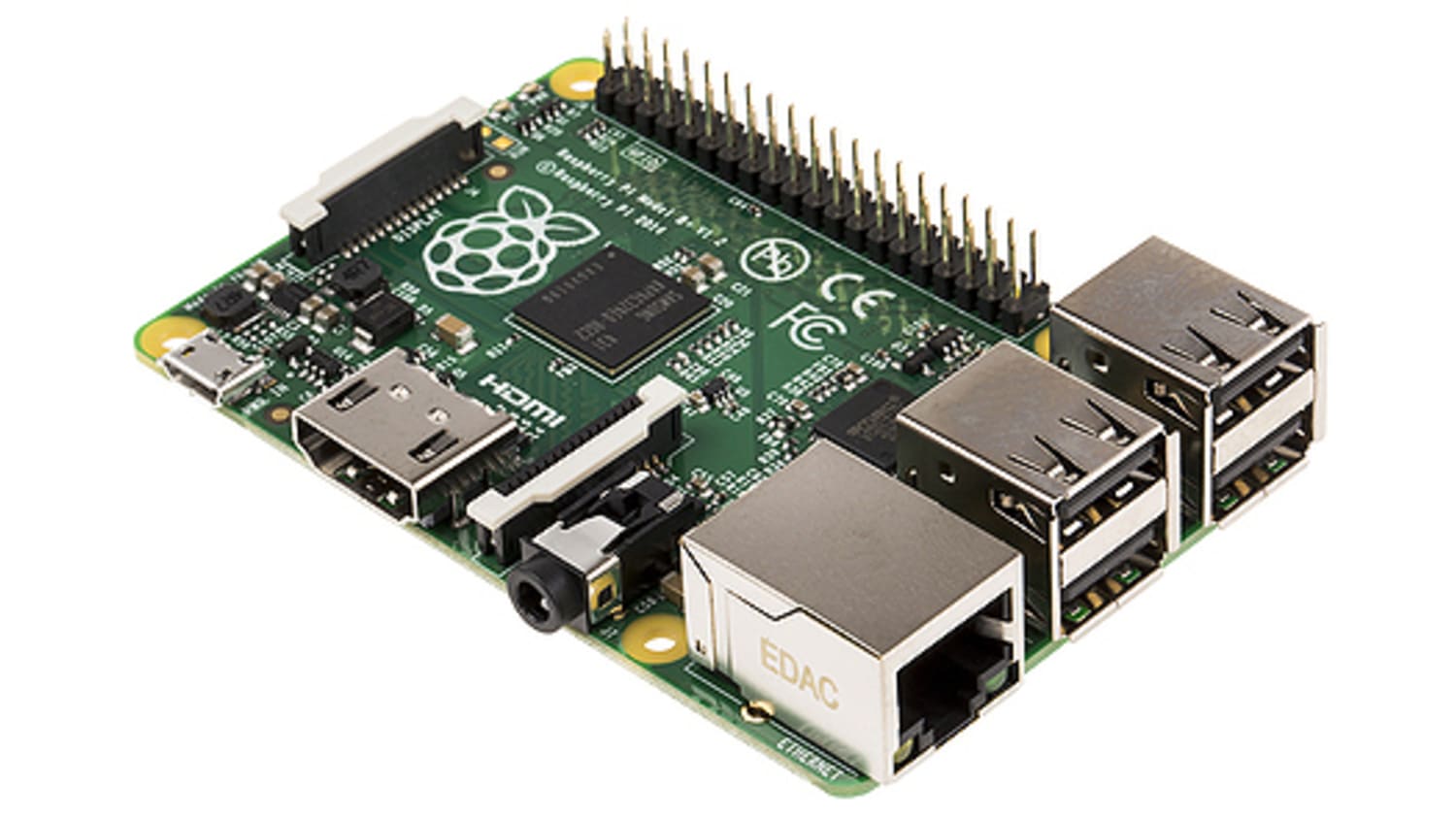
Elevator Control System with Simulink on Raspberry Pi





