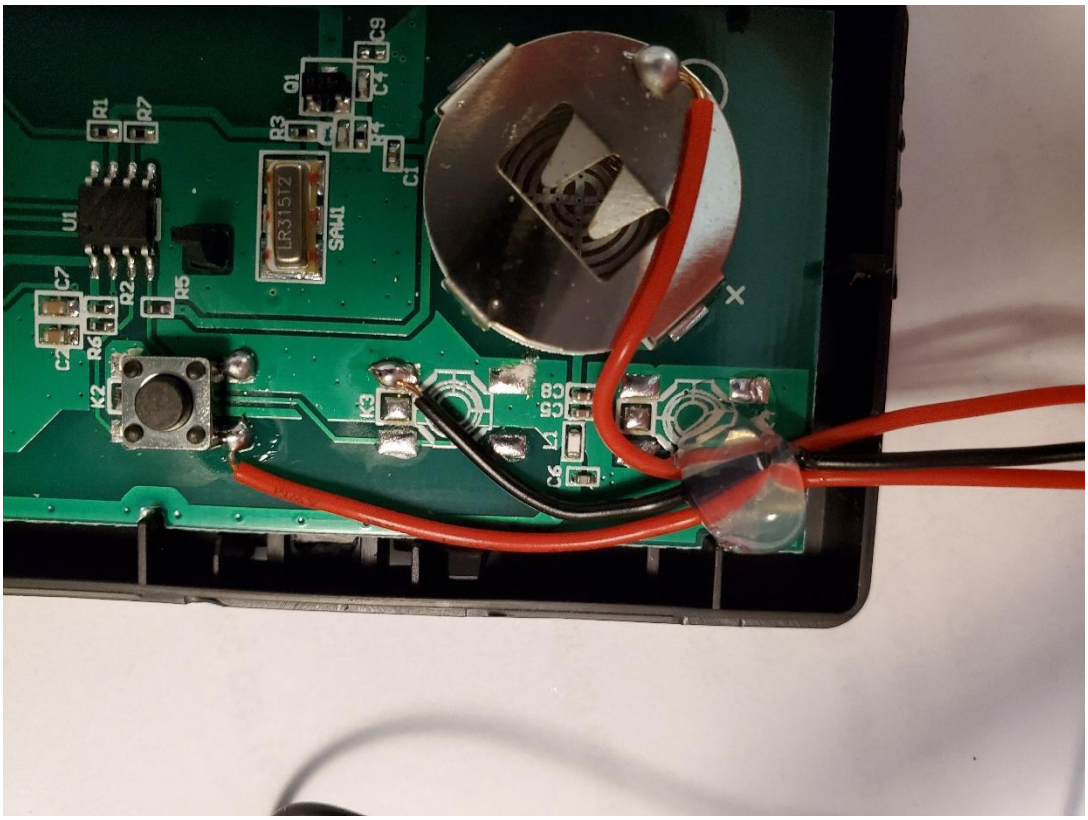

Figure 3-6. Hot Glue Strain Relief.

4. 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.

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.

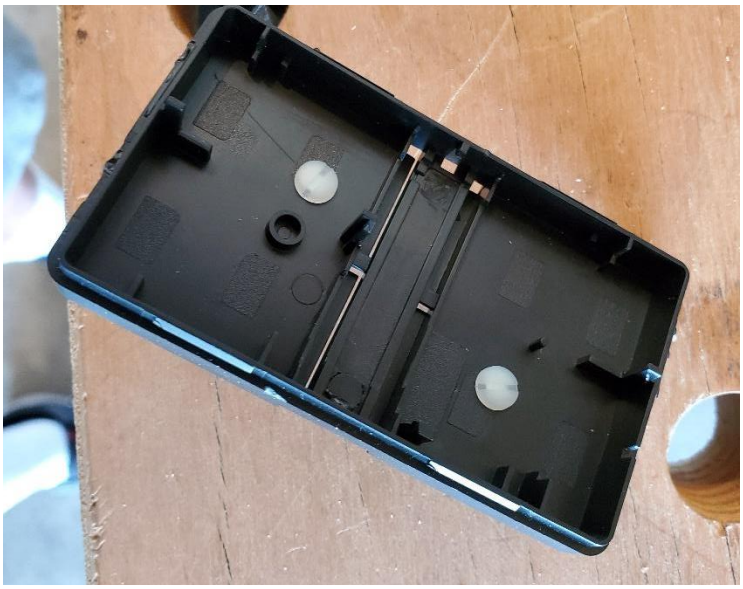

Figure 4-3. Modified remote housing, top view.

