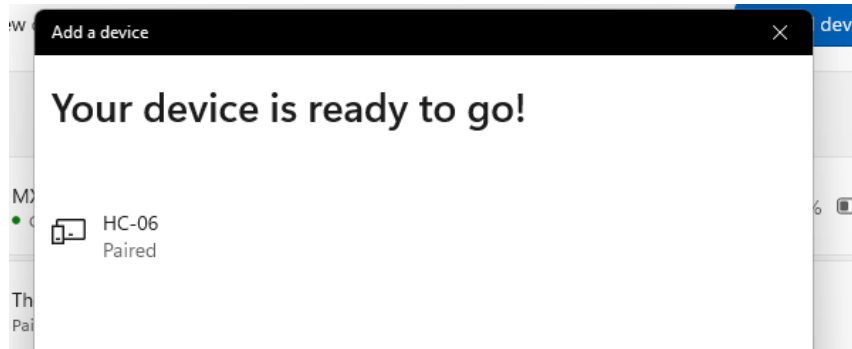
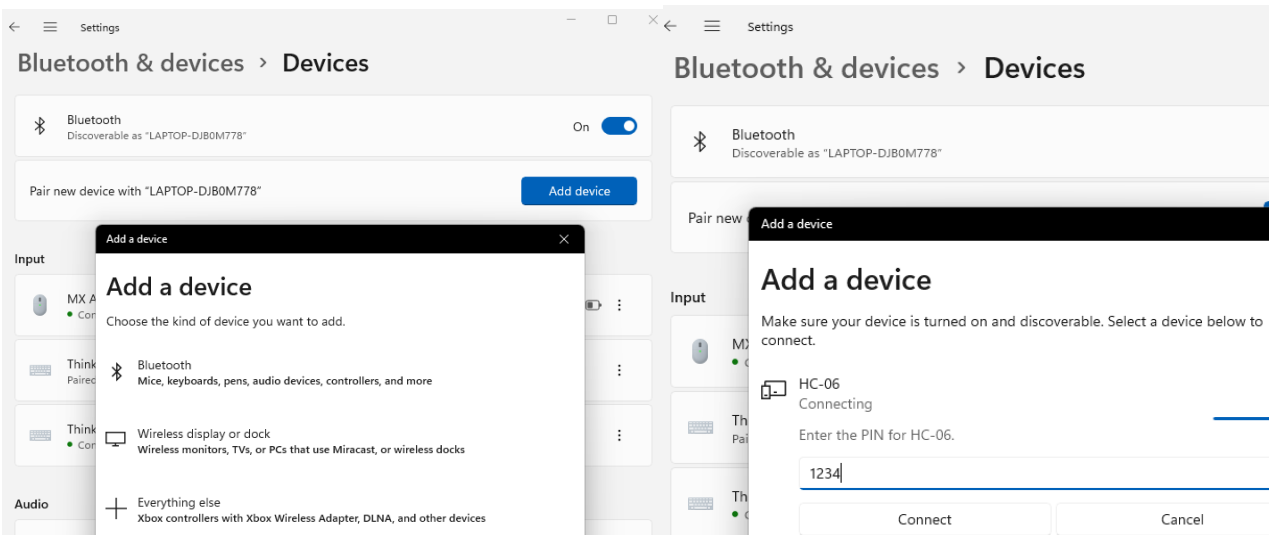
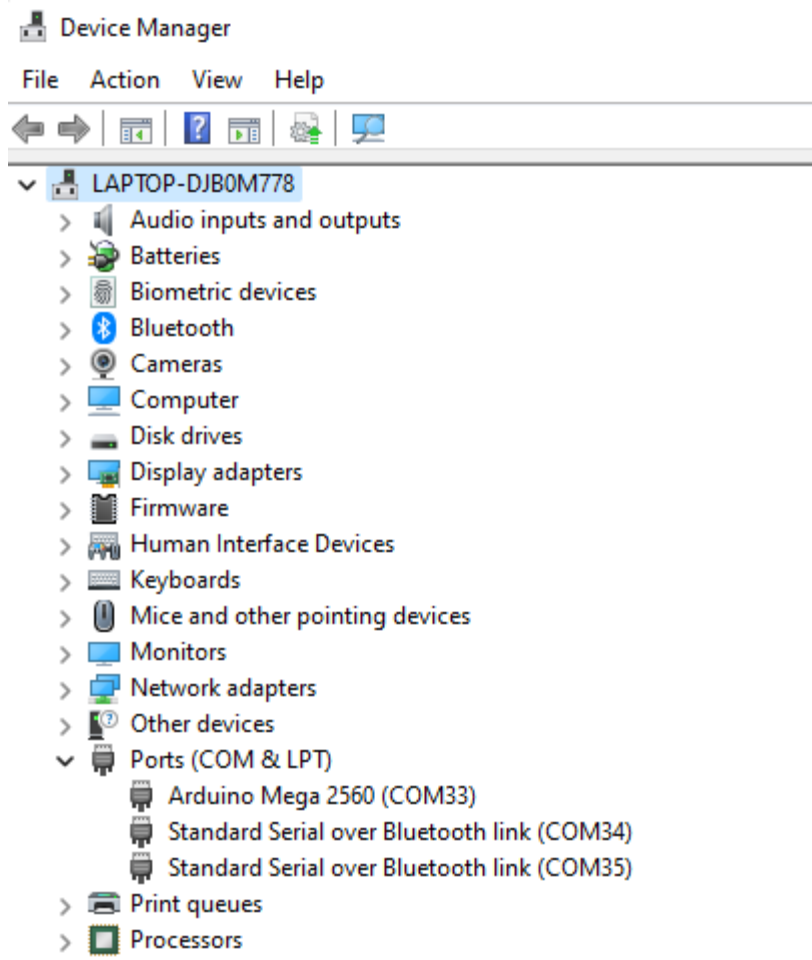


