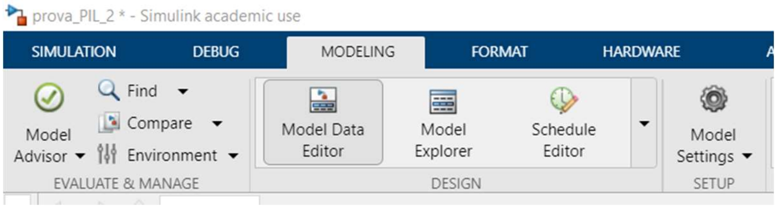
How to create a PIL block in Simulink (Matlab 2019b)

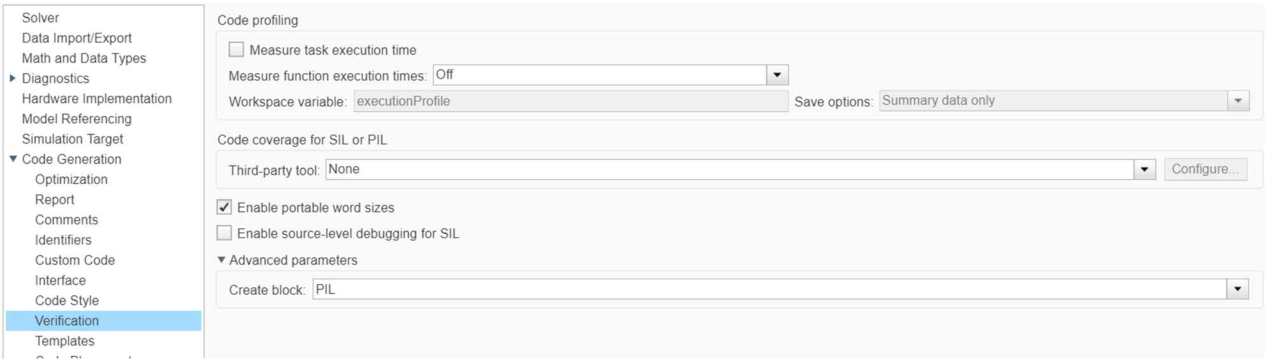
Note: instruction "as is" for Matlab 2019b; in previous versions the steps are the same, maybe only the names change a little

1. Create a new empty working directory and add to Matlab path
2. Use **"Get Add-ons"** to install the Hardware support package for target board; after the download, follow the guided procedure to install drivers etc.
3. In Simulink model page, collect whatever you want to run on target and create a **sub-block** containing it
4. Open **"Configuration Parameters"** (aka **"Model Settings"**):



   a. In **"Hardware implementation"** tab, select your target board

   b. Expand "Code generation" tab, go to **"Verification"** tab. Thick **"Enable portable word size"**, then click on **…** and **select PIL in "Create block"**



Code profiling

□ Measure task execution time

Measure function execution times: Off

Workspace variable: executionProfile

Save options: Summary data only

Code coverage for SIL or PIL

Third-party tool: None

☑ Enable portable word sizes

□ Enable source-level debugging for SIL

Advanced parameters

Create block: PIL

   c. Save and close "Configuration parameters"

5. Right-click on the target sub-block created at point 2. Select **"C/C++ Code"** → **"Deploy this subsystem to hardware"** (probably in older version is "Run on target hardware" or similar). It will create a lot of files in your working directory; at the end of compilation it will open a new Simulink page containing the sub-block compiled (marked with PIL). **Drag and drop** this new block to your Simulink model, and use it to replace the sub-block of point 2.

