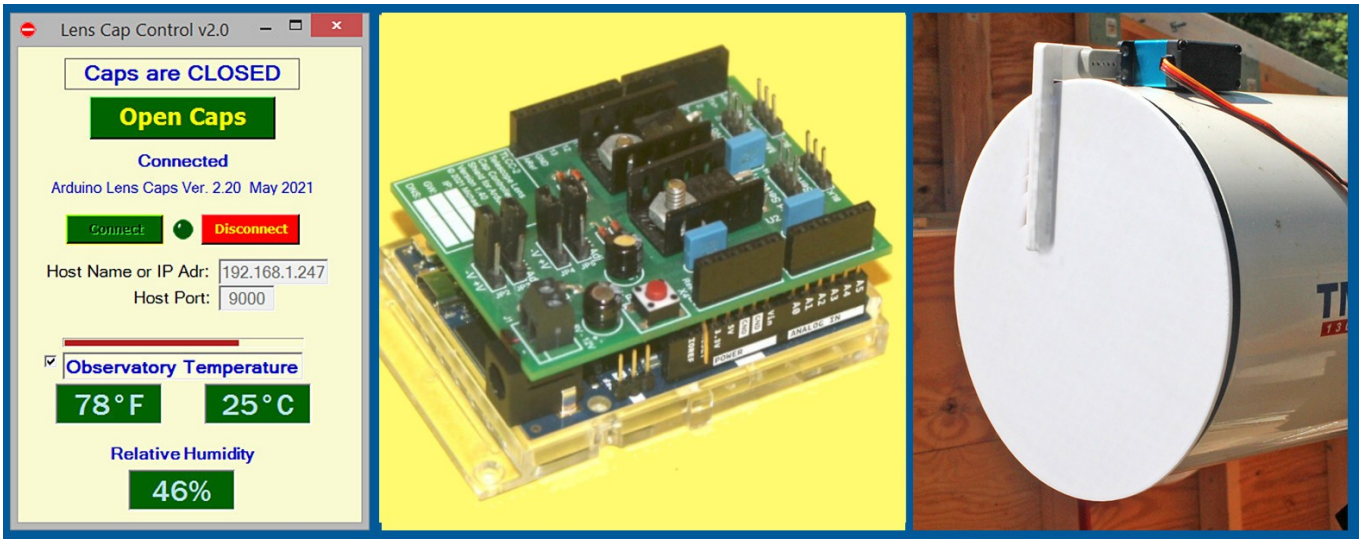
TLCC Telescope Lens Cap Controller

User Manual



by

Mike Dodd

mike@mdodd.com

Version 2.25

December 27, 2023

Table of Contents

Introduction

Many amateur astronomers operate their telescopes remotely via a network connection, often using specialized software to photograph the night sky. They can open the observatory dome, move the telescope, acquire images, and shut down at the end of a session without being present in the observatory.

If you're one of these astronomers, you've probably been frustrated with the telescope lens cap. Either you must go out to the observatory to remove it at night and replace it in the morning, or you just leave the cap off all the time.

The **TLCC Telescope Lens Cap Controller** solves this problem! Now you can open and close your telescope's lens cap remotely. The TLCC offers these features:

- ✔ Arduino microcontroller for reliable operation.
- ✔ One or two high-torque servo motors (“servos”) to move one or two lens caps.
- ✔ Windows PC-based control program (“the App”) connects to the Arduino controller via WiFi.
- ✔ Optional sensor reports observatory ambient temperature and relative humidity.

User manual organization

This manual contains red-colored “hyper links” to other pages and even websites. Click the mouse pointer on a link to go directly to that location. For example, click a red chapter number in the list below jump to that chapter.

- Chapter 1 describes the TLCC system.
- Chapter 2 explains how to install and connect the TLCC.
- Chapter 3 describes how to operate the TLCC.
- Chapter 4 describes how to install the TLCC on your telescope.
- Chapter 5 contains operating instructions.
- Chapter 6 has technical information about the TLCC..

Page numbers in the Table of Contents also are links, so you can click those, too.

Read your PDF reader's help file to learn how to return to the page you were reading when you clicked a link. This could be something like “previous page” or “previous view.”

What's in the box

Look for these items as you unpack your TLCC system:

- Two electronic circuit boards plugged together called the “stack” (Figure 3).

- A power adapter that plugs into a standard 120VAC household receptacle.
- A USB cable to connect the Arduino board to your PC when uploading a program to it. You will rarely, if ever, do this; the TLCC does not need a USB cable for normal operation.
- One or two servos, the quantity you ordered.
- With each servo, a lens cap mounting bracket and a machine screw to attach it to the servo shaft (Figure 1). Find the screw in the hole on the end of the shaft.
- A three-wire 12” extension cable for each servo.
- A flash drive containing this user manual, the control program, and the Arduino program.

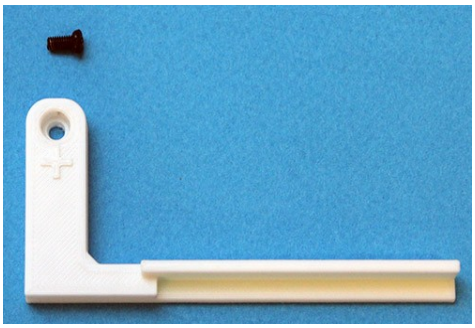


Figure 1: Cap mounting bracket and screw

Chapter 1 – TLCC System Description

The TLCC Telescope Lens Cap Controller has these components:

- **In the observatory:**
  - An Arduino Uno R4 microcontroller (“the Arduino” in this manual).
  - A circuit board called a “shield” that plugs into the Arduino, and supplies power and control signals to the servos.

    Definition: The Arduino and the servo shield plugged together are called “the stack.”
  - One or two high-torque (20 kg/cm) servo motors (“servos”) to move lens caps.
  - An optional temperature/humidity sensor that connects to the servo shield.
  - A power adapter that plugs into 120VAC house power and supplies 9 volts DC to the TLCC.
- **On your Microsoft Windows personal computer:**
  - A control program referred to in this manual as “the App”) that sends commands to the stack in the observatory, and receives status information back.

What you will need

Factory lens caps use friction to hold them on the telescope, so they are not suited to opening and closing with a servo. You will need to attach a lightweight flat lens cap and the included mounting bracket to a servo mounted on your telescope. Read Chapter 3 for instructions.

You will need a WiFi network connection between your PC and the stack. The Arduino is programmed with the static IP address you provided, and also with the WiFi SSID and password you provided, so it can connect to your WiFi router or access point. This information, plus as-tested servo configuration information, is printed on the back of my business card included with the system.

The App requires the Microsoft .NET (“dot-net”) Framework to be installed on your Windows PC. See the instructions below.

Chapter 2 – Setting Up and Testing the TLCC

This chapter explains how to set up and test the TLCC. It covers:

- Installing the App on your Windows PC.
- Plugging the Arduino circuit boards together.
- Connecting and testing the system.

You can do all this on a table or workbench with a nearby personal computer – you don't need to be at your telescope.

Copying and testing the App on your PC

The App sends commands to the Arduino via a WiFi connection. The App does not need to be "installed" – just copied – but it requires the Microsoft DotNet framework runtime to be installed on your Windows PC.. Click this link: https://dotnet.microsoft.com/download/dotnet-framework/thank-you/net48-web-installer to install the framework.

After installing the framework, Insert the TLCC flash drive into your computer's drive. It contains three folders:

- **TLCC** contains the Arduino firmware source code.
- **Control Program** contains the executable control program and related files.
- **User Manual** contains this user manual as a PDF file.

Copy the entire **Control Program** folder to your C: drive, then navigate to that folder and double-click **Telescope Lens Caps.exe** to launch the App. You should see a window that looks like Figure 2. If so, you have successfully installed the App. If not, send me an email (mike@mdodd.com) and I'll help you.

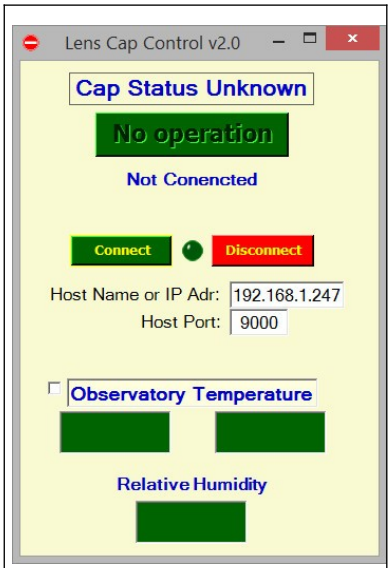


Figure 2: The App at startup

The IP address shown is the one you provided when ordering the system, and confirms that your entire system works.

