Beaglebone setting

- Connect the Beaglebone Black board (BBB) to the PC

- A window automatically opens. It contains the filesystem of the BBB. In case the windows does not open, push the Windows button, then access the This PC folder, finally open the folder BEAGLEBONE(F:). See the Figure 1

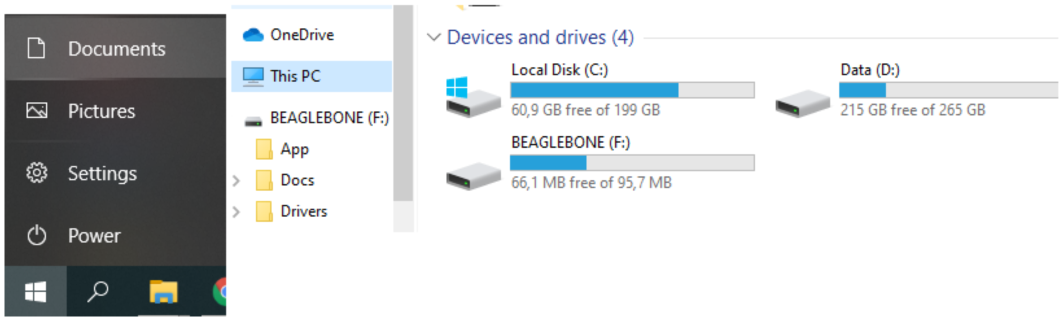


Figure 1 – BEAGLEBONE folder

- Open the README file and follow the instructions to install the driver (see Figure 2 and Figure 3). Read carefully the note especially if you have 64 bit Windows OS. On the left you have a counter that tells you if the installation of the driver is completed

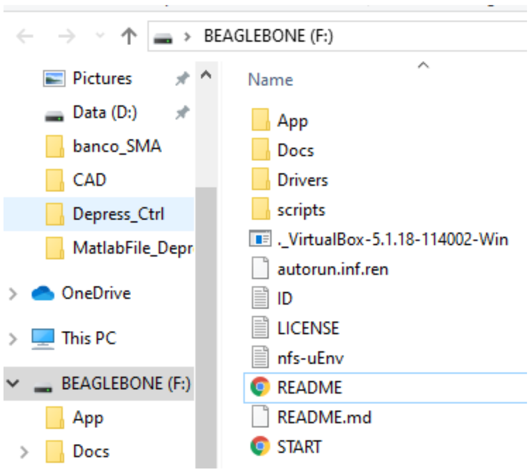
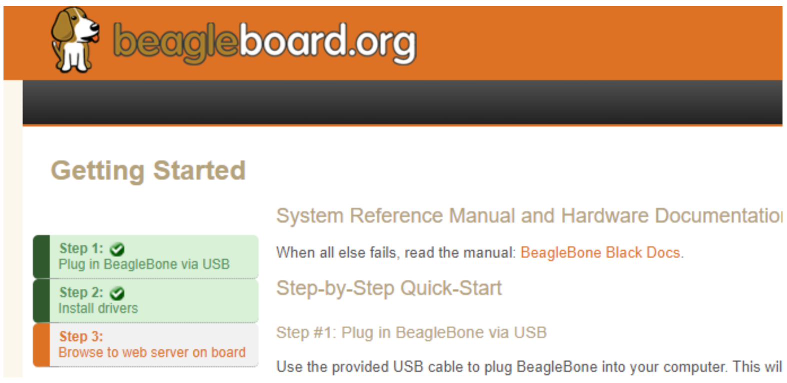


Figure 2 – README file



System Reference Manual and Hardware Documentatio

When all else fails, read the manual: BeagleBone Black Docs.

Step-by-Step Quick-Start

Step #1: Plug in BeagleBone via USB

Use the provided USB cable to plug BeagleBone into your computer. This wil

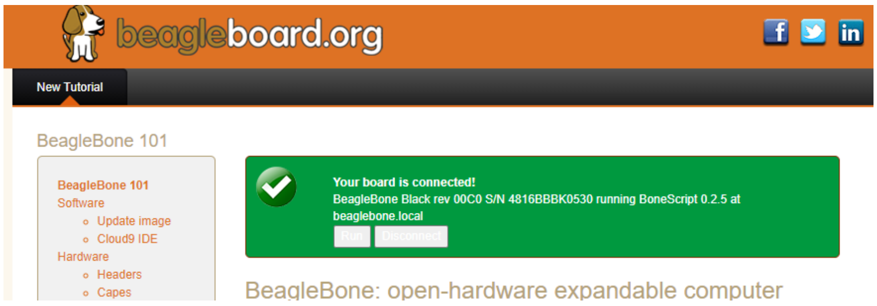
Figure 3 – Getting started with tick on the left to show the steps correctly ended

- In case of problems during installation look at this link:

https://www.element14.com/community/community/designcenter/single-board-computers/next-genbeaglebone/blog/2016/11/14/beagle-bone-black-window-10-driver-fix

- Restart computer whilst holding down the shift key
- Click on "Troubleshoot“
- Select "Advanced Options“
- Select "Startup Settings“
- Press F7
- Finally click restart

- When step 1 and step 2 are terminated push the link http://192.168.7.2 to check if the BBB is correctly connected to the PC (see Figure 4).



