

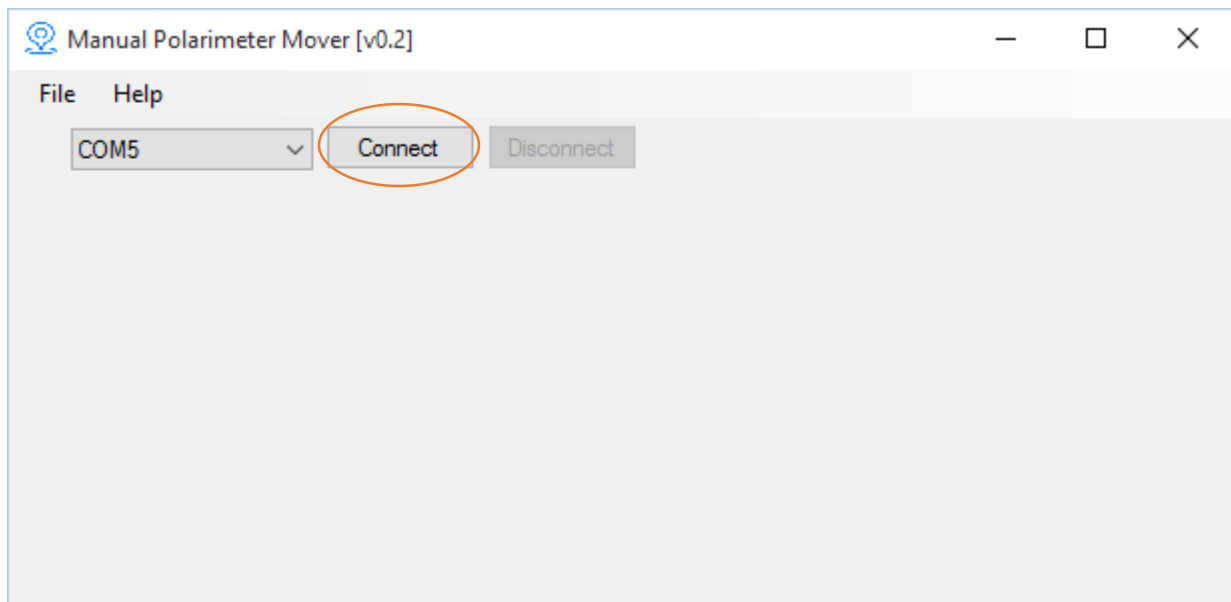
Interacting with the Manual Polarimetry Software

By Justin Moore | Feb. 4, 2016

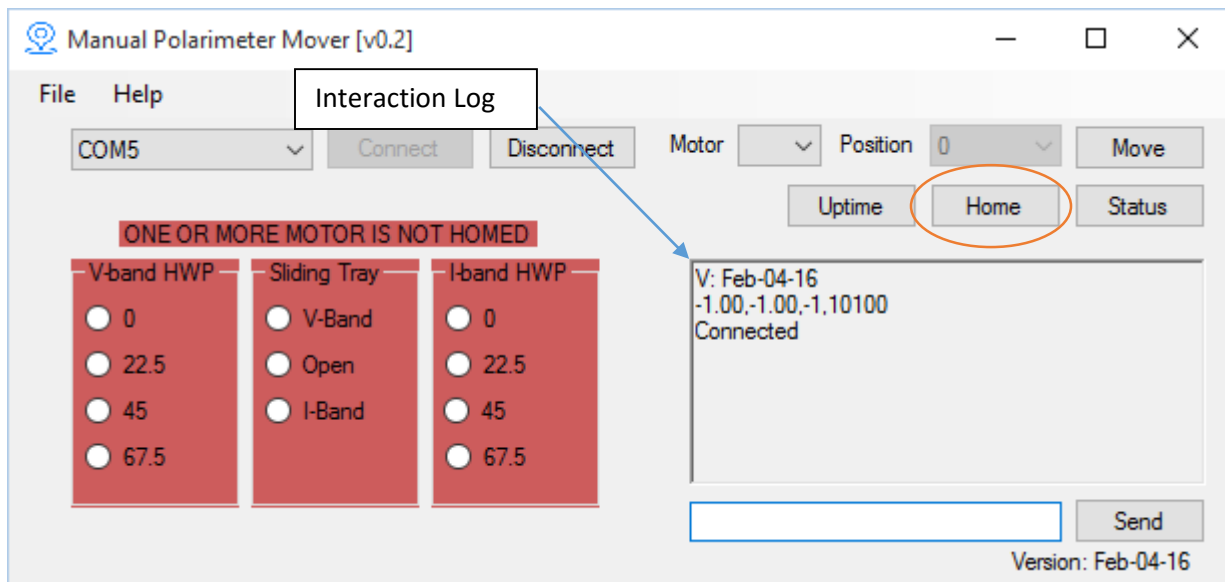
Open the Manual Polarimeter Mover C# application from the Desktop of PROMPT8.