Explanation: Before I copy the App to *your* flash drive, I run it, and connect to *your* TLCC for a final test. The program writes a configuration file with the IP address, and that customized file is burned to *your* flash drive along with the program file.

I suggest creating a shortcut to **C:\TLCC Control Program\ Telescope Lens Caps.exe** on your Windows desktop so you have quick access to the program.

Note: If the computer you are using to test is not the same computer you'll use to actually control your lens cap, be sure to copy and test the App on that one as well. You can copy the program to multiple computers, e.g., one in the observatory and one on your desk.

Ignore the **TLCC** folder. You need that only if you want to customize the Arduino firmware; see Customizing the TLCC sketch for details.

Remove the flash drive from your computer and store it in a safe place.

Plugging the circuit boards together

The TLCC ships with the stack (defined on page 3) assembled and ready to use. But if ever the circuit boards become separated, you need to know how to reconnect them.

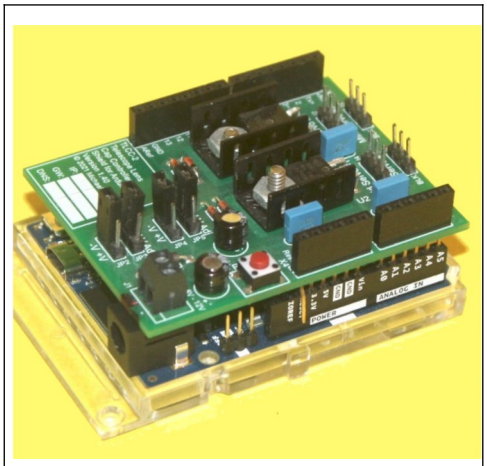


Figure 3: TLCC stack

Shields and headers

The Arduino has “headers” that allow stacking small circuit boards called “shields” to expand functionality. The TLCC uses a servo shield.

Shields have headers with sockets on top and pins extending below. The Arduino has headers with sockets on top but no pins below. Thus, a shield can be plugged into the Arduino or another shield.

The TLCC stack

Figure 3 shows the TLCC stack. The Arduino is on the bottom. The servo shield on top has headers with pins that plug into the Arduino’s sockets. The servo shield's headers also have sockets, but the TLCC has no other shield above it.

Assembling the TLCC stack

The header pins bend easily, and it can be tricky to get them all lined-up and inserted into the socket headers. I've found it easiest to align and partially insert the pins on the side with the longer headers, then align the pins on the opposite side. When all pins are aligned, gently press the shield down until it stops moving.

Important. The servo shield contains circuitry to reset the Arduino on power-up, but recent Arduino versions do not respond well to this external reset. Consequently, the Reset pin on the servo shield is bent outward, and does not mate with the socket on the Arduino board, as seen in this photo. You also can see this in Figure 3, above.

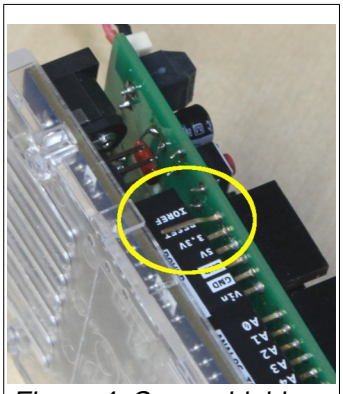


Figure 4: Servo shield Reset pin bent outward

Connecting and testing the system

After unpacking your TLCC, please connect everything and test for correct operation. Here are the steps:

- Copy the App on your PC as instructed on page 4.

- Ensure the boards in the stack are securely plugged together, as described above.
- Connect the two-pin connector from the power adapter to J1 on the servo shield. The red wire goes to the **+** terminal (left), and the black wire goes to the **–** terminal (right). Nothing will be damaged if you connect the pins backward, but the TLCC will not operate.
- Ensure your WiFi router or access point is configured correctly, and that your PC is connected to it.
- The USB port on the Arduino Uno is not used in normal operation, although it is used in the rare event that you need to upload a modified program to the Arduino. Instructions for doing this begin on page 27.
- Perform the following steps to verify that the TLCC powers-on and connects to your Wi-Fi network, and the App on your PC can connect to the TLCC.

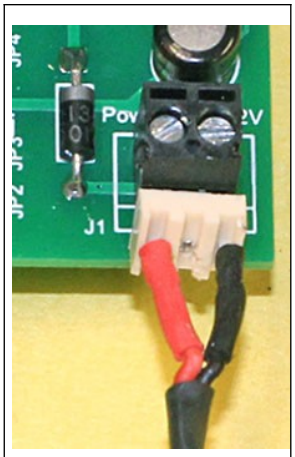


Figure 5: Power connections

- Plug the power adapter into a regular household 120VAC receptacle. The Arduino has an LED display that shows symbols or words to indicate what's happening. The display is mostly hidden by the servo shield and connecting pins, so look carefully to find it.
- After a couple of seconds, lights on the Arduino Uno should illuminate and you should see a horizontal dash on the Arduino's LED display. The servo might buzz quietly. If you purchased and connected the optional temperature/humidity sensor (page 20), the dash will change to **SEN** when the Arduino detects it.
- Over a period of 15 to 60 seconds, you should see **WIFI** on the Arduino's LED display, indicating the TLCC has connected to your WiFi network.

- Launch the App on your PC. The initial screen should look like Figure 2. If the IP address doesn't match what you specified when you ordered the system, enter the number you specified, and notify me of the discrepancy.
- Click the green **Connect** button. There will be a brief delay while the system initializes, then the window should look like Figure 6, right. You should see **Connected** and the Arduino firmware version should be displayed. In addition, the **Caps are...** text at the top should show the caps are closed, and the green button below that should display **Open Caps**. The green light between the buttons should be flashing. These all tell you the App has connected to the TLCC stack, and is receiving data from it.

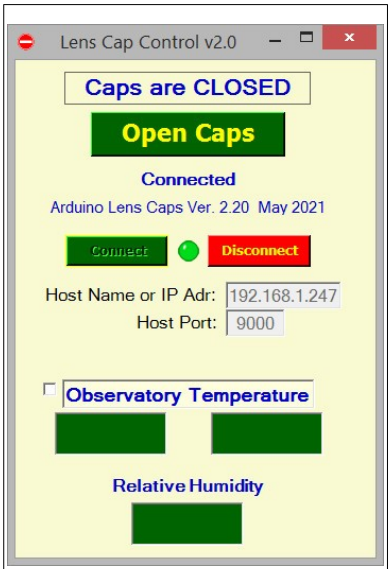


Figure 6: The App connected to the TLCC

With the App connected, you should see **APP** on the Arduino's LED display (Figure 7).

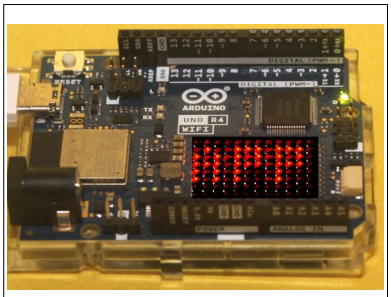


Figure 7: TLCC connected to the App

- Disconnect the App from the TLCC by clicking the red **Disconnect** button.
- Unplug the power adapter to remove power from the TLCC.
- Plug a servo into the **Std Servo #1** or **Std Servo #2** connector (J4 or J5) on the servo shield. The black or brown wire should be on the pin labeled **BLK** (left pin in Figure 8).

  It doesn't matter which **Std Servo** connector you use. If you have two servos, plug the second one into the remaining **Std Servo** connector.

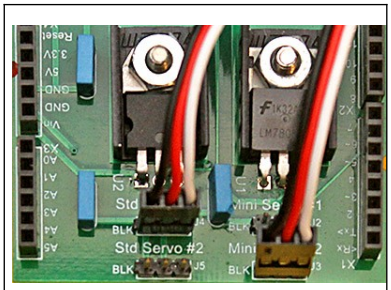


Figure 8: Servo connections

- Temporarily place a lens cap mounting bracket (page 12) on the servo's output shaft to better see servo movement in the following steps.
- Plug in the power adapter to start the TLCC. After 25-30 seconds, when you see **WIFI** on the LED display, launch the App and connect to the TLCC. Continue with the next step when you see **APP** on the LED display.