Your board is connected!
BeagleBone Black rev 00C0 S/N 4816BBBK0530 running BoneScript 0.2.5 at beaglebone.local

BeagleBone: open-hardware expandable computer

Figure 4 – Move on the board to see if it is linked

Matlab 2020b setting

- Run Matlab as administrator (the BBB is still connected)

- Push the Add-Ons button in the Home items of the main menu on the top (see Figure 5)

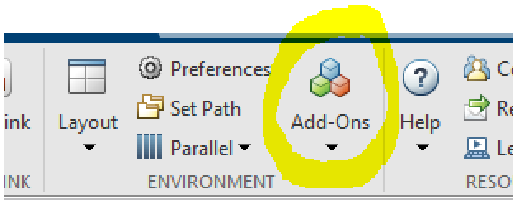


Figure 5 - Add-Ons button

- Write "Beaglebone Black" In the search bar on top (see Figure 6)

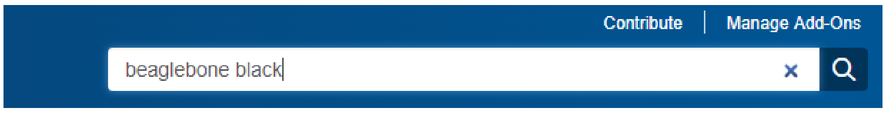
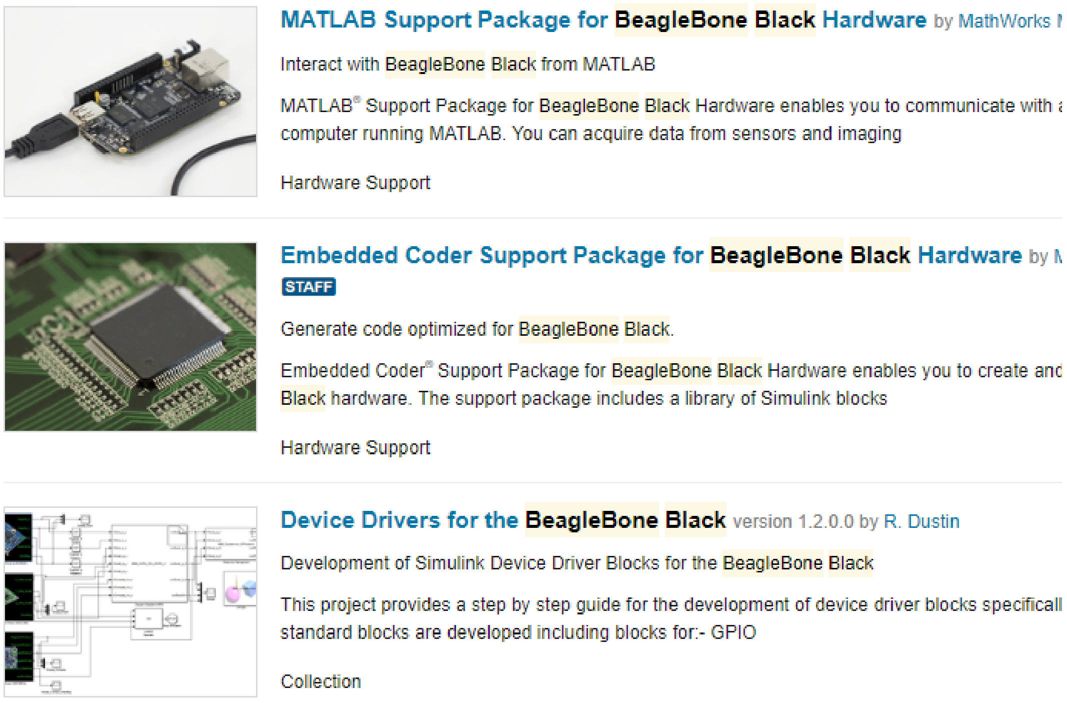


Figure 6 - Search bar

- Install "Embedded Coder Support Package for BeagleBone Black Hardware" (It requires the Embedded Coder toolbox, in case you did not install it during the installation of Matlab, you can search it in the Add-On repository, by pushing on the button Add-Ons, and install it). Notice that "Matlab Support Package for Beaglebone Black" will be automatically installed. See Figure 7.