Choose the **COM** port associated with the Arduino (Find this in Device Manager if you're not aware of it in advance)

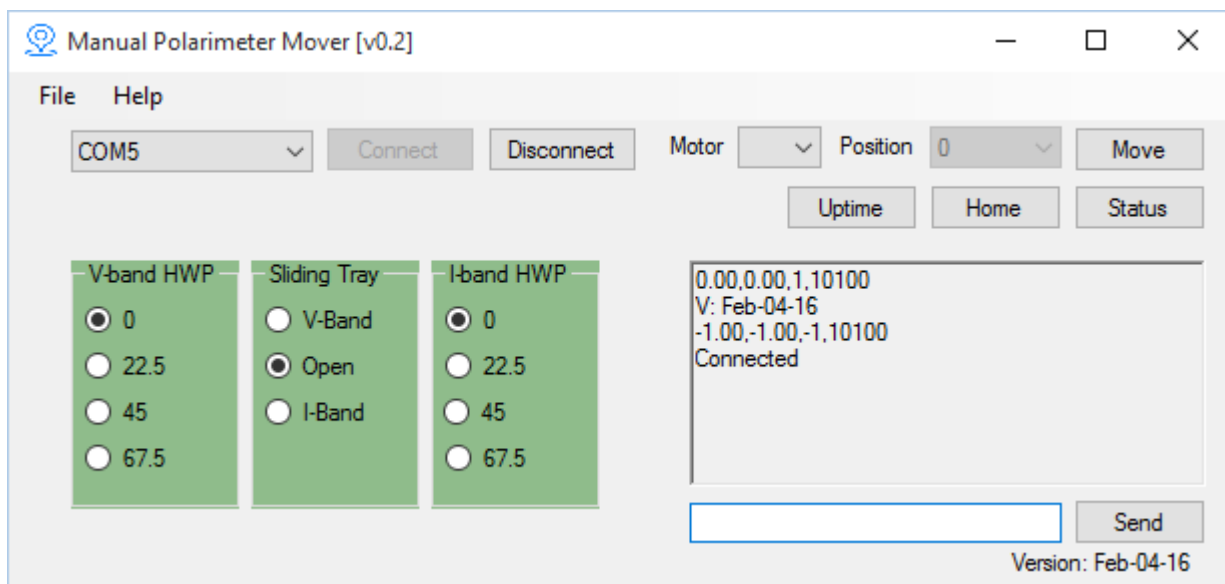
Click **Connect** to connect to the serial interface of the Arduino.



Once connected, if this is the first time, everything will be red. No motors are homed. If you try to move a motor, it will say you cannot because it has no position information. Status and uptime will return results but you can't do anything motor related before homing. Homing comes in a triplet, we have not implemented individual motor homing because it's not a standard use case. Homing one time should be enough (and a simple re-home of the HWP stepper motors after they've been released is accomplished quickly.) The general placement of the instrument will be with the tray resting in the Open position (triggering a limit), and therefore most of the time the tray should only need to be homed once.