   Note: you have to save the Simulink page that appear. Regarding the name, "it depends": most likely you will choose whatever name, and when you'll start the simulation it will not work (error: e.g. "Controller.slx not found"). You have to delete the Matlab folder from Raspberry home if using raspberry, recompile the target block and save the Simulink with the name indicated (in this example, Controller.slx)

6. To start PIL, simply **start the simulation** with your board attached to USB (or Ethernet if Raspberry). You can see it is working since e.g. the TX/RX LED will blink continuously. You can even set simulation time to inf to see what happens as the time goes by (e.g. you can adjust a slide while simulation run etc…)

Troubleshoot:

With Arduino, at point 6. error “Data type double is not supported for SIL or PIL execution” appeared. Solve it adding the block “Cast to single” to all data lines so that each data type is single and PIL can start

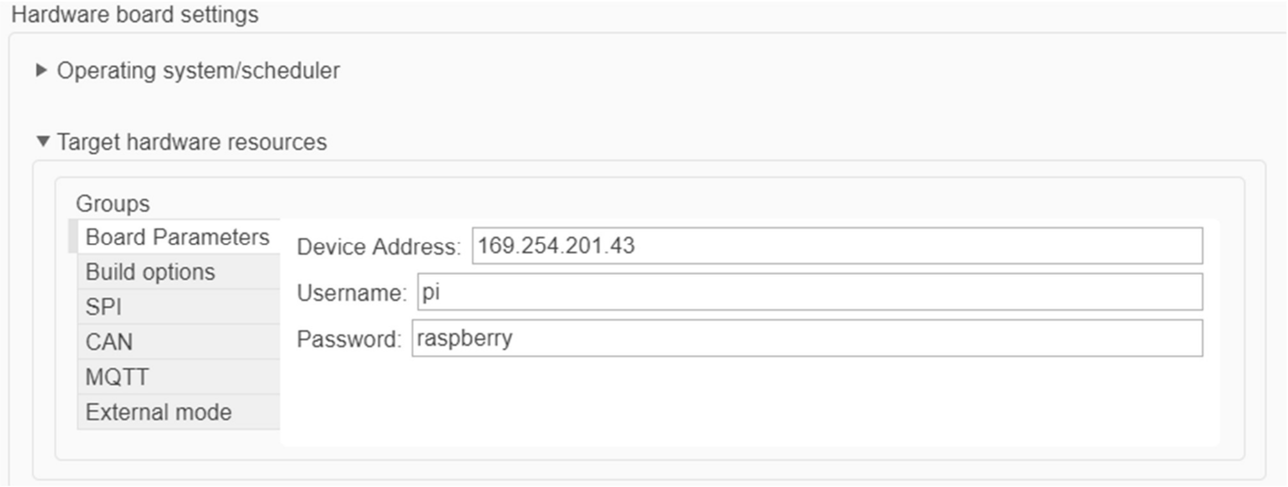
Special case with Raspberry Pi hardware (that host a Linux operating system)

Note: the code that will run on target will be automatically stored at /home/pi/MATLAB_rt/…

Make sure there is no MATLAB_rt folder in the Raspberry before you start

Note: the code that will run on target is directly compiled by GCC on the target

1. Create a new **empty directory** and add to Matlab path
2. Connect Raspberry to PC using Ethernet cable; get the IP of the Raspberry by booting into Raspbian and run ifconfig (for our Raspberry Pi2 → 168.254.201.43
3. Open “Configuration Parameters”; in “Hardware implementation” select Raspberry Pi, then expand “Target hardware resources” and set the IP address of Raspberry, username and password (for us: pi/raspberry)



Device Address: 169.254.201.43

Username: pi

Password: raspberry

4. Do the same as point 4b and 4c of previous page
5. Create the sub-block containing what you want to run on hardware, follow steps as for point 5 of previous page. Make sure you **save the block as “Subsystem.slx” in the same working directory** of point 1
6. To start PIL, simply **start the simulation** with your board attached to USB. You can see it is working since e.g. the TX/RX LED will blink continuously. You can even set simulation time to inf to see what happens as the time goes by (e.g. you can adjust a slide while simulation run etc…)

Troubleshoot:

Make sure the working directory is empty before you start, and that there is no Matlab folder in Raspberry (otherwise it is likely to happen a build error)