MATLAB Support Package for BeagleBone Black Hardware by MathWorks

Interact with BeagleBone Black from MATLAB

MATLAB® Support Package for BeagleBone Black Hardware enables you to communicate with
computer running MATLAB. You can acquire data from sensors and imaging

Hardware Support

Embedded Coder Support Package for BeagleBone Black Hardware by
STAFF

Generate code optimized for BeagleBone Black.

Embedded Coder® Support Package for BeagleBone Black Hardware enables you to create and
Black hardware. The support package includes a library of Simulink blocks

Hardware Support

Device Drivers for the BeagleBone Black version 1.2.0.0 by R. Dustin

Development of Simulink Device Driver Blocks for the BeagleBone Black

This project provides a step by step guide for the development of device driver blocks specificall
standard blocks are developed including blocks for:- GPIO

Collection

Figure 7 – "Embedded Coder Support Package for BeagleBone Black Hardware" Add-On installation

- After the installation, you have to manage the configuration of the settings. Open the "Manage Add-Ons" in the Home menu (see Figure 8) . Push on the gear button next to the previously installed Add-Ons (see Figure 9).

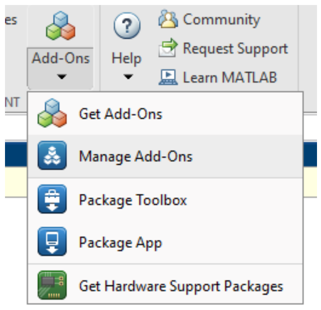
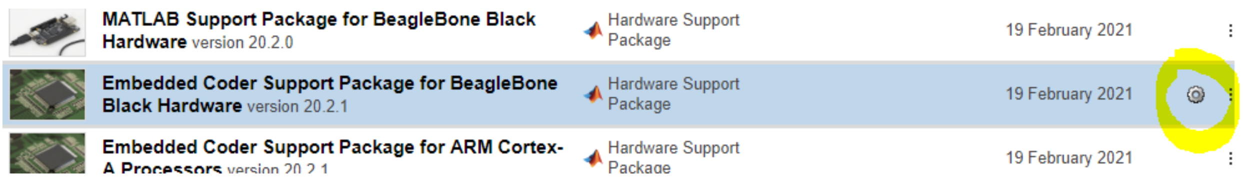


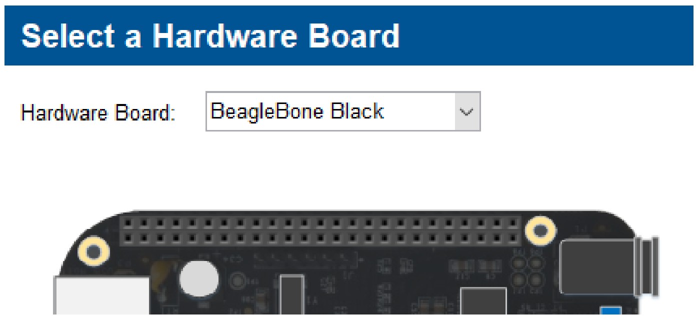
Figure 8 – Manage Add.Ons menu



MATLAB Support Package for BeagleBone Black Hardware version 20.2.0	Hardware Support Package	19 February 2021
Embedded Coder Support Package for BeagleBone Black Hardware version 20.2.1	Hardware Support Package	19 February 2021
Embedded Coder Support Package for ARM Cortex-A Processors version 20.2.1	Hardware Support Package	19 February 2021

Figure 9 – Setup button

- In the windows that opens, in Hardware Board, select the "Beaglebone Black" (it should be already selected). Then push "NEXT".

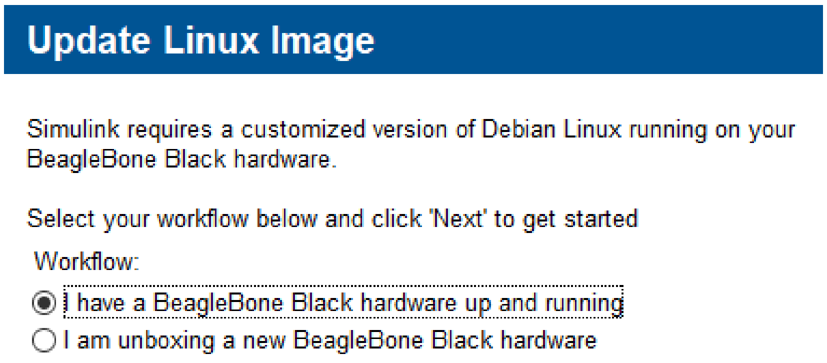


Select a Hardware Board

Hardware Board: BeagleBone Black

Figure 10 - Select Board

- Select the item "I have a Beaglebone Black hardware up and running" and push "NEXT".