- Click the App's green **Open Caps** button. The bracket should rotate smoothly, then stop. If you have two servos, one will move first, then the second.
- After clicking the green **Open Caps** button, it should become disabled, and the **Caps are...** text above it should display **MOVING** for several seconds, then change to **OPEN**. Finally, the green button should become enabled again, and show **Close Caps**. If a servo does not move, check that the three-pin plug is not reversed where it plugs onto to the servo board.
- Click the **Close Caps** button. The bracket should rotate smoothly back to its original position. Same for the second servo if you have two. After several seconds, the green button should change to **Open Caps**, and the **Caps are...** text above it should change to **CLOSED**.
- Ignore the **Observatory Temperature** displays at the bottom of the window. See the note below for more information.
- Now let's discover and document which way the servo rotates when opening and closing. Do this:
  - With the servo in the closed position from the above step, remove the mounting bracket from the servo, orient it pointing straight out from the servo, then press it on the shaft again. This is our reference.
  - Click the green **Open Caps** button, and watch the bracket rotate to the open position. It should point about 145° away from the reference closed position when it stops.
  - Now click the green **Close Caps** button, and watch the bracket rotate back to the closed position.
  - The servo will be installed horizontally on the telescope, as shown in the photo on the title page and in Figure 15. The servo shaft must project to the left when viewed from the telescope's objective end. The cap mounting bracket must rotate *upward* when opening the cap. Try to visualize this, or actually hold the servo horizontally while you command the caps to open and close.
  - Once you are sure the bracket rotates upward while the servo is horizontal, write **Bottom** on the side of the servo that is facing downward. This is the side that will rest on the telescope. Also, plan on installing the servo with its shaft closest to the end of the telescope. In other words, if you were to place the servo on the objective end of your telescope right now, the mounting bracket would extend slightly beyond the end of the scope, and when opening, would rotate upward and toward the back of the scope.
  - With the servo in the closed position, remove the bracket so it doesn't fall off later.
- Click the red **Disconnect** button to disconnect from the TLCC stack. The App again should again look like Figure 2.
- Close the App and remove power from the stack.

***Congratulations!* Your TLCC system is working correctly.** Here are a few notes:

- The Arduino can actuate two servos, but only one at a time because the servo shield doesn't have the current capacity to move both at once. With only one servo, you might notice the following:
  - After clicking **Open** or **Close**, a delay before your servo moves, because the Arduino is moving the other servo first. If you have only one servo, you'll see nothing while the Arduino waits for the non-existent "other" servo to move. Be patient.
  - After clicking **Open** or **Close**, a delay before the lens cap status changes, while the Arduino moves the other servo. Be patient.
  - If you click **Open** or **Close** immediately after your servo stops moving, the App might not respond because that button is disabled until the Arduino signals that *both* servos have moved. Be patient.
- From a practical standpoint, it doesn't matter how quickly the servo begins moving or the status changes, but it can be disconcerting if you're watching the lens cap as you click the button. Plug the servo into the other connector to see which delay matters least.
- The Arduino rotates the servo in 1° steps, pausing 10 milliseconds (0.01 second) after each step. Thus is takes about 1.45 seconds to open or close a lens cap 145°.
- A servo can rotate up to 180 degrees. The Arduino is programmed so 0° is the closed position, and 145° is the open position. You can change this by modifying the Arduino's program, as explained starting on page 27.
- If you purchased the optional temperature/humidity sensor, temporarily connect it as instructed on page 20 to confirm it's working properly.

Introducing Amazing Goop® adhesive

The "secret sauce" for installing a TLCC is Amazing Goop adhesive. This solvent-based viscous adhesive dries into a flexible bonding agent that is waterproof and very strong, yet can be removed easily. It is extremely useful for many tasks. I've used it to:

- Glue a servo to the telescope.
- Glue a servo bracket to a lens cap.
- Glue a TLCC stack to a telescope dovetail bracket.
- Glue the Arduino and shields together for a secure connection.
- Glue servo connectors to the servo shield's pin connectors for extra security.

Figure 9: Amazing Goop, 3.7oz. tube

- Glue pin connectors (e.g., from the temperature sensor) to socket headers on the servo shield, to prevent them coming loose.

Goop comes in many “variations” such as Automotive, Household, Plumbing, Marine, etc. I haven't found much difference in them, so I use Plumbing, which is available at Lowe's: https://www.lowes.com/pd/Amazing-GOOP-3-7-fl-oz-Extreme-Condition-Clear-Multipurpose-Adhesive/3033208. Wal-Mart also carries Goop, but the price is much higher. I recommend buying the large 3.7-ounce tube; it comes in handy for many other tasks.

How to use Goop

Squeeze out the needed amount of Goop onto a toothpick or screwdriver, then apply it to one of the parts to be joined. Press the parts together, then tape or otherwise secure them for 6-8 hours while the Goop cures thoroughly.

You can apply a dab of Goop to the outside of two items (e.g., plugged connectors) to hold them together.

Wipe excess Goop from a tool with a tissue or paper towel. If some gets on your fingers (it will) rub your fingers together to make the residue ball-up, then peel it off.

Separating a Goop joint

A very nice feature of Goop is that a joint can be separated by prying or pulling the items apart. The cured Goop stretches and ultimately pulls away from one or both parts. From there, you can use fingers or pliers to grab the dried Goop and peel it off the part. Sometimes I use needle nose pliers to “wind-up” a length of dried Goop, reducing the amount of pulling force needed to peel it off.

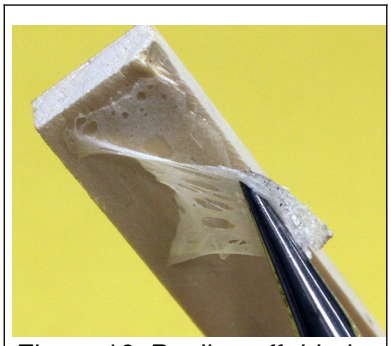


Figure 10: Peeling off dried Goop

Bottom line

I strongly recommend using Goop for your TLCC lens cap system. It is so “right” for this project, that any other method of securing the components just makes the job harder.

Chapter 3 – Fabricating a Lens Cap

Factory-supplied slip-on lens caps are not suited to opening and closing with a servo motor, so you need to fabricate a flat lens cap and a mounting bracket to attach it to your servo.

After trying several methods (which I won't go into here), I discovered *3D printing is best.* A 3D-printed lens cap only 1/10" thick is lightweight, flat, and rigid, and a 3D-printed mounting bracket is far better in all aspects to one made from thin wood, as were my original brackets.

If you have access to a 3D printer, you can print your own lens cap. Click http://astronomy.mdodd.com/files/tlcc_3d_lens_caps.zip to download a ZIP file containing lens caps in various sizes. The files are in .stl format. Unzip an .stl file and open it in a slicer program, slice it for your printer, and save in .gcode format that a printer uses to print objects. The ZIP file also contains files for at least two mounting brackets with slightly different size holes for the servo shaft.

If you don't have a 3D printer, contact me to buy a lens cap in the size you need. My email address is on the title page of this manual. Or you can send a .stl file to a 3D printing service to print your cap.

The photos in this section illustrate a typical installation. I 3D-printed the sample cap in yellow and the mounting bracket in red to make it easy to distinguish them.

The cap

The goal is to have a lightweight flat cap that will close snugly against your telescope's objective lens ring or rigid dew shield. My 3D-printed caps are 0.08" or 0.10" thick, weigh little, and are rigid.

The photo on the title page shows a 7-inch 3D-printed cap installed on a 130mm refractor's dew shield. The servo is glued to the top-center of the dew shield, and the mounting bracket is glued to the cap. The top of the bracket is attached to the servo's shaft. (Barely visible in that photo is a gray piece originally used to attach the bracket to the servo. This piece was later incorporated into the included mounting bracket – simplifying installation and adjustment.)

Lens cap mounting bracket

Figure 11: Stock servo servo horns are not included with the TLCC

Servos typically are sold with plastic fittings known as “horns” that fit onto a servo’s spline shaft to transfer motion to a device, as seen in Figure 11. These arms aren’t suitable for the TLCC because the servo is installed on its side, and the extra legs will hit the telescope and prevent the servo from operating. *I do not include plastic horns with the* TLCC.

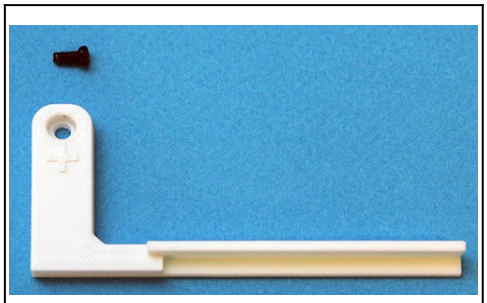


Figure 12: 3D-printed mounting bracket and screw

Instead, I designed and 3D-printed a lens cap mounting bracket to attach to the servo’s shaft, Figure 12. The bracket’s long leg has a flange to glue or screw to the lens cap. The short leg attaches to the servo shaft with a screw.