If you're ready for the motors to be moved, and everything has been provided power, keep reading. The Home button will home all three motors sequentially. First rotating Stepper VHWP for homing V, then Stepper IHWP for homing I, and then finally homing the entire tray. **Do not click home unless you are ready for motor motion!** If you are ready, click **Home**.

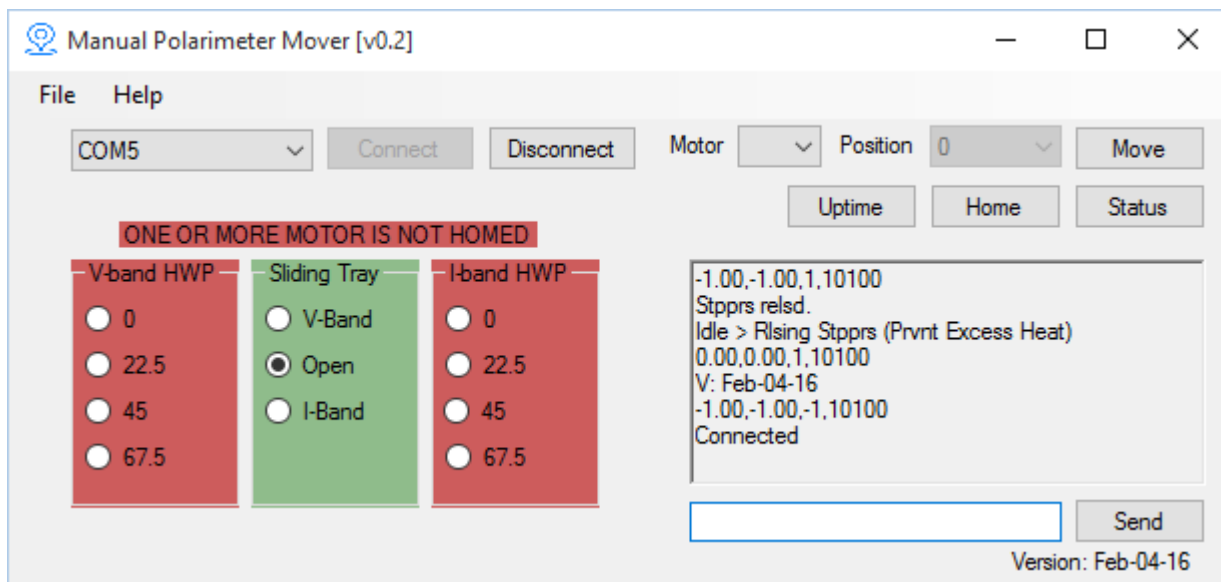


Now we are homed and everything ready to go. The limits will likely all be 1 when you first load as I have things set up for my test case here in the lab. 0 means the limit is triggered in our protocol. A green background is simply saying a limit is triggered. Since there are only 0.0 position limits on the HWP rotators, you will only see a green background at position 0. This is normal behavior! The limits are listed at the end of the status line, in the above image "10100" means LA_V is high, LA_Open is low [homed], LA_I is high, V_HWP is low [homed], and I_HWP is low [homed]. These limit indicators provide a quick look at physical states of the micro switches.