Update Linux Image

Simulink requires a customized version of Debian Linux running on your BeagleBone Black hardware.

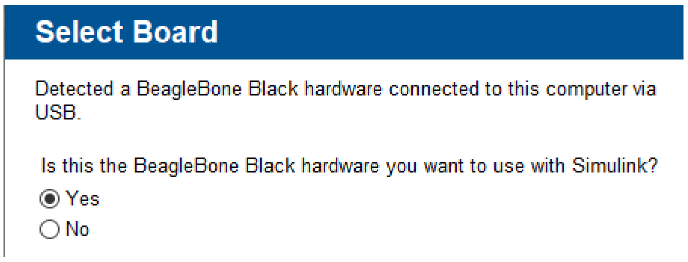
Select your workflow below and click 'Next' to get started

Workflow:

● I have a BeagleBone Black hardware up and running
○ I am unboxing a new BeagleBone Black hardware

Figure 11 – Update linux Image Windows

- Select “YES” and push “NEXT” (see Figure 12).


Select Board

Detected a BeagleBone Black hardware connected to this computer via USB.

Is this the BeagleBone Black hardware you want to use with Simulink?

[✓ Yes] No

Figure 12 - Yes window

- In the next window, leave all as it is and click on “NEXT”

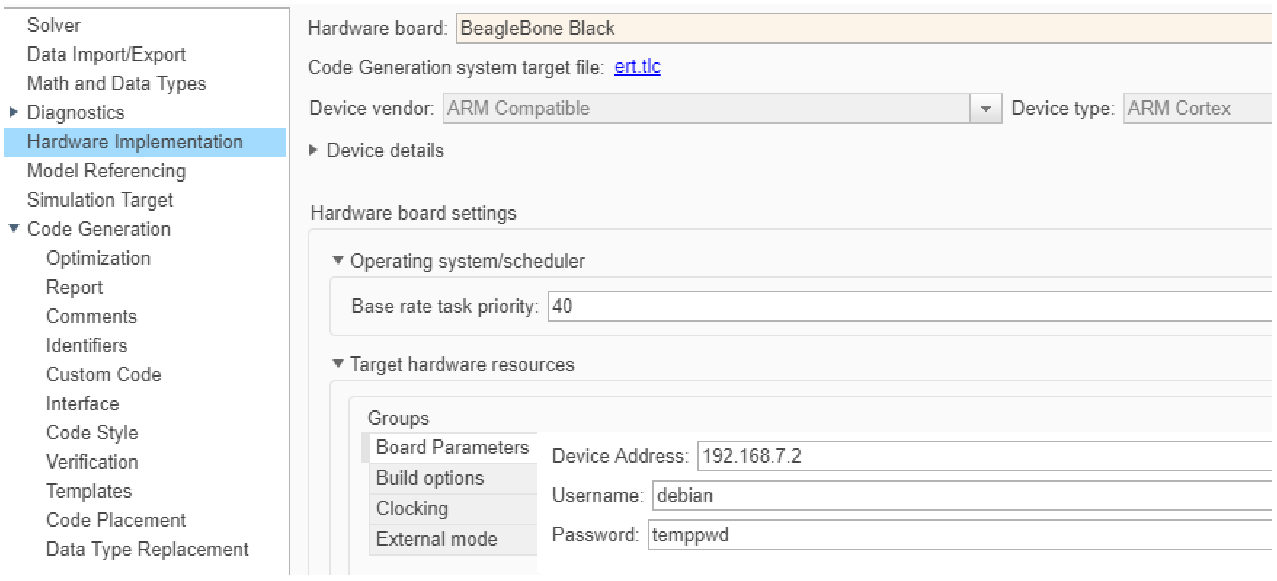
- Press “START” to start the BBB configuration. The operation may require some time (15 minutes or more – be patient). In case the procedure fail, please proceed uninstalling the “Embedded Coder Support Package for BeagleBone Black Hardware” Add-on and try it again. To uninstall the Add-on follow the steps in the Appendix A

- To check that everything works correctly try to reproduce the Matlab tutorial shown at this link:

https://it.mathworks.com/help/supportpkg/beaglebone/examples/getting-started-with-beaglebone-black-support-package.html

Please notice carefully:

1) In the “Modeling” menu push “Model Settings” and in the “Hardware Implementation” item configure as shown in the Figure 13.