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. Drill out the four *Wireless I/O Board* mounting holes with a 1/8" bit.

- 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.

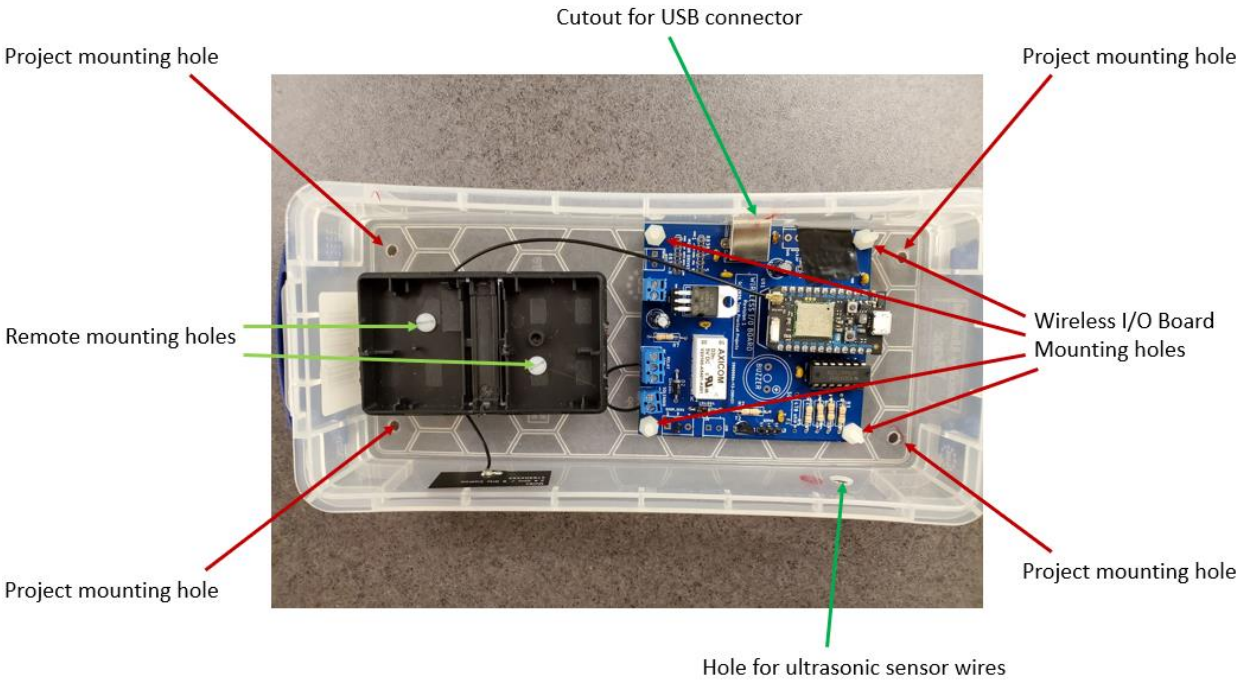

Figure 4-5. Mounting electronic Parts in the Project Enclosure.

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.