Author: Kristian Snel

Tutor: Jeedella, Jeedella

First release: 07-12-2021

Contents

[1. Project description 2](#_Toc63591573)

[1.1. Scope 2](#_Toc63591574)

[1.2. Functional requirements: 3](#_Toc63591575)

[1.3. Specifications 3](#_Toc63591576)

[2. System overview 4](#_Toc63591577)

[2.1. Equipment 4](#_Toc63591578)

[Laptop with Matlab (2020a) 4](#_Toc63591579)

[USB type A to micro usb cable 4](#_Toc63591580)

[24VDC power supply and optionally a Raspberry Pi 5VDC power supply 4](#_Toc63591581)

[Multimeter 4](#_Toc63591582)

[Ethernet cable 4](#_Toc63591583)

[2.2. Hardware 4](#_Toc63591584)

[Sabertooth2x12 V 1.00 motor control board 4](#_Toc63591585)

[Raspberry Pi 4 B 5](#_Toc63591586)

[Raspberry Pi Arduino motor encoder readout protoshield 5](#_Toc63591587)

[Maxon Motor (exact specifications unsure) 5](#_Toc63591588)

[Wiring diagram 6](#_Toc63591589)

[2.3. Setting up the system 6](#_Toc63591590)

[2.4. Suggested improvements 6](#_Toc63591591)

[3. Appendix 7](#_Toc63591592)

[3.1. How to use Simulink Support Package for Raspberry Pi Hardware 7](#_Toc63591593)

[Setup Raspberry Pi hardware 7](#_Toc63591594)

[Select hardware (after reopening project) 11](#_Toc63591595)

[3.2. Hardware setup in Simulink: 12](#_Toc63591596)

[3.3. F.A.Q. 13](#_Toc63591597)

# Project description

## Scope

The goal of this project is to setup a motor controller with a Maxon brushless DC motor using a Raspberry Pi 4 B and Simulink. This system will then be used by students to design an elevator control system.

## Functional requirements:

The system functional requirements are as following:

|  |  |  |
| --- | --- | --- |
| **Req. Tag** | **Function** | **MoSCow** |
| #1 | Elevator vertical movement with a brushless DC motor | Must |
| #2 | System modeling based on Simulink | Must |

## Specifications

The system specifications are determined from the functional requirements.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Spec. Tag** | **Dimension** | **Specification** | **Unit** | **Value** |
| #1 | Rotational speed | Brushless DC motor speed | rpm | 60 |

# System overview

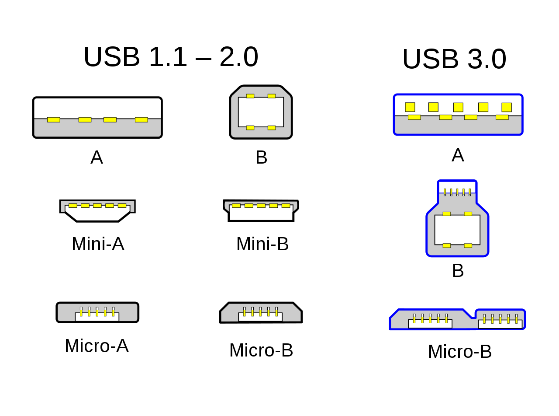
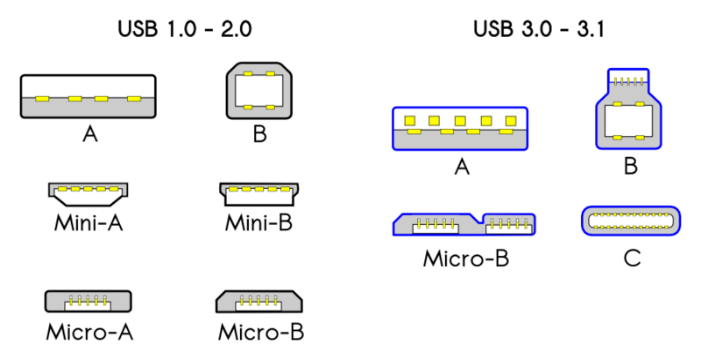
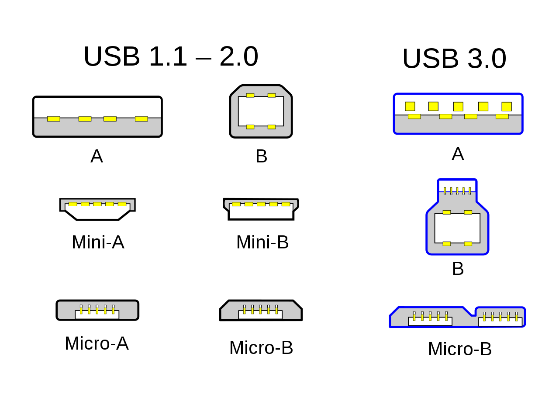
## Equipment