Hardware board: BeagleBone Black

Code Generation system target file: ert.tlc

Device vendor: ARM Compatible

Device type: ARM Cortex

Base rate task priority: 40

Device Address: 192.168.7.2

Username: debian

Password: temppwd

Figure 13 - Hardware implementation configuration

2) In the main menu the “Hardware” item will appear. Here it is possible to manage the creation and download of the code to the BBB

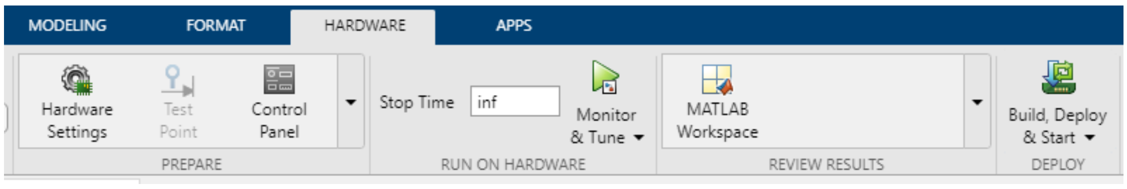


Figure 14 – “Hardware” menu

- On the right it is possible to choose among Build (compile), Build and Deploy (compile and download to the board) or Build, Deploy and Start (compile, download and run the program)

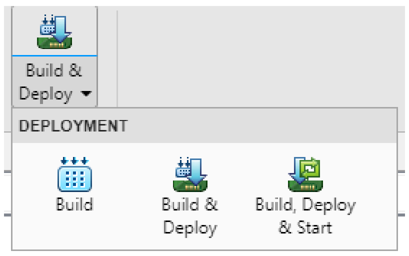


Figure 15 - Compiling and download options

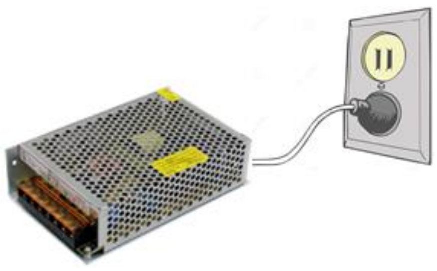
- Use these options in this way:
  - Choose the Build option
  - Push Monitor & Tune to download and run the program on the board and to monitor the behavior of the board on the Simulink model.

Take confidence with the system and find the slider limits

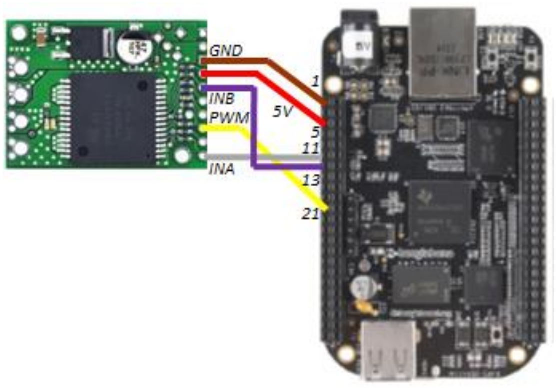
Slider limits definition procedure

- Check that the 24V switch on the rear side of the system is set to OFF.

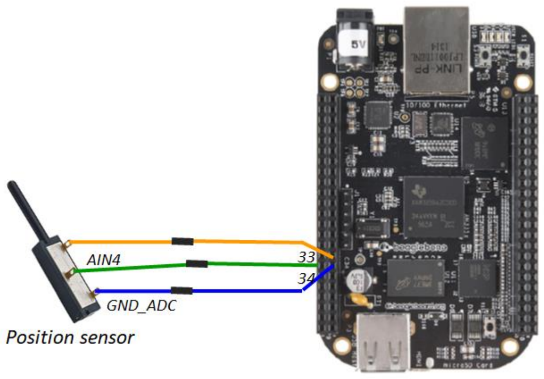
- Connect the 24V power supplier to the 220V network plug.



- Connect the BBB to the Driver.



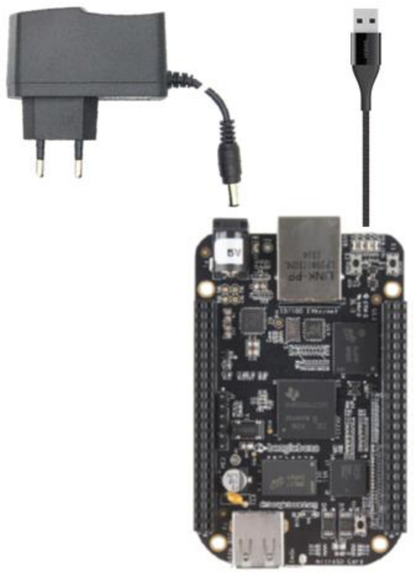
- Connect the BBB to the sensor.



NOTICE: use the wires provided in the kit, do not use the wires of the sensor (they are too short and it is possible the sensor breaks)

- Connect the BBB to PC with the USB cable.

- Connect the 5V power supply to the BBB.