Pairing and setting up the Bluetooth:

1. Plug in the Bluetooth (BT), make sure the board is powered. There should be flashing red light on the BT.
2. In windows Bluetooth menu choose, Add device, Bluetooth. You should see the module show up as HC-06, enter the password 1234, click Connect.

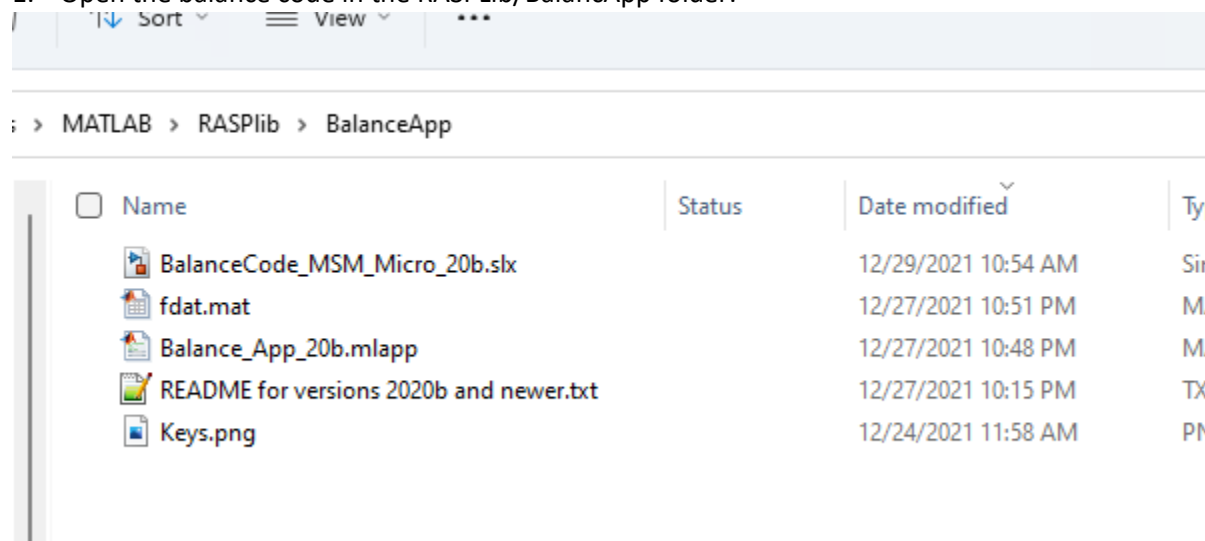


3. In the device manager it should add two Bluetooth devices. You will typically want to connect to the smaller number, in this case COM34.



Setting up and downloading the balancing code with the remote control App

1. Open the balance code in the RASPLib/BalancApp folder:



2. Make sure the COM port is the port for the Bluetooth (34). Make sure the Arduino Serial corresponds to where your Bluetooth module is plugged in:

- MinSegMega – plugged in on the right side near the potentiometer – Serial 3
- MinSegMega – plugged in the center port towards the top – Serial 2
- M2V5 – plugged in on the side of the shield – Serial 0 (which is also used for USB, and the BT module must be removed when programming)

If you wanted to use the USB cable and run the balance code and App tethered, you would use the COM port of your Arduino (33 in this case) and Serial 0.

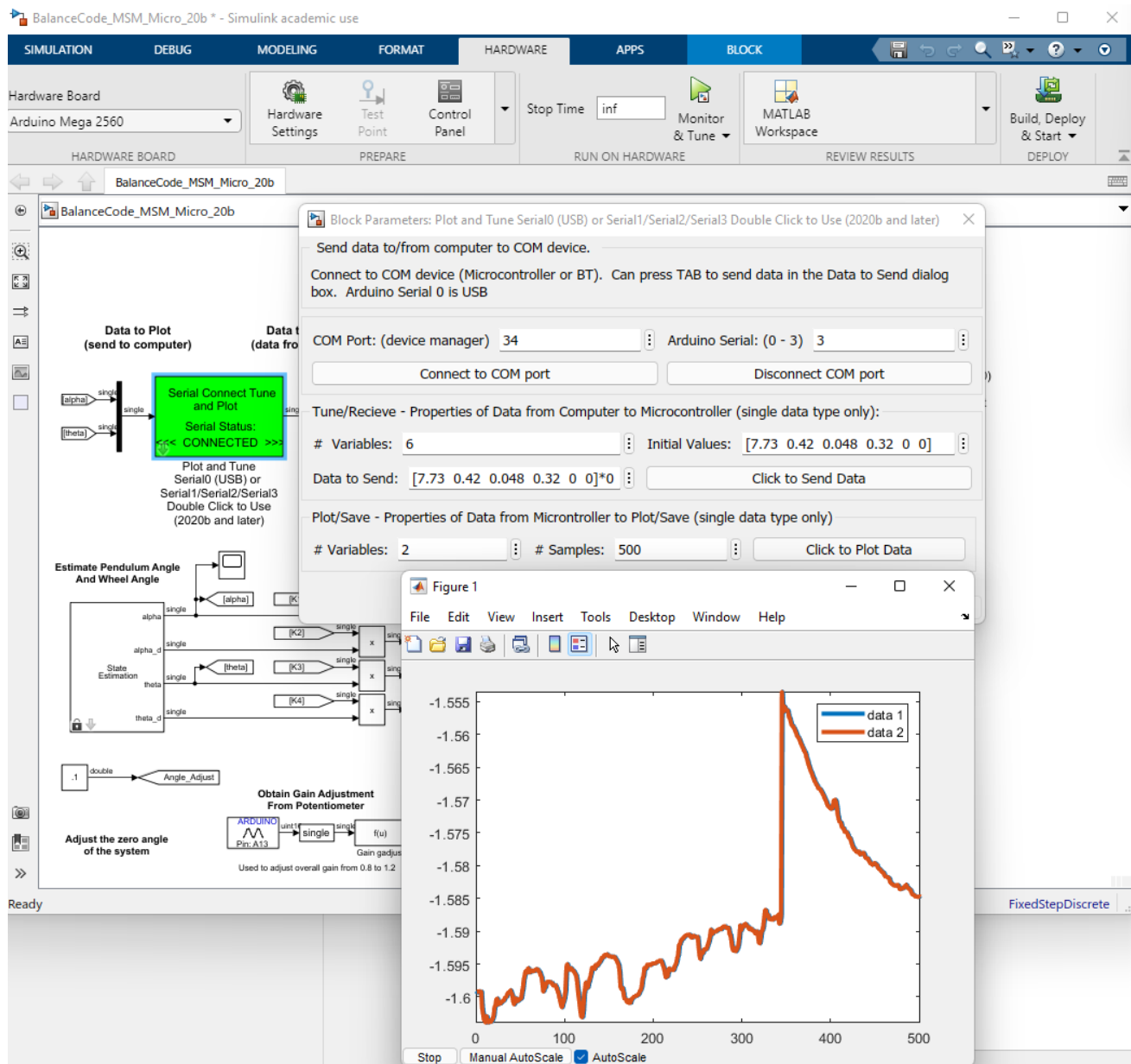
3. Click Apply to save the changes.