Commanding motion of the motors is as easy as choosing your motor from the **Motor** drop down box, then choosing a desired position from the **Position** dropdown and then clicking **Move**. This will lock the UI until the

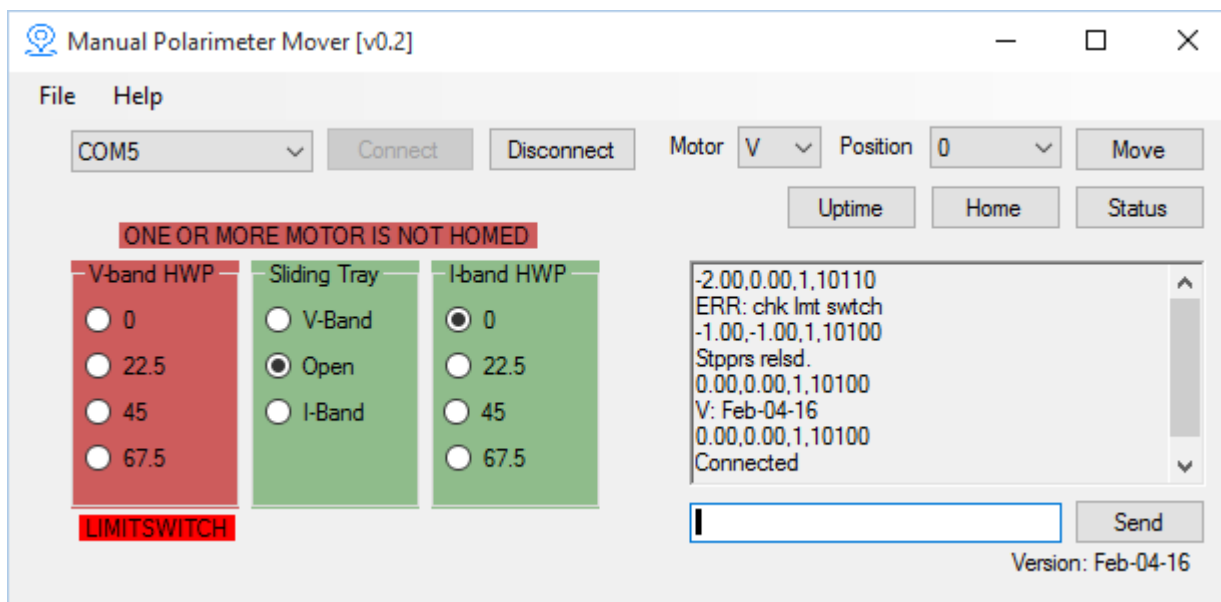
move is complete. If the move completes, then everything will unlock and you can interact again. Several error modes are built in to the software. I will list these error modes below:

- 1) Timeout – after 10 minutes of no activity, the stepper motors will be released. This is because in order for them to maintain their position – they must use energy, which generates heat. It turns out that the heat generated is pretty substantial. So we've implemented a power save feature that also saves us from problems created by this heat. The UI will update like below if this inactivity timeout happens:

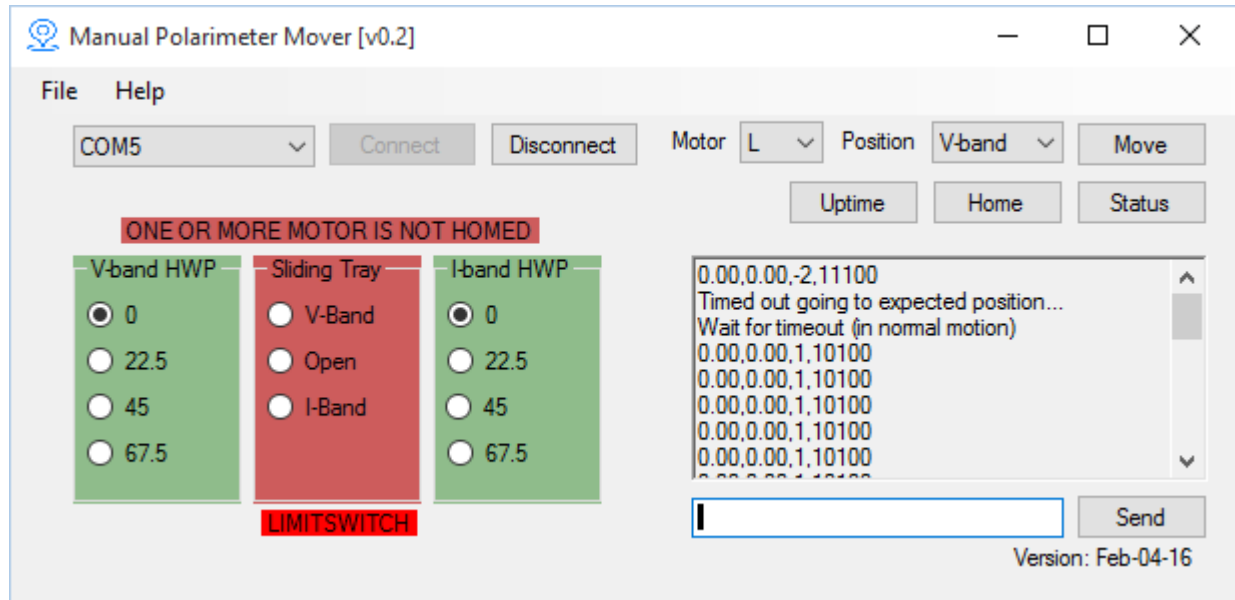


Homing fixes this problem and re-enables the stepper motors. The steppers will move one step forward and then backward if the release happened while the home limit was triggered to re-engage the stepper magnets.

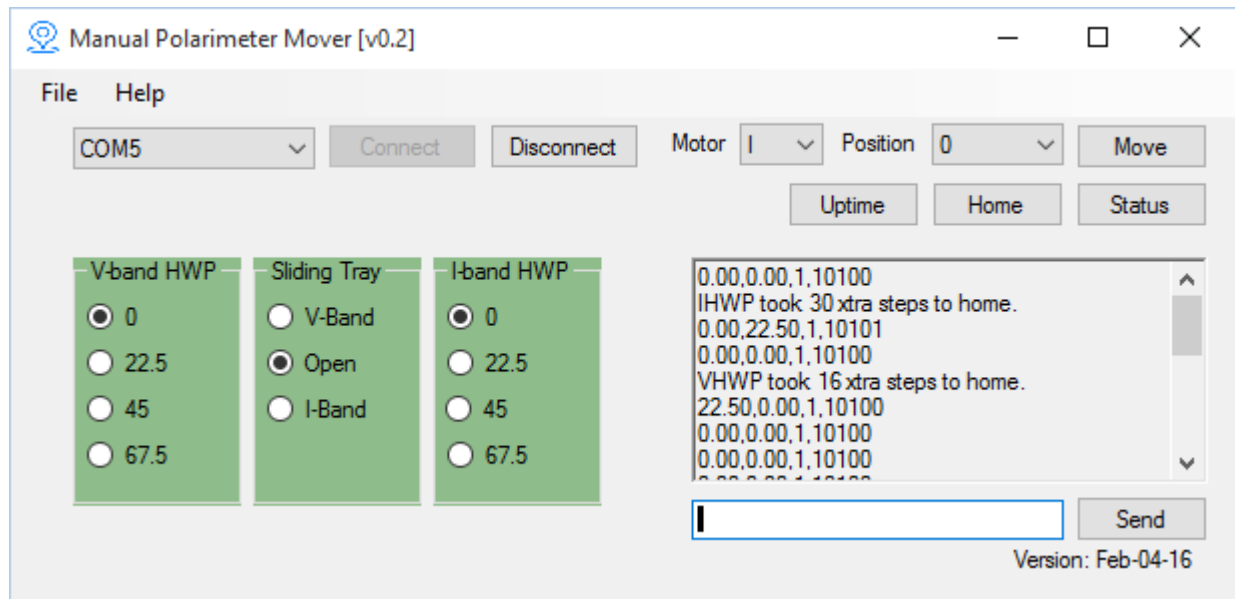
- 2) On homing, the HWP gear can only go around one time without triggering the limit before throwing an error. If this is exceeded, the limit switch problem will light up and the known position box will become red. This should only happen if something is physically wrong with triggering the limit and it may mean a hands on fix is required. I've demoed this with V-band HWP below.