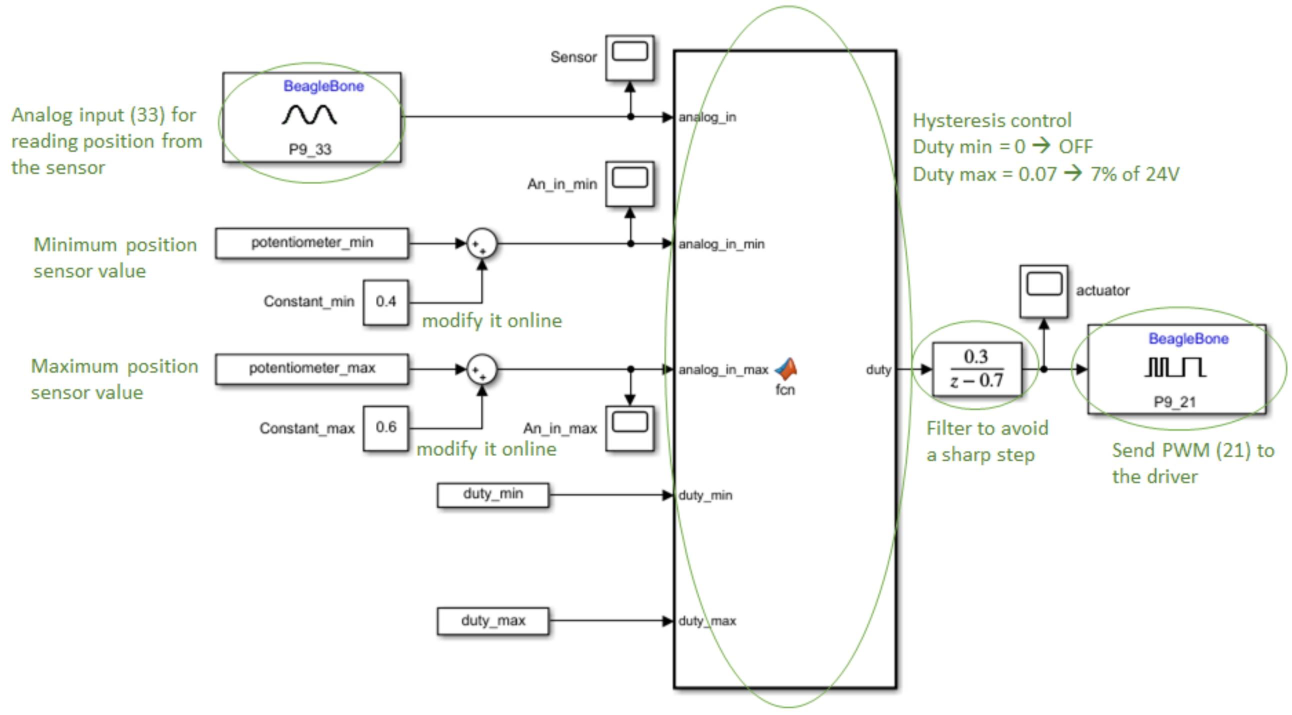


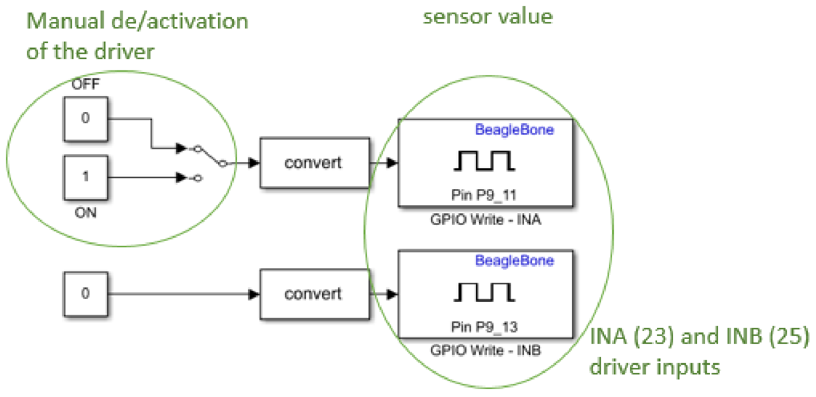
NOTICE: The 5V power supply is necessary to allow the BBB to have 5V on the output pin number 1 and to supply the driver.

- Run Matlab as Administrator

- Open the Simulink model called “FindLimits20XXx.slx”. The model implements a simple two state control such that

- If the position sensor value is lower than the minimum threshold (called potentiometer_min), the driver is turned on at the maximum value (duty_max maximum PWM percent) and it warms up the SMA wire;
- If the position sensor value is higher than a maximum threshold (called potentiometer_max), the driver is turned off (duty_max = 0).