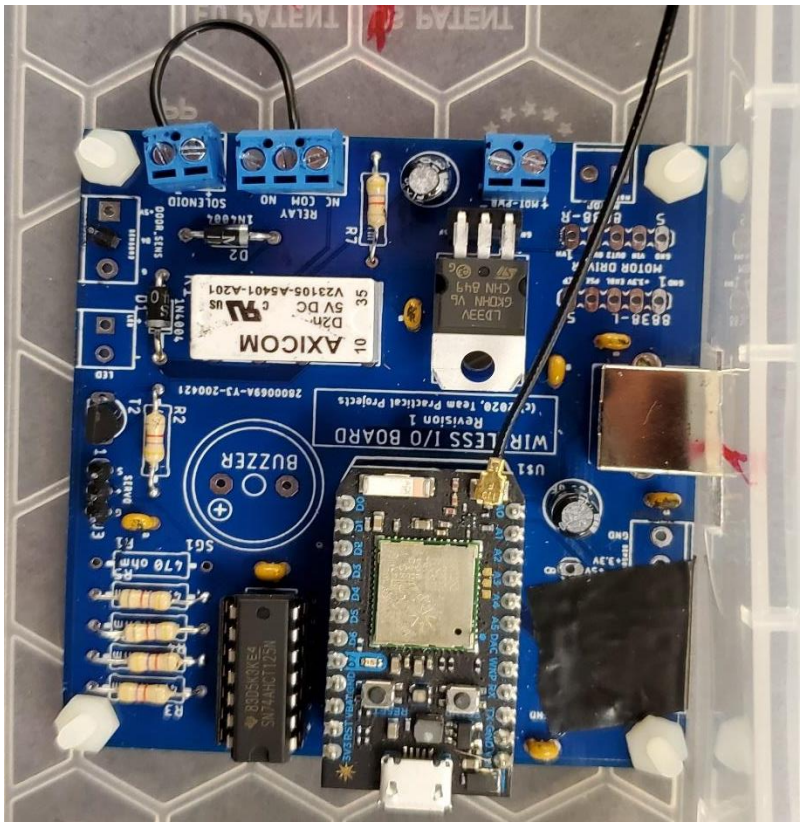

Figure 4-6. Relay Common Wiring.

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.

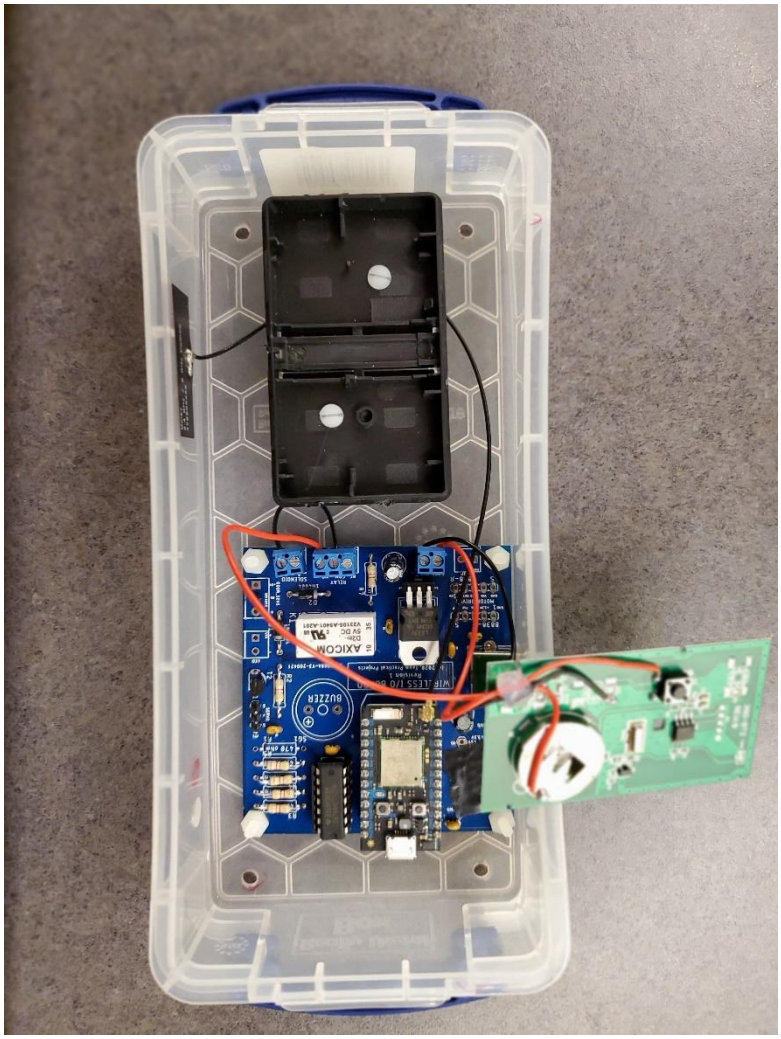

Figure 4-7. Wiring Up the Remote Unit.

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.