The bracket’s hole doesn’t have splines, but pressing it onto the servo’s spline shaft cuts shallow grooves in the hole. A machine screw holds it securely in place.

Servo shafts are not always exactly the same size, and 3D-printed holes can vary slightly as well. When testing your system, I selected a bracket that fits your servo. The bracket might have a **+** or **–** symbol on it to tell me the hole size, as seen in Figure 13.

When installing the bracket on the servo, press it onto the shaft, then use the machine screw to draw it down a bit more for a tight fit. *Do not tighten the screw yet – wait until you’ve mounted the servo and lens cap on your telescope (next chapter).*

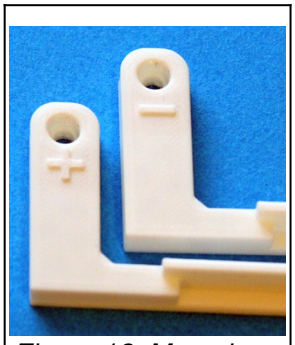


Figure 13: Mounting racket shaft hole size markings

Chapter 4 – installing the TLCC on Your Telescope

This chapter explains how to how to install the servo and lens cap, then the stack, on your telescope. You will need:

- A cloth to place under the stack while installing the servo and lens cap.
- An AC receptacle for the stack power adapter.
- A WiFi router or access point for the Arduino to connect to.
- A computer with the App on it, and connected to the same WiFi network as the Arduino.
- Painter's tape.
- Goop. ☺

Where to put the stack

The servo and lens cap will be installed at the objective end of the telescope, and the stack must be located between there and the power cable that plugs into it. The stack needs these connections:

- Three-wire cable from Each servo to the servo shield. The servo's cable measures about 11 inches long, and I provide a 12-inch extension cable.
- Cable from the power adapter to the servo shield. The power adapter must plug into a 120VAC receptacle.
- You do *not* need a USB cable plugged into the Arduino. That's only for uploading a program to it.

The fundamental question is, where will you place the stack? If you place it *on* the telescope, you'll need to run the power to it, with slack for unimpeded movement, just as with cables to a camera, anti-dew heater, or other equipment on the scope.

The power cable can be shorter if you place the stack *off* the telescope, but then you might need a longer servo cable. You can buy extension cables on eBay or through online radio-control hobby retailers.

Note: The servo cable carries a control signal to the servo that could be degraded if the cable is too long. I don't know what length is “too long,” but radio-control enthusiasts seem to believe that 30 inches is okay. If your servo behaves erratically, reduce the cable length to see if that makes it work correctly.

I placed my stack on the dovetail plate that holds the telescope, about midway along its length. Velcro with self-adhesive backing holds it securely. I bundled the power cable with the other imaging equipment cables.

Do this:

- Identify a location for the servo on top of the telescope (see Figure 15) and near the objective end, where the lens cap will be. Use a cloth to clean this area.
- Connect the servo cable and power cable to the stack.
- Set the stack on a cloth (to electrically insulate it from metal telescope parts) in its planned location.

Checking servo motion

With the stack resting on a cloth, do this:

- Power-up the stack.
- Launch the App on your PC, and connect to the stack. The App should indicate the cap is closed. If not, click the **Close Caps** button to rotate the servo to the closed position.
- Press the mounting bracket onto the servo shaft with the short leg parallel to the telescope and extending in the direction the telescope is “looking.” The long leg should extend downward.

Installing the servo and lens cap

Follow the instructions in the table on the next page to install the lens cap, servo, and bracket.

Look at Figure 15 to understand where the servo will be mounted on your telescope relative to the lens cap. Looking from the front of the scope, the mounting bracket is on the *left* side of the servo, and the long leg is pointing downward over the lens cap. Now attach the bracket to the servo as instructed below.

Use three pieces of painter's tape to hold the lens cap against the end of the telescope, with the edge evenly spaced all around.

Figure 14: Lens cap taped to telescope

For simplicity and ease of installation, the servo should sit flat on the top-center of the telescope. This means *the mounting bracket will be offset to the left from the center of the lens cap,* illustrated here.

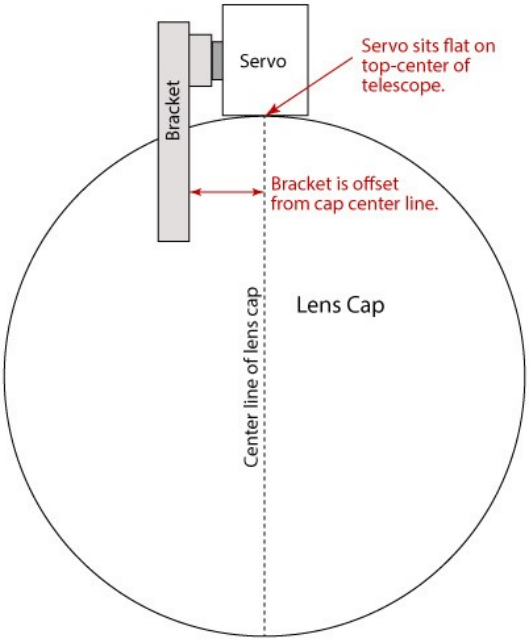


Figure 15: Servo, bracket, and cap mounting arrangement

We need to check that the servo rotates the bracket correctly when commanded to open and close the cap. Do this:

- Set or tape the lens cap onto the objective end of your telescope, as seen in Figure 15.
- Connect the servo cable and power cable to the stack.
- Set the stack on a cloth (to electrically insulate it from metal telescope parts) in its planned location.
- Power-up the stack.
- Launch the App on your PC, and connect to the stack. The App should indicate the cap is closed. If not, click the **Close Caps** button to rotate the servo shaft to the closed position.
- Hold the servo above the telescope, and press the mounting bracket onto the servo's shaft, pointing downward in front of the lens cap.
- In the App, click the **Open Caps** button to rotate the servo to the open position, as you did during initial checkout in Connecting and testing the system. The servo should rotate the bracket *upward and away from the telescope.*
- Click the **Close Caps** button. The bracket should rotate back to its original position – downward and parallel to the lens cap.
- *Leave the stack powered-on, and the servo energized in the closed position.*

This next step adjusts the servo's fore-aft position on the telescope so the mounting bracket rests flat against the lens cap.

Hold the servo in its designated position atop the telescope with the bracket pointing downward over the lens cap.

- If there is a gap between the **bottom** of the bracket and the lens cap (Figure 16), slide the servo **forward** and rotate the bracket until it rests flat against the cap.
- If there is a gap between the **top** of the bracket and the lens cap (Figure 17), slide the servo **backward** and rotate the bracket until it rests flat against the cap.
- Once you have positioned the servo fore-and-aft so there's no gap between the bracket and the cap, as seen in Figure 18, you might want to tape it in place, but you will be removing it in the next step to apply Goop to it and the bracket.