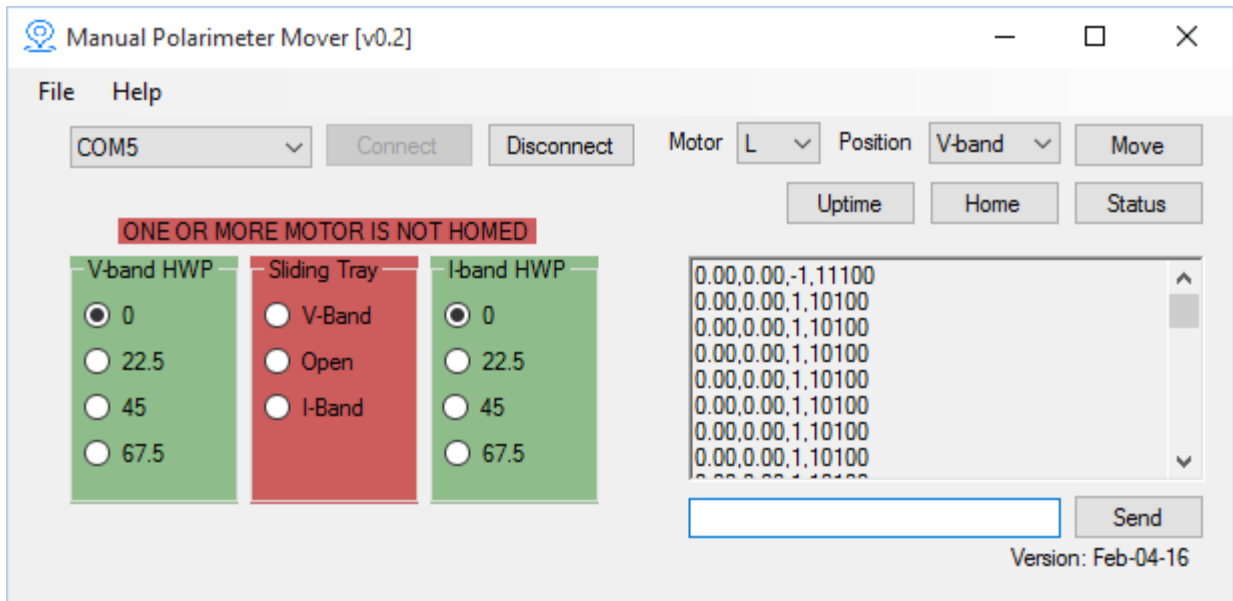
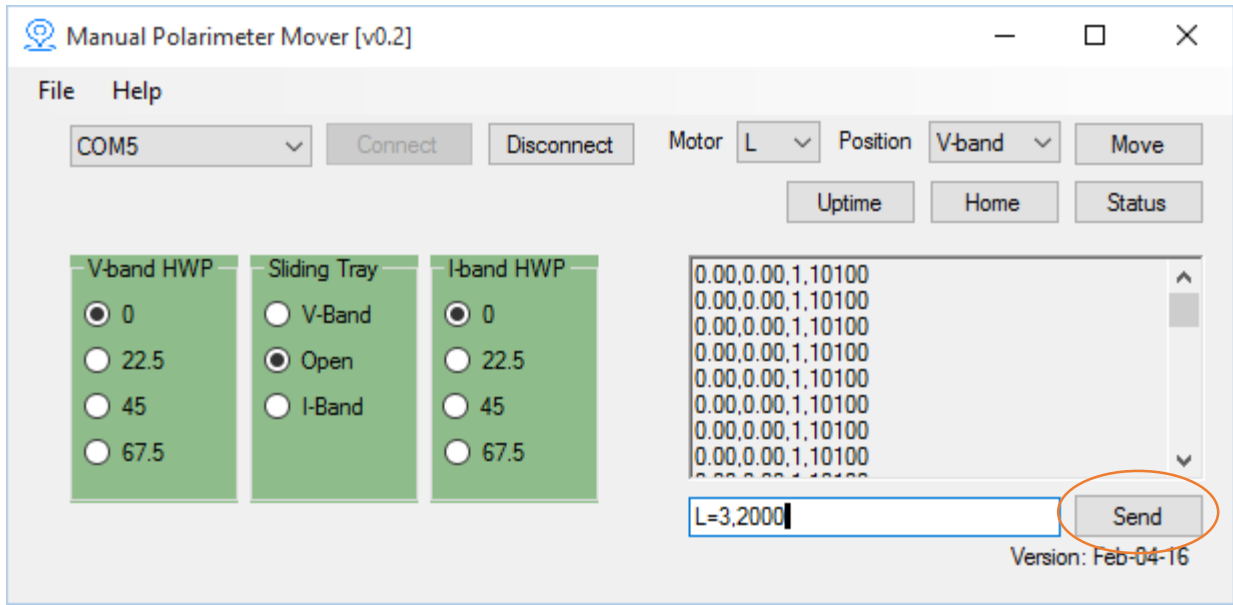
- 3) If the tray is moving longer than 15 seconds without triggering a limit, it will also throw an error. This can happen in normal motion or in homing. In the case of homing, we have developed a technique for switching the direction of the motor after the first timeout in case somehow the tray made its way outside of our limits and into the land of hard limits. If it times out twice on homing, something is very wrong.