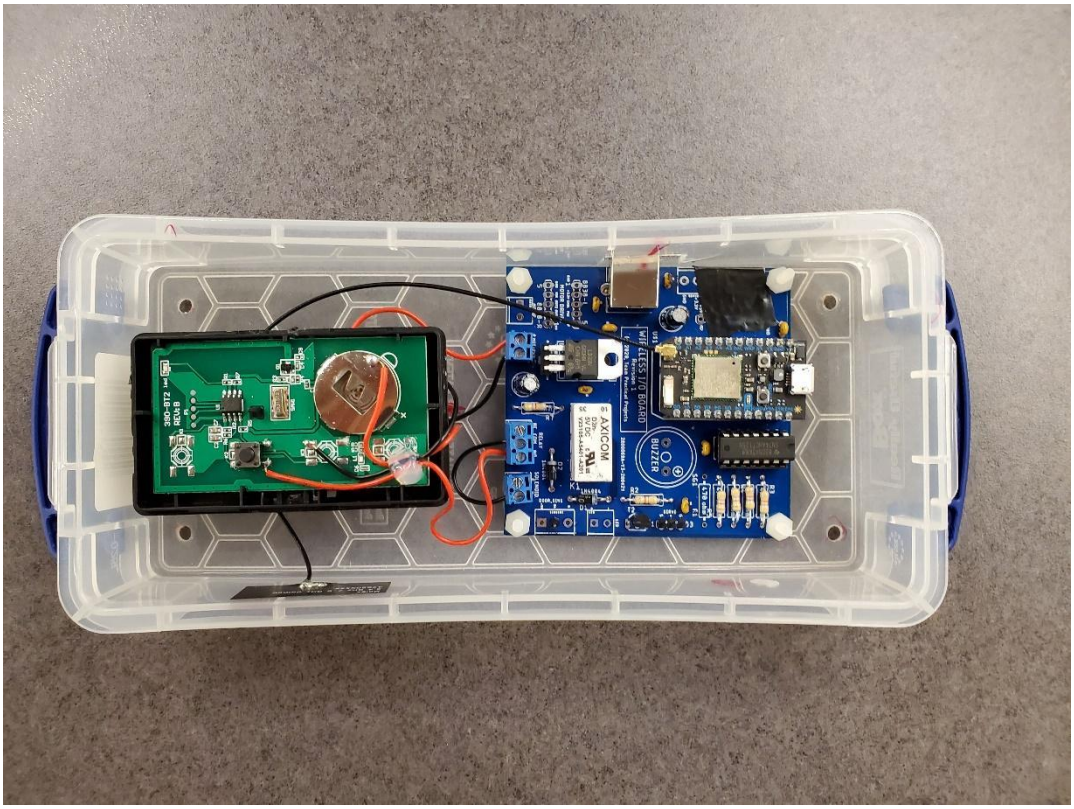

Figure 4-8. Completed Enclosure.

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).

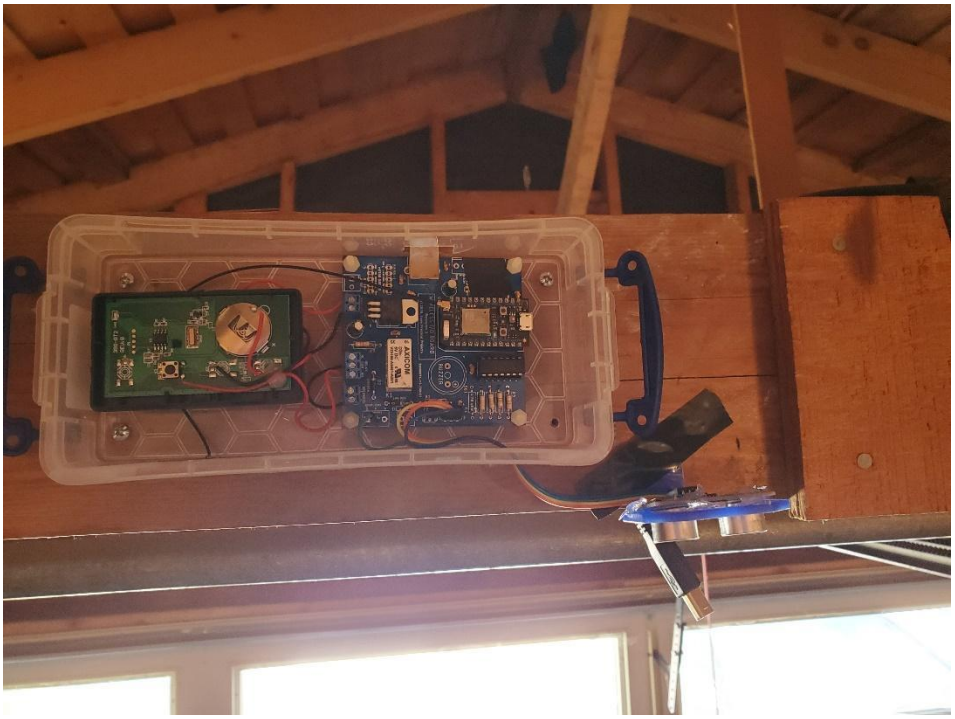

Figure 4-9. Representative mounting in a garage above the garage door track.