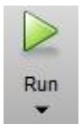




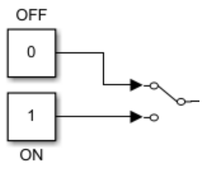
- Open the "Parameters.m" file. The file contains the parameters to be use in the Simulink scheme that are
  - potentiometer_min: minimum position sensor value,
  - potentiometer_max: maximum position sensor value,
  - duty_min: minimum PWM percent value,
  - duty_max: maximum PWM percent value.

- Set the previous variable values to:
  - potentiometer_min = 0,
  - potentiometer_max = 2,
  - duty_min = 0,
  - duty_max = 0.1.

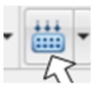
- Run the "Parameters.m" script



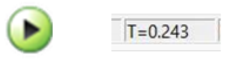
- In the Simulink model, check that the manual switch block is set to OFF



- Build it and press Start simulation



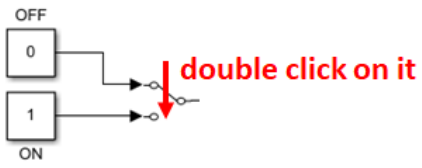
- Run the simulation and wait until the simulation starts



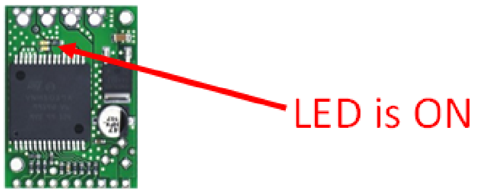
- Switch the 24V driver supply button to ON

- The driver LED should be OFF

- Set to ON the manual switch

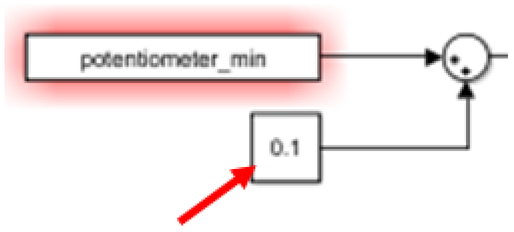


- The driver LED should be ON



- If not, then the *potentiometer_min* value is smaller than the current position value measured by the sensor.

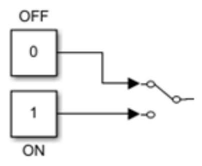
-Use the "Constant_min" block to increase the *Analog_in_min* input value to obtain that current sensor value.



- Now the driver LED is ON and the slider start moving

- After some time the slider stops moving because the SMA wire reaches its shortest length.
Do the follow very quickly:

1) Read the maximum value measured by the sensor (*max_position*)
2) Set the Manual switch to OFF and check if the Driver LED is OFF

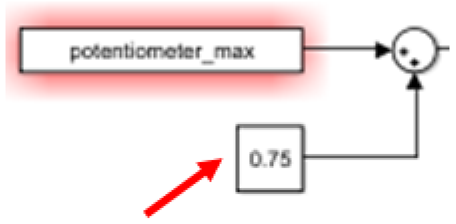


➕

If the Driver
LED is still ON

ATTENTION: The SMA wire get damaged if the current flows for too long after that the wire has reached its minimum length.

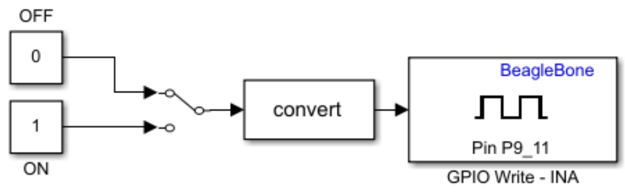
- Decrease the “Constant_max” block value to have *Analog_in_max = max_position*



- Now the slider should reach the top limit, when the driver is ON, then the driver should automatically turn OFF, until the slider reaches the bottom limit. The movement is repeated periodically.

- It is possible to **turn off the system** and save the *Analog_in_min* value and the *Analog_in_max* value to use them in the project.

- Disable the driver by switching OFF the INA signal



- Set the power supply switch to OFF.

- Stop the simulation pushing the stop button.

■

ATTENTION: Do not push the stop button before having turned off the 24V switch. The driver will become active unexpectedly.

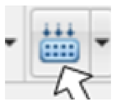
- Check that the LED is OFF

- Do not touch the SMA wire, it is hot!

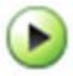
Right procedure to download the program on the BBB

- Set the power supply switch to OFF.

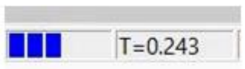
- Download your code to BBB using the Embedded Coder button in Simulink.



- Run the simulation pushing the play button.