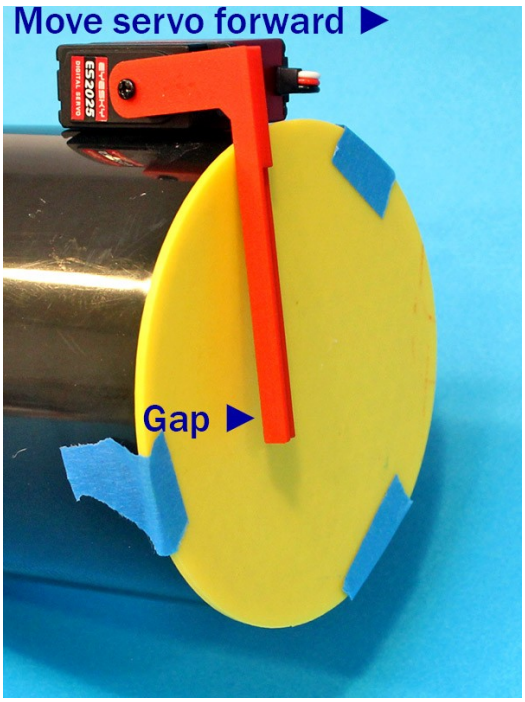


*Figure 16: Move servo **forward** to remove **bottom** gap*

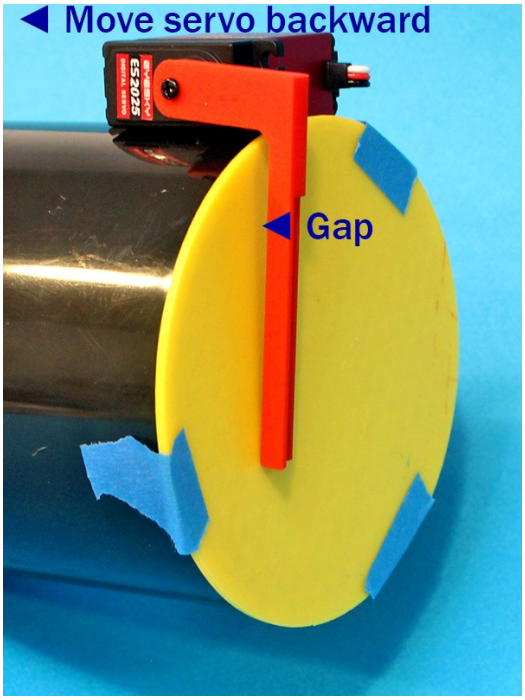


*Figure 17: Move servo **backward** to remove **top** gap*

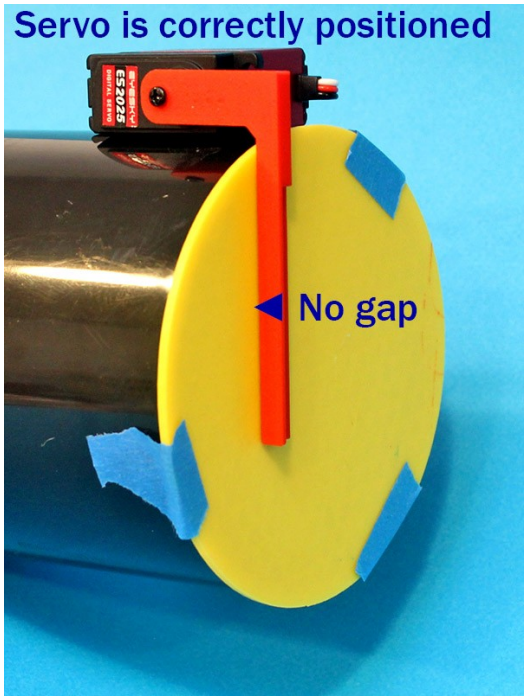


Figure 18: Servo correctly positioned with no gap

Lookin' good!

You're doing great! It's time to glue it all together.

At this moment, the lens cap is taped to the front of the telescope and the servo is resting on top of the telescope The servo is energized in the closed position. The bracket is on the servo shaft, and there's no gap between it and the lens cap.

Before performing these next steps, read through them to understand what's involved.

• Ensure that the lens cap is well-centered over the telescope's lens opening. If not, loosen the tape, adjust the cap, and secure the tape again.

• Peel up any tape holding the servo to the telescope. Lift off the servo with the attached mounting bracket.

• Apply a thin layer of Goop to the mounting bracket flange that presses against the lens cap.

• Apply two ¼"-diameter "blobs" of Goop on the bottom side of the servo, one near each end.

• Place the servo in its designated spot, and gently press down to spread the Goop. The center of the servo should rest on the scope, with Goop spreading outward toward the sides, where the telescope curves down.

• Press the bracket against the lens cap still taped to the telescope to tweak the servo's final position.

• Tape the bracket to the lens cap until the Goop cures. There is no need to tape the servo, but you may do so if you wish.

• ***Walk away.*** As before, leave the stack powered-up with the servo energized in the closed position while the Goop cures. Come back in 4-6 hours, after the Goop has begun to cure.

After 4-6 hours, install and tighten the screw holding the bracket to the servo shaft, and remove all tape holding the lens cap and bracket to the telescope.

That's it – you're done! You have fabricated a lens cap and bracket, attached them to the servo, and attached the servo to your telescope. Time to see it work.

First test – open the lens cap

First be sure no tape is holding the lens cap to the telescope. With the stack still powered-on, and use the App to connect to it. Click the Open Caps button. The servo should open the cap smoothly.

Final test – close the lens cap

Click the Close Caps button in the App. The servo should close the cap smoothly. If the cap does not close correctly, Loosen the bracket screw and rotate the bracket so the cap is snug against the telescope. Tighten the screw.

The only other option is to remove the servo from the telescope and the bracket from the lens cap, and repeat the steps above to achieve a better closed fit. But fear not! This situation is unlikely if you performed the steps above carefully. But if you need to try again, rest assured that removing parts glued with Goop is very easy if done within 4-6 hours.

First remove the mounting bracket from the servo shaft, and take the cap and bracket to a table or workbench. Use a screwdriver or putty knife to pry the bracket off the lens cap. Grasp a string of Goop, and peel the Goop off both surfaces. Use a thumb to ball-up the Goop until all is removed.

Rock the servo gently side-to-side on the telescope to stretch the Goop. When loose, remove it from the telescope, then remove the Goop as described above.

Use the command program to ensure the servo is in the closed position. Repeat the steps above to locate the servo on the telescope so there’s no gap between the bracket and the lens cap.

Glue the bracket and servo in place, wait 4-6 hours, then test again.

Final installation

With the system operating correctly, remove the cloth from under the stack, and permanently install it in your chosen location. Be sure to route the cables so they won't snag during telescope

movement. Also leave sufficient slack so the cables don't pull on the stack. Adhesive-backed Velcro does a good job of holding the stack in place, and makes it easy to remove later. A couple dabs of Goop also will work, but removing the stack is not as easy.

Goop to the rescue

Not only is Goop is handy for installing the lens cap bracket and servo, but also for securing connections to the stack:

- I recommend using a tiny drop on the servo cable plug to the servo board. It's unlikely this connection will come loose, but better safe than sorry.
- I also recommend applying a small smear of Goop where a servo cable plugs into an extension cable, to keep that connection from pulling apart.
- Goop is very useful when installing the optional temperature/humidity sensor; see below.

As a general practice I apply Goop to *all* cables plugged into my imaging equipment. I've had USB and Ethernet connections fail; USB connectors especially are notorious for this. If a connection isn't screwed together, it gets Goop.

Installing the optional temperature/humidity sensor

The Arduino can read a sensor and transmit the ambient temperature and humidity in the observatory to the App which displays it at the bottom of its window.

I supply the sensor (Figure 19) installed on a 3D-printed plastic arm to make it easy to mount near the stack. In addition, I apply colored tubing to the cable pins, and paint their mating socket header locations on the servo shield.

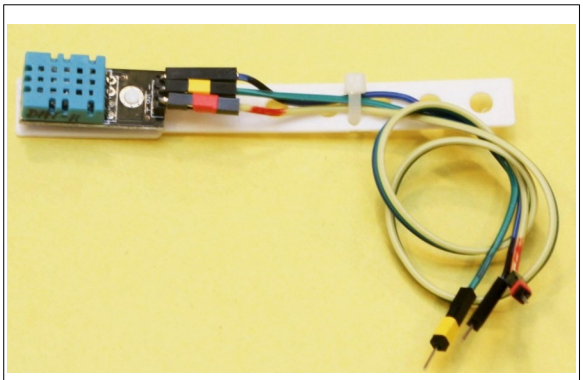


Figure 19: Sensor on plastic arm

Note: the sensor in Figure 19 is mounted on a small circuit board to which the cables plug. Some sensors don't have this board, so they are attached directly to the plastic arm, and the cables plug directly to the sensor pins.

Find a suitable location for the sensor near the stack, then plug the three pins on the sensor wires into the three header sockets with paint colors that match the tubing colors. See Figure 20. ***Important!*** Ignore the wire colors and match the tubing colors.

- The red pin plugs into the socket labeled **5V** on the **X4** header, and the black pin plugs into one of the sockets labeled **GND**.

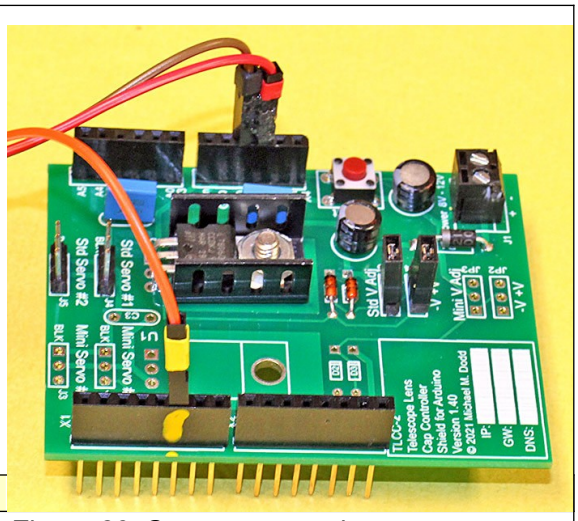


Figure 20: Sensor connections

- The red and black pins are bonded together with Goop because they always plug into adjacent pins on the **X4** header. *Do not reverse these pins!*
- The wire with the yellow pin plugs into the socket labeled **5~** on the **X1** header, located on the opposite side of the servo shield.