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.

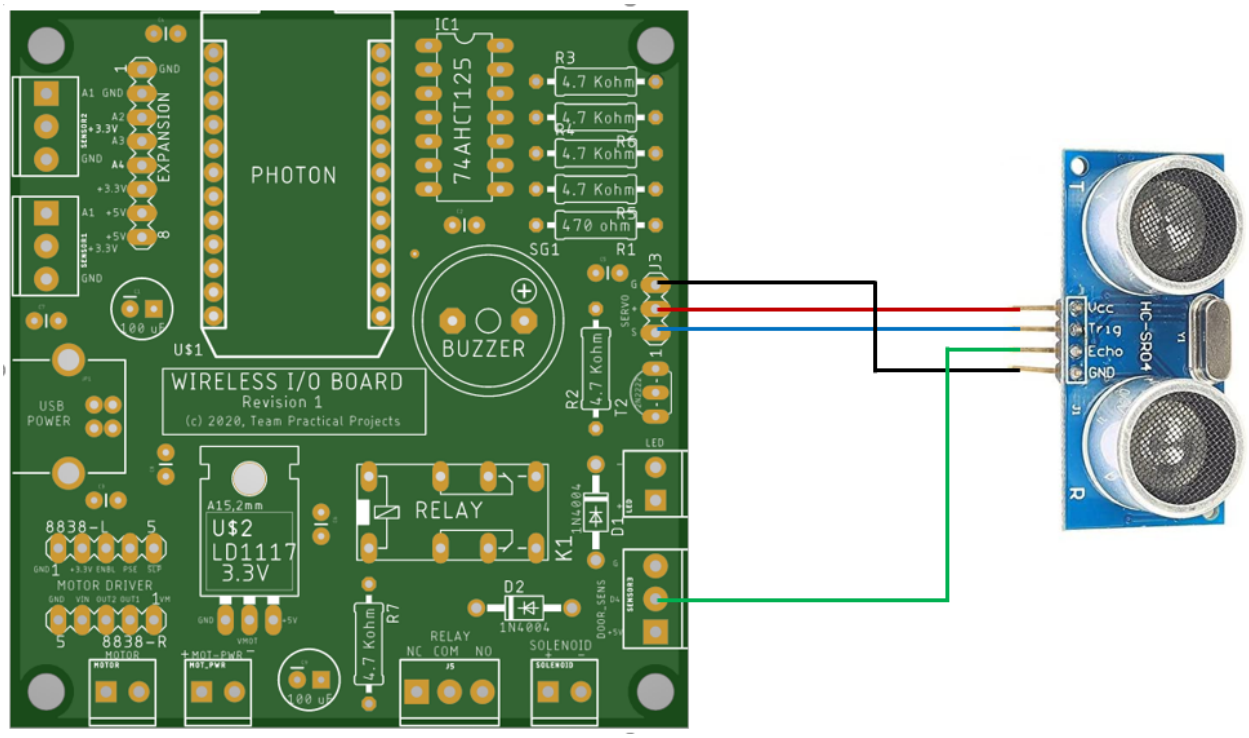

Figure 4-10. Wiring Diagram for the Ultrasonic Sensor.

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.

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.

5. FIRMWARE INSTALLATION AND TESTING INSTRUCTIONS.

Firmware is the program (software) that you need to install on your Photon. The firmware for this project is provided in the form of source code. You may 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 of 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 activates 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 26, 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.

- Line 27 of the firmware contains a constant that is the decision threshold between *door open* and *door closed*. Our default is 24.0 inches (2 feet). If the measured distance is greater than this distance, the door is declared to be closed; otherwise, the door is declared to be open. Your installation may dictate changing this criteria. If so, change line 30 to whichever number of inches your need:

  ```
  const float DECISION_DISTANCE = 24.0;   // door is closed if distance > 24 inches.
  ```

- Lines 13 and 14 of the firmware 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 14) and uncomment line 13, 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, 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]. Make the changes described above and save the file to your computer. Now copy everything in the program and paste it into

[2] When using a word processing program for this purpose, be sure that it is in plain text mode.