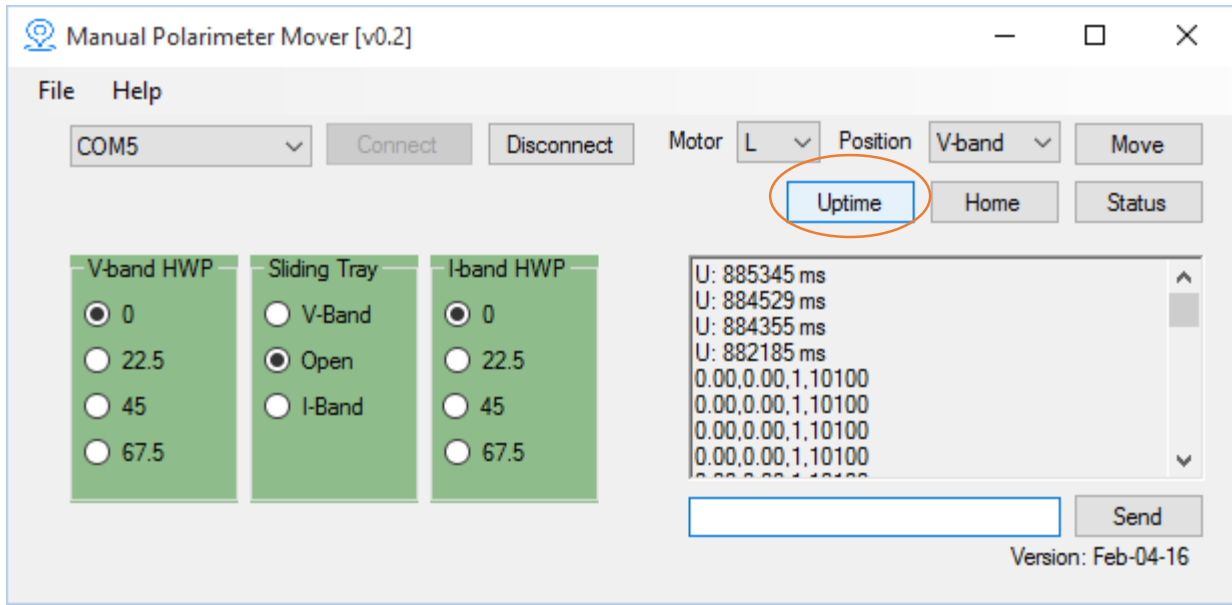
- 4) If you command the stepper motors to the 0.0 position, and it does not hit the homing limit, it could indicate that some steps were missed in the back and forth process. The UI will warn the user of this, and fix it accordingly. If this happens consistently, it could mean the limit is not triggering properly and the trigger mechanism or the limit itself may need to be replaced.