Testing the sensor

To test, power-up the stack and launch the App. Click the green **Connect** button on the App. The window should look like Figure 6. You should see **Connected** and the Arduino firmware version should be displayed.

Now click the check-box next to the words **Observatory Temperature**. A red progress bar steps across the screen (Figure 21.) When it reaches the right end, the current temperature and humidity are displayed. This repeats as long as the box is checked. When you un-check the box, the last values remain, but are dimmed.

Reading the sensor consumes time, and can slightly slow response to open/close commands, so un-check the box to turn off the sensor when you don't need that information.

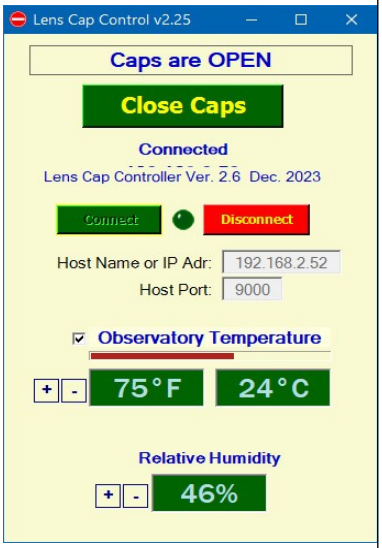


Figure 21: The App displaying temperature and humidity

Calibrating the sensor

Notice that [**+**] and [**-**] buttons appear next to the temperature and humidity displays when the box is checked. These adjust the values read from the sensor so they're accurate. Here's now to do that.

- Place the stack and the sensor on a table.
- Place a known-accurate thermometer (and hygrometer if you have one) near the sensor. An electronic weather station with an indoor sensor works well.
- Wait several minutes for the sensor and the instruments to stabilize to ambient temperature and humidity.
- On the App, check the box to activate the sensor displays.
- Compare the readings with your instruments. If they are different, click the [**+**] button to increase the displayed value, and click the [**-**] button to decrease it. **Note:** the readings won't change until the red progress bar reached its right end.
- Each click of a **[+]** or **[-]** button changes the sensor data by only 1%, so the reading might not change with every click. Be patient and wait for the next reading to appear.
- The Arduino stores the calibration values in non-volatile memory so you need to calibrate the sensor only once.

Final sensor installation

The plastic arm has several holes you can use to mount it near the stack, but I recommend using Goop to glue it instead. Mine is glued to the top of the OTA. Select a location close enough for the wires to reach the servo shield, but with the sensor projecting away from the telescope equipment.

Tape the arm in place while the Goop dries, then remove the tape, and the sensor is ready for operation.

Chapter 5 – Operating Instructions

This chapter brings together the operating instructions spread throughout the manual. You've done everything before, but sometimes it helps to have the instructions in one place.

Power-on

Turn on power to the stack when you are ready to begin observing. The Arduino remembers the last lens cap position before power is shut off, and commands it to that position on power-up (no movement unless it was physically moved since the last use).

Generally, *you should never move the lens cap manually when power is off.* Doing so could damage the servo's gears. Always use the App to move the lens cap.

The App

With the stack powered-up, launch the App and connect to the stack. Refer back to page 7 for details. The flashing green light near the center of the window indicates the program is communicating with the stack on the telescope.

Can't connect to the stack

If nothing happens when you click the green **Connect** button, perhaps there's a problem with your network, or the stack isn't powered-on or has a problem. After about 20 seconds an error message will appear. Click the OK button, then track down and fix the problem.

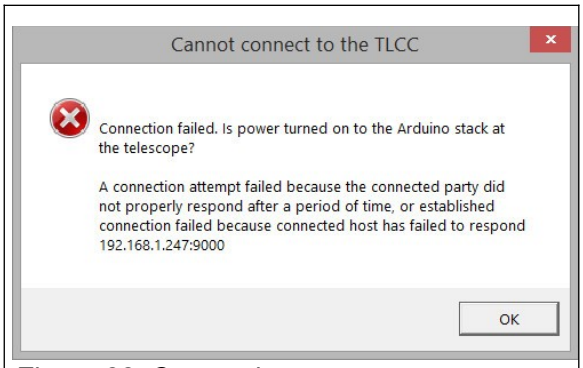


Figure 22: Connection error message

Note: The App does not respond to the keyboard or mouse clicks while it is trying to connect. If you wildly click on its window while waiting for the connection, eventually you'll get a Windows message that says the program is not responding. Close the App and try again, this time waiting the full 20 seconds for the “cannot connect” message or a good connection.

Opening and closing the lens cap

The App displays the current lens cap open/closed state at the top, and the large green button's legend changes accordingly between **Open Caps** and **Close Caps**. When you click that button, the App displays **Caps are MOVING** until it receives a different status from the stack.

Important: The servo does not provide position information to the Arduino, so the Arduino does not really *know* if the lens cap is open or closed – it only knows where the cap *should be.*

It is good practice to use a network-connected camera to look at the telescope to verify that the cap is where you expect it to be. I use a security camera feeding a Web browser to view my observatory's interior (right). The white caps make it easy to see their positions.

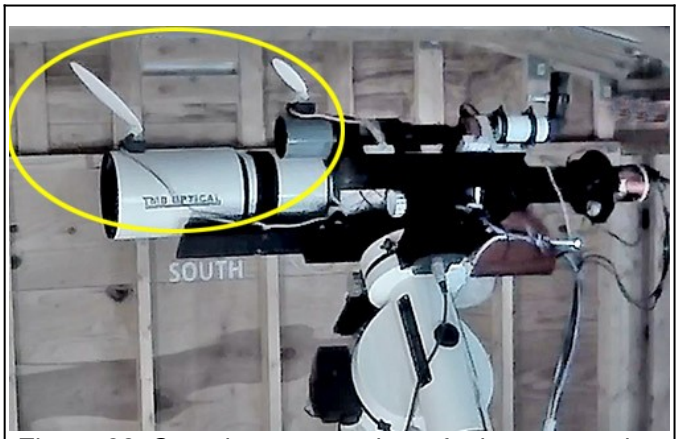


Figure 23: Security camera view of telescope and lens caps

Reading the temperature and humidity

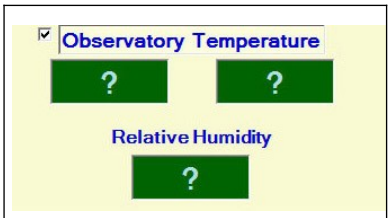


Figure 24: No sensor data

Click the check-box next to the words **Observatory Temperature** to read the senor in the observatory. The ambient temperature and relative humidity appear at the bottom of the window (Figure 21).

If the sensor is defective or not connected to the servo shield, the three fields display question marks (image, left).

Disconnecting the stack

Click the red **Disconnect** button to disconnect the App from the stack.

Note: To avoid possible network problems, I disconnect after opening the lens caps at the start of a session, then connect again hours later when it's time to close them. Even while disconnected from the PC, the stack continues to energize the servo to hold the lens cap in its last-commanded position.

Remember to turn off power to the stack when you're done using it.

The App automatically disconnects when you close it, even if you don't click the **Disconnect** button first.

Chapter 6 – Technical Information

This chapter contains information about the TLCC's servo shield, the App, and the Arduino firmware. It also explains how you can adjust the servo power voltage if necessary, and customize the Arduino firmware to meet specific requirements.

The flash drive contains the executable App and the Arduino firmware source code to upload to your Arduino. The CD contains three folders:

- **Control Program (the App)** contains the executable control program and a configuration file that is read each time you launch the App, and written each time you close the program.
- **TLCC** contains the Arduino firmware source code. You can customize this if needed, and upload it to your Arduino Uno.
- **User Manual** contains this user manual as a PDF file.
- There's also a **readme.txt** file you can open in Notepad if you're totally lost. ☺

About servo voltage

The servo shield has two voltage regulators, one intended to power the high-torque servos used by the TLCC, and a second that produces slightly lower voltage intended for *miniature* servos that aren't powerful enough to move telescope lens caps (but the servo shield could be used in other applications where miniature servos are suitable).

One might expect all high-torque servos to be the same, but this is not the case – some work better on slightly higher voltage, while others work better on lower voltage.