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 the *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 left hand side of the screen to see a list of the Particle devices that are in your account. Select the Photon device that is used for this project. Make sure that the *Wireless I/O 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 to the Device screen on the Particle Console. You should see two variables and one function at the lower right hand side of the screen, similar to figure 5-1.

Figure 5-1. Cloud Variables and Function.

Under “FUNCTIONS”, you should see “toggle_garage_door”. Clicking on “CALL” will call this function which should activate the garage door remote for 2 seconds (or whatever time you changed the default to). The function “toggle_garage_door” does not require any arguments; you can leave the “Argument” text field blank.

Our garage door remote has a little LED in the upper left corner of its circuit board. It will flicker when the remote is activated, so you can test that the garage door activation is working even if you are performing this test out of range of the garage door opener.

Under “FUNCTIONS”, you should also see “take_measurement”. Clicking on “CALL” will call this function which should take an ultrasonic distance measurement and return one of the following codes:

- “-1” if there was a measurement error from the ultrasonic sensor
- “0” if the garage door is closed
- “1” if the garage door is open

The function “take_measurement” does not require any arguments; you can leave the “Argument” text field blank.

The same Console screen shows one “VARIABLES”. The variable named “door_distance” should display the distance (in inches) that the ultrasonic sensor measured after CALLing “take_measurement”. Click “GET” to obtain the reading. If the project is in its final location, the distance measurement should correspond to the distance to either the garage door or to

whatever is behind the garage door. If you haven't yet placed the project in its final location, try displaying a few distance measurements by placing your hand or some other object in front of the ultrasonic sensor.

A sample display of the measured value is shown in figure 5-2. In this example, the measurement is 42.7 inches which indicates that the garage door is closed (when the garage door is open in our installation, the door is less than 24 inches from the sensor).

Figure 5-2. Cloud Variable Sample Data.

When your project is complete and mounted, you might want to use the Particle Console "Health Check" to make sure that your Photon is running properly; specifically, to determine the WiFi signal strength. See figure 5-3, which shows this part of the Console's Device page. Click "Run health check" at the right side of the screen. The left/middle of the Console screen presents the health check results.

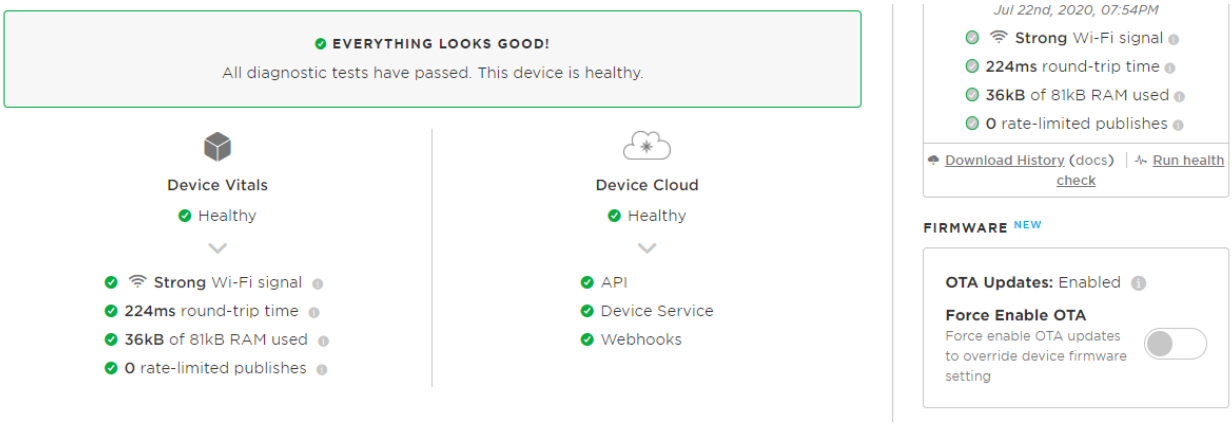

Figure 5-3. Console Health Check.