- 5) If you use the nudge functionality the tray and move it off of a limit, we no longer know where it is in space. Therefore its current position will be set to -1 and it will need to be homed again. The command below is **L=3,2000** or move the tray right for 2 seconds. Click **Send** to send the custom command.

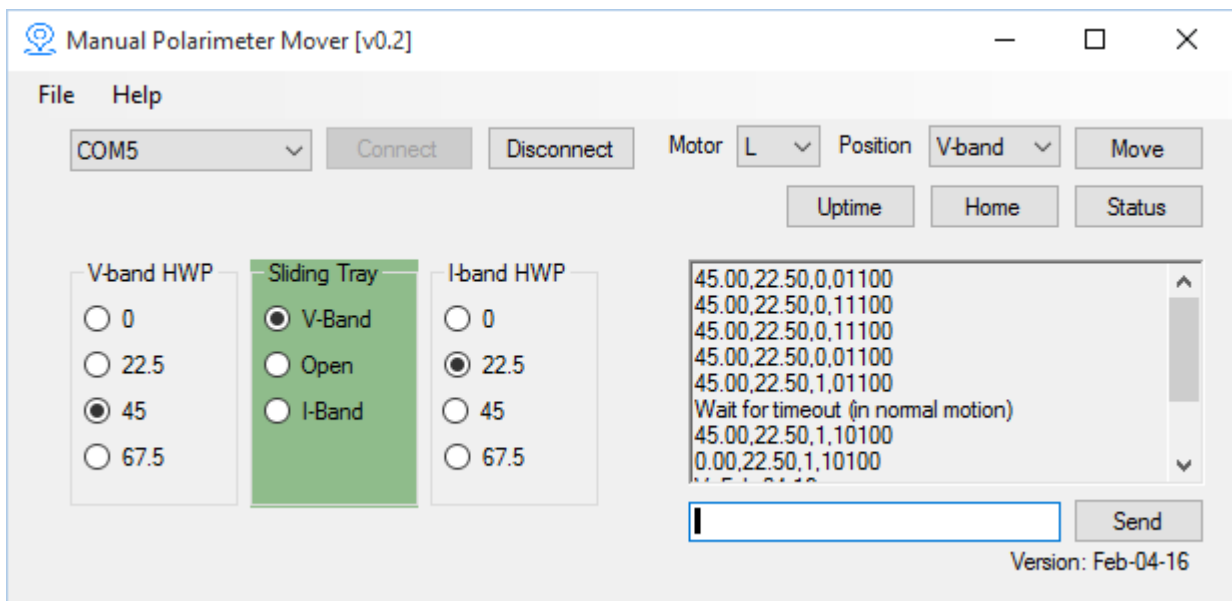


- 6) In general the convention is: current positions of -1 are the result of nudges, current positions of -2 are the result of time out errors during motion.
- 7) **Uptime** is a feature used to see how long the Arduino has been running without being reset. This is very useful when debugging, as the Arduino reboots so fast that it can be easy to miss that it has reset.



Finally, note that the state of the system is maintained even if the C# GUI is not connected to the Arduino. So if you click **Connect** for the first time and see that the steppers are released, this is normal behavior, if it has been sitting idle for a long period of time. Click **Uptime** to check how long it has been running.

The following just shows another normal use case. Don't read into the limits in the status information because I've used a jumper on the home limits of the HWP in this test, normally they would be 1 if the position is not 0.0.



That should be everything! Email with any questions. jpm@unc.edu