The servo shield has small shunts to adjust the servo voltage. Before shipping your system, I tested it and adjusted the voltage for reliable operation with *your* servo.

The servo shield also has two sets of servo connectors – standard and mini: **Std Servo #1** and **Std Servo #2** (J4, J5) supply a nominal 6 VDC, and **Mini Servo #1** and **Mini Servo #2** (J2, J3) provide a nominal 5 VDC. Normally a high-torque servo is plugged into a **Std** connector, but if a high-torque servo works better on a lower voltage, it must be plugged into a **Mini** connector. The only difference between the **Std** and **Mini** connectors is the voltage – the current capability is the same on both, so you can plug a high-torque servo into either.

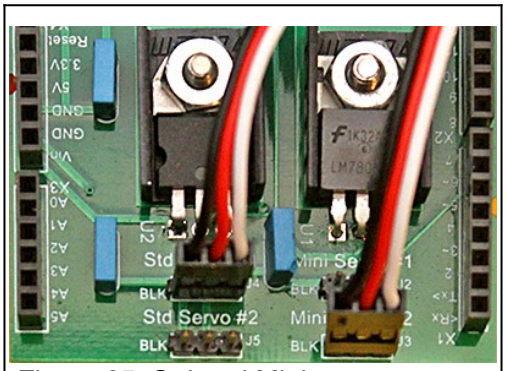


Figure 25: Std and Mini servo connectors

Note: During testing, I might have found that one or both of your servos work better in a Mini connector. I applied a tiny tag to each servo's cable to indicate where it was plugged (photo, right). This might help you assemble your TLCC system the same way I had it assembled.

The tag's reverse side has an empty rectangle so you can write your own connector ID if it's different from mine.

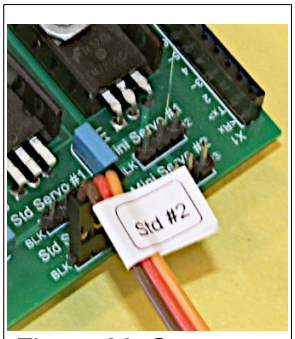


Figure 26: Servo cable tag

Adjusting servo voltage

Here is how to adjust the servo voltage if your servo is not operating properly.

There are two pairs of three-pin headers, one pair labeled **Std V Adj** (JP4, JP5) and the other pair labeled **Mini V Adj** (JP2, JP3). A black shunt (yellow arrows in Figure 27, right) is plugged onto two pins on each header. With both shunts plugged onto the -**V** pins on **Std V Adj**, the servo shield applies a nominal 6V to the **Std** servo connectors, J4 and J5. With both shunts plugged onto the -**V** pins on **Mini V Adj,** the servo shield applies a nominal 5V to the **Mini** servo connectors, J2 and J2

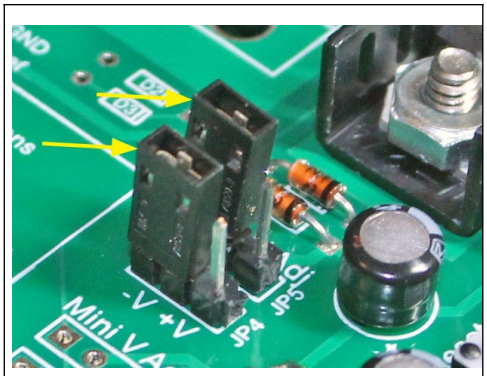


Figure 27: Voltage-adjust shunts

When a shunt is plugged onto the **+V** pins, it raises the nominal voltage by about 0.6V. Putting both shunts on the **+V** pins raises the servo voltage by about 1.2V. Thus, the voltage on the **Std Servo** connectors can be set to 6V, 6.6V, or 7.2V. The voltage on the **Mini Servo** connectors can be set to 5V, 5.6V, or 6.2V.

Follow the instructions below to adjust the servo voltage. Move the shunts on the **Std V Adj** headers if your servo is plugged into a **Std Servo** connector. Adjust the shunts on the **Mini V Adj** headers if your servo is plugged into a **Mini Servo** connector. The references to JP4 and JP5 refer to the **Std V Adj** headers. JP2 and JP3 in parentheses refer to the **Mini V Adj** headers.

1. Note where the shunts are plugged, e.g., to **-V** or **+V**.
2. Move both shunts to the **-V** pins, and test the servo. If the servo operates well, you're done; skip the following steps.
3. Move the shunt on JP4 (JP2) to **+V** to raise the servo voltage by 0.6V, then test the servo again. If the servo operates well, you're done; skip the following steps.
4. Move the shunt on JP5 (JP3) to **+V** to raise the servo voltage by 0.6V, then test the servo again. If the servo operates well, you're done; skip the following steps.

5. If the servo still does not operate well, connect it to the other type (**Std** or **Mini**) servo connector, and repeat steps 1-4 with the other set of headers and shunts.

Contact me if the servo does not operate reliably after trying **Std** and **Mini**. I will ask you to measure the voltage from pin 2 (+) to pin 1 (common) on the **Std** Servo and **Mini** Servo connectors.

A final note about servos

I've found that some servos don't operate well at low temperatures. Sometimes after an imaging session on a cold morning, I'll command the lens caps to close, and one does, but the other one doesn't move. Repeated open-close commands might make the reluctant servo respond, but not always. In this case, my solution is to wait until the ambient temperature rises, then try again. Usually the servos work properly with higher temperature.

The App

You should have copied the entire **TLCC Control Program** folder from the flash drive to your hard drive, following the instructions on page 4. You may copy the program to as many computers as you wish, but run only one at a time – attempting to simultaneously connect multiple Apps to the stack will cause unpredictable results.

Arduino firmware

The program code for the Arduino Uno is called a "sketch" (probably to make it sound friendly to non-programmer hobbyists).

Insert the TLCC flash drive into your computer drive and copy the **TLCC** folder to your C: drive. This folder contains the sketch for the TLCC Arduino Uno.

Uploading a sketch to the Arduino

To develop (or modify) and upload software to any Arduino microcontroller, we use the Arduino Integrated Development Environment ("IDE") which must be installed on your computer. It is available for Windows 10 and later, plus Mac OS X, and Linux. You can download it here: https://www.arduino.cc/en/software.

Before you attempt to modify the TLCC sketch, you should learn how to launch the Arduino IDE, connect to the Arduino Uno via a USB cable, open a sketch in the IDE, and upload it to the Arduino. Here are the basics:

- Unplug the servo shield from theArduino. For some reason, the Arduino will not accept a sketch upload from the IDE if the servo shield is attached.
- Plug a USB cable into your Arduino and into a USB port on your computer.
- Launch the Arduino IDE.

- On the IDE menu, click Tools → Port to identify the port the Arduino is connected to; you should see it listed by name. Click this port so the IDE can talk to your Arduino. (The IDE remembers the port for the current Arduino, so this step is necessary only when you plug in a new Arduino)
- It's best to start with a simple sketch to confirm that your Arduino can receive and execute a sketch. The IDE comes with one that flashes an onboard LED. Navigate to File -> Examples -> 01.Basics -> Blink to open that sketch in the IDE.
- Click the right-pointing arrow in a circle on the IDE's upper-left toolbar to upload the sketch to the Arduino.
- Lights on the Arduino flash rapidly while it receives the data, then it executes the sketch. One LED should flash repeatedly, 1 second on, 1 second off.
- Close the IDE. It is not needed once the sketch has been uploaded to the Arduino.
- Unplug the Arduino's USB cable, wait a few seconds, then plug it again. The Arduino's LED should flash repeatedly as before. The Arduino retains the sketch when power is off, and executes it every time power is applied.
- Now load the TLCC sketch into the IDE, and upload it to the Arduino. On the IDE menu, click File → Open, then navigate to the **C:\TLCC** folder, and select the sketch named **TLCC.ino**.
- Click the right-pointing arrow in a circle on the IDE's toolbar to upload the sketch to the Arduino.
- To confirm that the Arduino is executing the sketch, launch the App and connect to the Arduino. Confirm that the version information appears, and that the Arduino responds to open and close commands. The servo shield is removed, so the servo won't move, but the App will receive the open/close status information from the Arduino. But the Arduino can't send temperature/humidity data because the sensor plugs into the removed servo shield.
- You should perform the above test each time you upload the TLCC sketch to the Arduino.
- When done, close the IDE, unplug the USB cable from the Arduino, re-install the servo shield, power-up the system normally, and again confirm that everything works correctly.

Customizing the TLCC sketch

When you're ready to look at the TLCC sketch, perform the steps on page 27 to connect the Arduino to your PC and launch the IDE.