6. ANDROID APP INSTALLATION INSTRUCTIONS.

This project includes an Android app. This App allows the user to query the status of the garage door from anywhere there is Internet access. The App also allows the user to activate the garage door – to open (if it is closed) or to close (if it is open).

We have provided both source code and an installation file for this App. Source code (in MIT App Inventor 2 source file format) can be found at:

https://github.com/TeamPracticalProjects/Garage_Door_Controller/blob/master/Software/App/src/GarageDoorController.aia

You need this file only if you want to view and/or modify the app.

The installation file can be found at:

https://github.com/ TeamPracticalProjects /Garage_Door_Controller/blob/master/Software/App/executable/GarageDoorController.apk

The installation file is called *GarageDoorController.apk*.

In order to install this app on your Android device, you must transfer this installation file to your device. There are several ways to do this; here are three suggestions:

- Downloading: Download the file *GarageDoorController.apk* to your Android device using a web browser on the device. Make sure that you know the folder on the Android device where the browser stores downloaded files.

- Sideloading: Download the file *GarageDoorController.apk* to your desktop or laptop computer. Attach a USB cable from your desktop/laptop to your Android device. The Android device's flash memory should appear as a drive on your computer. Drag and drop the installation file to a location on the Android device where you will be able to find it later.

- E-mail: Download the file *GarageDoorController.apk* to your desktop or laptop computer. Email this file to yourself as an attachment. Open email on your Android device and download the attachment, making sure that you know where the email program/browser stores attachments.

Regardless of the method that you use to get the file *GarageDoorController.apk* onto your Android device, you now use a file explorer on your device to locate the file and tap on it to start the installation. The App should now install on your Android device. NOTE: you may get a message on your Android device saying that the App is "untrusted" or "unverified". This means that Android knows that the App did not come from the Google play store. Go ahead and install it anyway – it is safe! The installer will list the permissions that the App needs to run. Go ahead and accept them. The App should install properly.

The App uses our "Particle App Template". This template provides the app with a Setup screen where you can:

- Log in to your Particle Cloud account using your Particle user ID and password.
- Obtain a list of all of your Particle devices from the Particle Cloud.
- Select the device that you are using in this project.
- Exit setup to go to the App's main screen

You will need to set up the App the first time that you use it. After the initial setup, the App will remember your settings and you do not need to return to Setup unless you wish to change devices or Particle accounts.

Once the App is installed on your Android device, tap on its icon to open it. You should see a screen similar to figure 6-1, except that the currently selected station message field at the bottom right of the screen will be yellow for some time and ultimately turn red with an error message. This is because the App needs to be set up the first time that it is used.

Figure 6-1. App Main Screen.

Tap the “Setup” button at the bottom left of the App screen to go to the App’s Setup screen. The Setup screen looks like the picture in figure 6-2, with no information in the text boxes at the top.

Figure 6-2. App Setup Screen.

Enter your Particle User ID and Particle User Password (PW) in the fields at the top of the Setup screen. Once the Particle User ID and Particle User PW are entered, tap on the button “Request Device List” which is directly under the Particle User ID text field. If the information that you entered was correct, you will get a screen similar to figure 6-3.

Figure 6-3. Request Device List.

Note the following in figure 6-3:

- A long, hexadecimal string of characters will appear to the right of “user:”. This is the Particle OAUTH2 user access token from the Particle Cloud for the Particle account that you just logged into.
- “Response:” should show “OK”
- “Response Code:” should show “200”
- The button immediately below “Response Code” should change from “No devices” to “Select device”

If you don’t get this response information, then you probably typed in an incorrect User ID or PW. Triple check this; it is easy to make a mistake. Also, make sure that your Android device is connected to the Internet; otherwise, the Particle Cloud is unreachable.

After getting the screen in figure 6-3, tap the button “Select device”. You will now get a pick list showing all Particle devices in your Particle account; see figure 6-4.