The screenshot displays the Simulink academic use interface. The top toolbar includes tabs for SIMULATION, DEBUG, MODELING, FORMAT, HARDWARE, APPS, and BLOCK. The hardware board is set to 'Arduino Mega 2560'. A dialog box titled 'Block Parameters: Plot and Tune Serial0 (USB) or Serial1/Serial2/Serial3 Double Click to Use (2020b and later)' is open, showing the following settings:

- Send data to/from computer to COM device.
- Connect to COM device (Microcontroller or BT). Can press TAB to send data in the Data to Send dialog box. Arduino Serial 0 is USB
- COM Port: (device manager) 34
- Arduino Serial: (0 - 3) 3
- Buttons: Connect to COM port, Disconnect COM port
- Tune/Receive - Properties of Data from Computer to Microcontroller (single data type only):
 - # Variables: 6
 - Initial Values: [7.73 0.42 0.048 0.32 0 0]
 - Data to Send: [7.73 0.42 0.048 0.32 0 0]*0
 - Click to Send Data
- Plot/Save - Properties of Data from Microcontroller to Plot/Save (single data type only):
 - # Variables: 2
 - # Samples: 500
 - Click to Plot Data
- Buttons: OK, Cancel, Help, Apply

The background Simulink model shows a pendulum system with state estimation and feedback control. It includes blocks for 'Data to Plot (send to computer)', 'Data to Receive (data from computer)', 'Estimate Pendulum Angle And Wheel Angle', 'Full State Feedback Control', and 'Output to DC Motors'. The status bar at the bottom indicates 'Ready', '79%', and 'FixedStepDiscrete'.