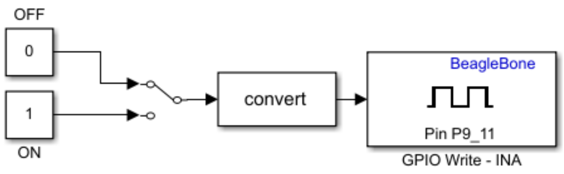


- Check that the simulation is started (the timer on the bottom right is counting)

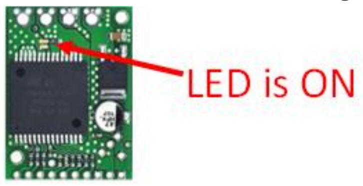


- Set the power supply switch to ON.

- Enable the driver by switching ON the INA signal

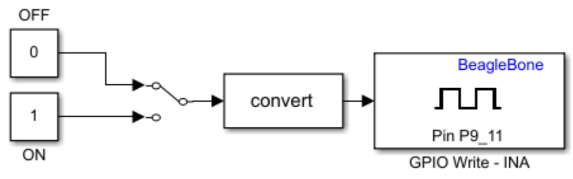


- Look at the driver, if a red light is on the current is flowing in the wire.



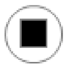
Right procedure to switch off the system

- Disable the driver by switching OFF the INA signal



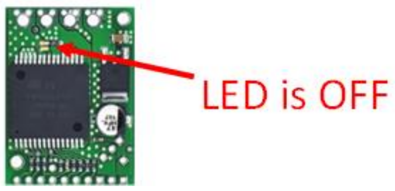
- Set the power supply switch to OFF.

- Stop the simulation pushing the stop button.



ATTENTION: Do not push the stop button before having turned off the 24V switch. The driver will become active unexpectedly.

- Check that the LED is OFF



- Do not touch the SMA wire, it is hot!

IMPORTANT NOTES

- The Duty in the PWM Simulink block goes from 0 to 1 and it represents the **percent of the maximum voltage** (24V). Current = Voltage / Resistance

- Use a **limited PWM** (in the example is 0.1). PWM determines the current that flows on the wire, the higher the current the faster is the wire response. Start with a low limit and then when you are confident that your algorithm works correctly, try with a higher one (below 0.5)

- Always **check the LED** of the driver to see if it is unintentionally working. In this case Switch OFF the system with the power supply button in the rear side of the system

- Put a **safety limit on the slider position** (if it crosses the limit, set PWM=0)

- **Do not keep the current flowing continuously into the wire for too long** (if the slider does not reach the desired position after some time simply it can not). A too high temperature can wear out and damage the wire.

- When you terminate a test**,** set the power supply OFF **before** stopping the Matlab simulation**.**

- **Do not stay to close to the SMA wire with your face**.

Appendice A

Sometimes support packages become corrupt and cannot complete installation or encounter errors that you cannot get past. In these scenarios, it is beneficial to perform a clean re-installations of your Support Package rather than a regular re-installation. To do this, follow these steps:

<u>For R2016b and newer</u>

1. Open MATLAB and choose the drop-down menu for "Add-Ons", followed by the "Manage Add-Ons" option. Find your Support Package on the generated list and click the "Uninstall" button across from it. Once this uninstall is complete, close out of MATLAB.

2. Navigate to the following folder on your computer and delete the contents you find there:
   Windows = *C:\ProgramData\MATLAB\SupportPackages\R20XXx\*
   Mac = */Users/<username>/Documents/MATLAB/SupportPackages/R20XXx/*
   Linux = */home/<username>/Documents/MATLAB/SupportPackages/R20XXx/*

   ProgramData is a hidden folder, but you can use the shortcut *%ProgramData%* from Windows Search to jump right to the current user's *ProgramData* folder.

3. Open the 'Downloads' folder on your computer, then navigate to
   \MathWorks\SupportPackages\R20XXx\
   Delete the contents of the "R20XXx" folder.

4. Run the Support Package installer again and re-install your Support Package now.

<u>For R2016a and prior</u>

1. Open MATLAB and choose the drop-down menu for "Add-Ons", followed by the "Get Hardware Support Packages" option. In the Support Package Installer choose the "Uninstall" radio button, then click "Next". Follow the remaining menus to uninstall the desired Support Package.

2. Navigate to the following folder on your computer and delete the contents you find there:
   Windows = *C:\MATLAB\SupportPackage\R20XXx\*
   Mac = */Users/<username>/Documents/MATLAB/SupportPackages/R20XXx/*
   Linux = */home/<username>/Documents/MATLAB/SupportPackages/R20XXx/*

3. Open the 'Downloads' folder on your computer, then navigate to
   \MathWorks\SupportPackages\R20XXx\
   Delete the contents of the "R20XXx" folder.

4. Open MATLAB again and run the following commands:
   ```
   >>restoredefaultpath
   >>savepath
   ```

5. Run the Support Package installer again and re-install your Support Package now.

Note: This will remove all support packages that you currently have installed.