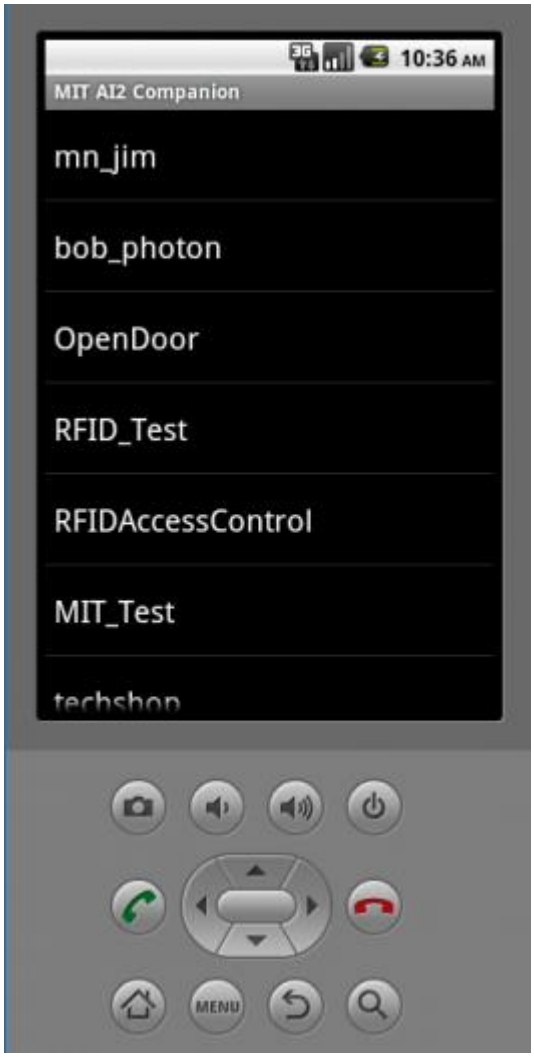

Figure 6-4. Device List Picker.

Tap on the device that you want to configure (the Photon used in this project). The screen will return to the Setup screen, similar to figure 6-5.

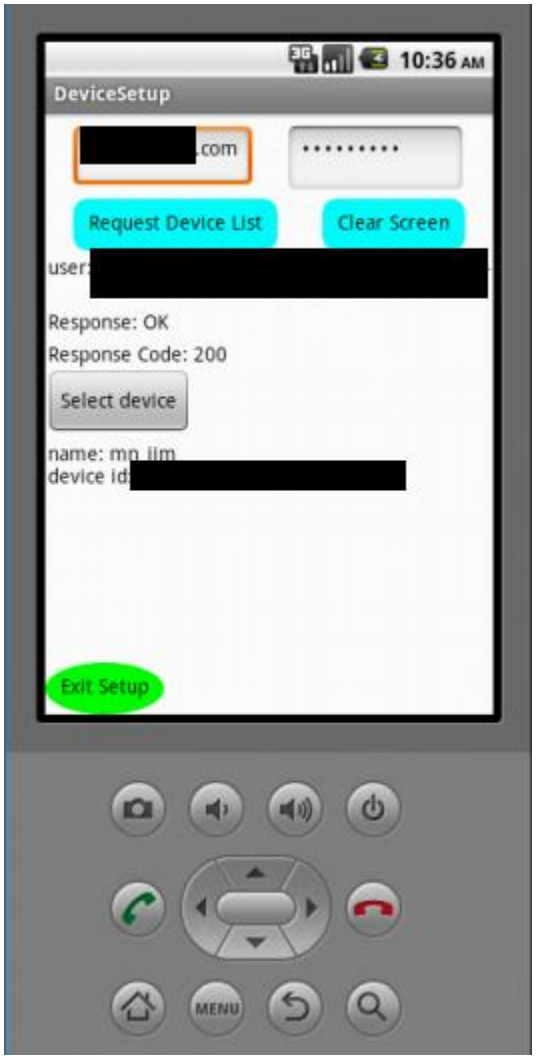

Figure 6-5. Device Selected – Exit Setup.

Referring to figure 6-5, you should now see “name:” followed by the name of the device in the pick list that you selected and also “device id:” with a long hexadecimal string after it. The Particle device (Photon) has now been selected for access by the App. Now tap the button “Exit Setup” at the bottom left of this screen. The App should now return to the opening App screen as shown in figure 6-1.

Note that now the device message at the bottom right of the screen should be green (after a brief time in yellow) and should show the name of the device that you selected and the fact that it is currently online.

NOTE: The information that you provided on the Setup screen is now stored in the App. Hereafter, when you open the App, you should get the screen shown in figure 6-1 with the previously selected device showing at the bottom right of the screen.

The App is now installed and set up. See the following document for instructions about using the App:

https://github.com/TeamPracticalProjects/Garage_Door_Controller/blob/master/Docs/Garage%20Door_Controller_User_Manual.pdf