In the IDE, click File → Open, then navigate to the **C:\TLCC** folder, and select the sketch named **TLCC.ino**. You should see these lines at the top:

```
TLCC.ino
 *
  * Telescope motorized lens cap control for Ethernet and WiFi.
```

```
 *
 * Written by: Mike Dodd (mike@mdodd.com)
```

...and much more. If you look closer and scroll down about 40 lines, you'll see this like:

```
#include "tlcc_secrets.h"
```

You can edit **tlcc_secrets.h** in

Now click File → Open, then select the file named **tlcc_secrets.h**. This file is where you can customize network settings without modifying the actual TLCC sketch. Much of the file consists of comments, which are not part of the program code. A comment block begins with the characters /* and ends with the characters */. The Arduino IDE can be configured to display comments in a different color than normal program statements.

tlcc_secrets.h surrounds lines that may be customized with lines containing **<--- CONFIGURE --->** and **<--- END CONFIGURE --->>**

You may safely modify the lines between these CUSTOMIZE lines. Example:

```
// <--- CONFIGURE --->
#define SSID_1 {"my_home_net", "here i am"}
#define SSID_2 {"your_astro_net", "let me come in"}
#define SSID_3 {"x", "x"}     // Not used
// <--- END CONFIGURE --→
```

These three lines specify three WiFi SSIDs (one is not used). They should contain the SSIDs and passwords you provided to me when I built your TLCC. If they change later, make the changes to **tlcc_secrets.h**, and save the file.

Comment, don't delete! I strongly recommend you preserve the original values of every item you change. To do this, copy the entire line, then paste it directly under the original. Type two or three slashes at the start of the original line to turn it into a comment. I usually type *three* slashes to make my comments easier to search for.

For example, to change a WiFi SSID, copy and paste one line, comment the original, and modify the copy:

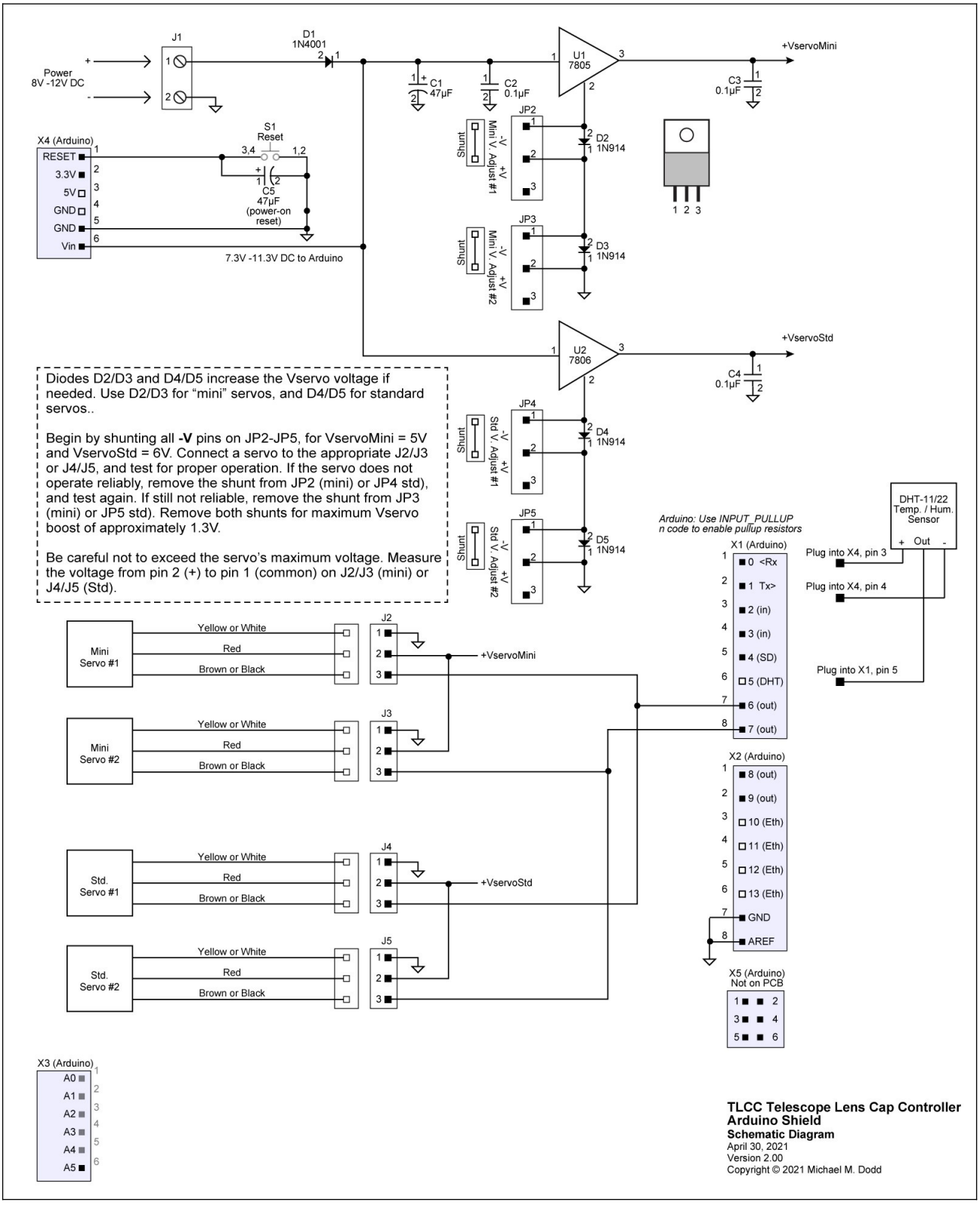
```
///#define SSID_1 {"my_home_net", "here i am"}
#define SSID_1 {"home_wifi", "password"}
```

When done with your customization, upload the sketch to the Arduino as instructed on page 27.

Remember to upload the sketch to the Arduino after making any change!

Servo Shield Schematic Diagram

The servo shield contains two voltage regulators, one for standard servos, the second for miniature servos, plus connectors for those servos. Two sets of shunts adjust the voltages produced by these regulators. The servo shield schematic is on the next page.


Power
8V -12V DC
J1
D1
1N4001
C1
47µF
C2
0.1µF
U1
7805
+VservoMini
C3
0.1µF
JP2
D2
1N914
Shunt
-V
Mini V. Adjust #1
+V
JP3
D3
1N914
Mini V. Adjust #2
1 2 3
X4 (Arduino)
RESET
3.3V
5V
GND
GND
Vin
S1
Reset
3,4
1,2
C5
47µF
(power-on
reset)
7.3V -11.3V DC to Arduino
U2
7806
+VservoStd
C4
0.1µF
JP4
D4
1N914
Std V. Adjust #1
JP5
D5
1N914
Std V. Adjust #2
Diodes D2/D3 and D4/D5 increase the Vservo voltage if needed. Use D2/D3 for “mini” servos, and D4/D5 for standard servos..
Begin by shunting all -V pins on JP2-JP5, for VservoMini = 5V and VservoStd = 6V. Connect a servo to the appropriate J2/J3 or J4/J5, and test for proper operation. If the servo does not operate reliably, remove the shunt from JP2 (mini) or JP4 std), and test again. If still not reliable, remove the shunt from JP3 (mini) or JP5 std). Remove both shunts for maximum Vservo boost of approximately 1.3V.
Be careful not to exceed the servo’s maximum voltage. Measure the voltage from pin 2 (+) to pin 1 (common) on J2/J3 (mini) or J4/J5 (Std).
Arduino: Use INPUT_PULLUP
n code to enable pullup resistors
X1 (Arduino)
0 <Rx
1 Tx>
2 (in)
3 (in)
4 (SD)
5 (DHT)
6 (out)
7 (out)
DHT-11/22
Temp. / Hum.
Sensor
+ Out -
Plug into X4, pin 3
Plug into X4, pin 4
Plug into X1, pin 5
Mini
Servo #1
Mini
Servo #2
Std.
Servo #1
Std.
Servo #2
Yellow or White
Red
Brown or Black
J2
J3
J4
J5
+VservoMini
+VservoStd
X2 (Arduino)
8 (out)
9 (out)
10 (Eth)
11 (Eth)
12 (Eth)
13 (Eth)
GND
AREF
X5 (Arduino)
Not on PCB
X3 (Arduino)
A0
A1
A2
A3
A4
A5
TLCC Telescope Lens Cap Controller
Arduino Shield
Schematic Diagram
April 30, 2021
Version 2.00
Copyright © 2021 Michael M. Dodd