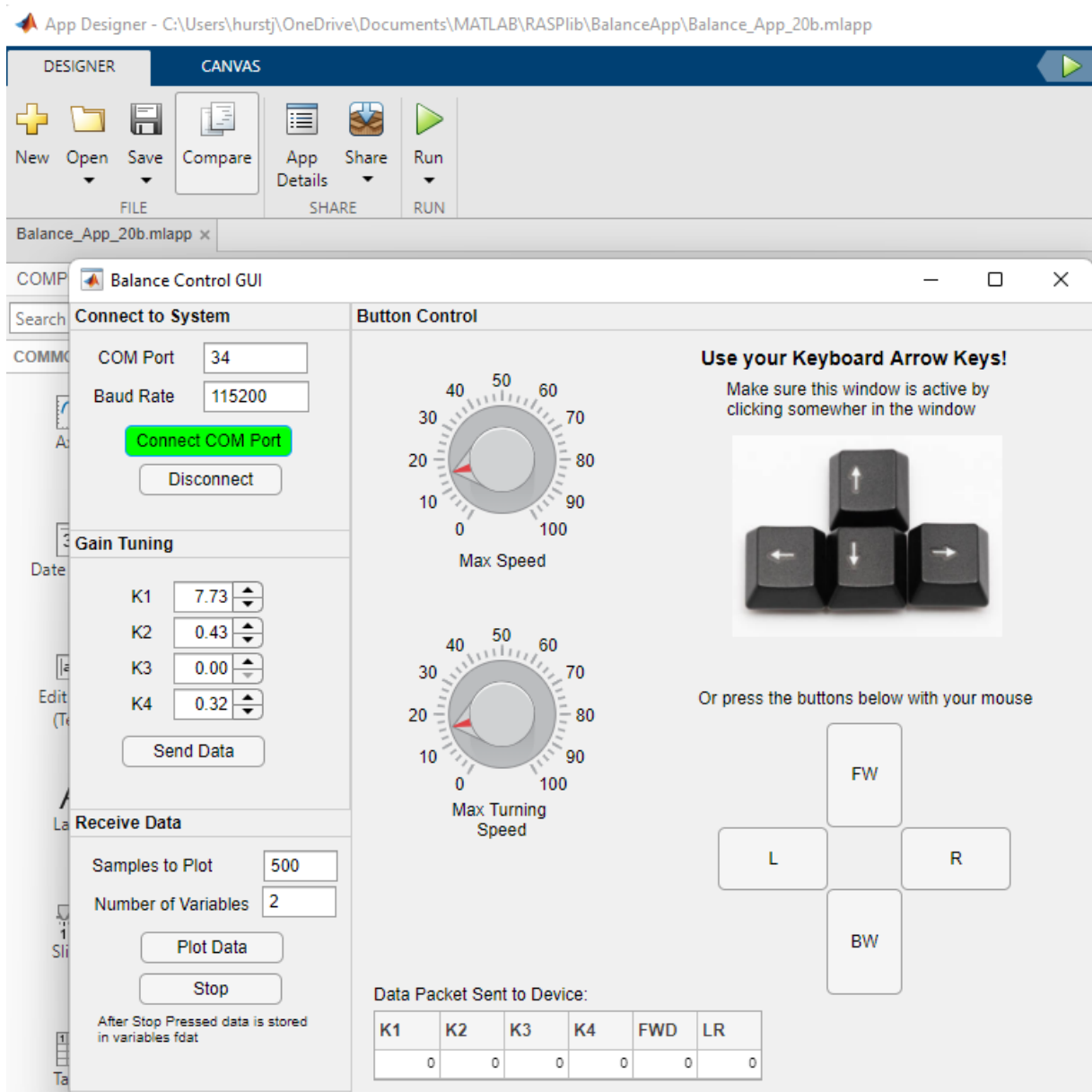
4. Click Build Deploy & Start to program the Arduino with the balance code.
5. After the code is downloaded you can click "Connect to COM port" and flashing red light on the Bluetooth module should say solid red. You are connected. Click "Click to Plot Data" to verify it is communicating with the Bluetooth.



Click Stop to close the graph, you can then of the data logged by typing `plot(fdat)` in the command line. It also stores the data in a file `fdat.mat`. Click Disconnect COM port to close the connection and close the file.

Using the Balance App

1. Open the balance App in the RASPLib/BalancApp folder
2. Click Run to open the app. Verify the COM port is the Bluetooth port, click Connect and flashing red light on the Bluetooth module should say solid red.
3. You can now Send, Plot and interact with the device. With batteries installed, and the system balancing y use the app to drive and steer the MinSeg.



If you are experiencing latency make sure your laptop is plugged in or the power settings are optimized for performance

Please see the product page for your device for the latest videos on getting started and using the code.