### Laptop with Matlab (2020a)

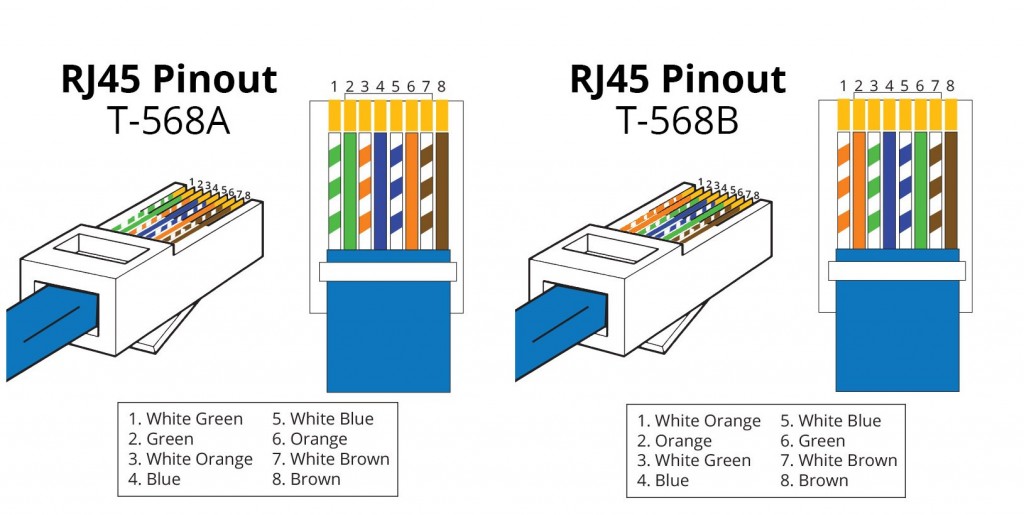
A laptop with Matlab version 2020a or later is required (earlier versions of Matlab are not sufficient as the Raspberry Pi 4 B is not supported pre 2020a).

### USB type A to micro usb cable

Any USB type A (1.1 – 3.0) to USB Micro-B cable is required to power the Raspberry Pi from a laptop or USB type A phone charger. (USB type C to USB type Micro-B also possible with USB type C phone charger).

### 24VDC power supply and optionally a Raspberry Pi 5VDC power supply

A 24VDC power supply unit is required to power the motor controller. A 5V power supply for the Raspberry Pi is optional but this requires the 5V connection from the Motor controller to the Raspberry Pi to be disconnected.

### Multimeter

A multimeter for hardware debugging purposes is suggested.

### Ethernet cable

Any RJ45 male to male straight through ethernet cable cat 5 (or later).  
<https://community.fs.com/blog/introduction-of-the-rj45-interface.html#:~:text=RJ45%20is%20a%20standard%20type,are%20also%20called%20RJ45%20cables>.

## Hardware

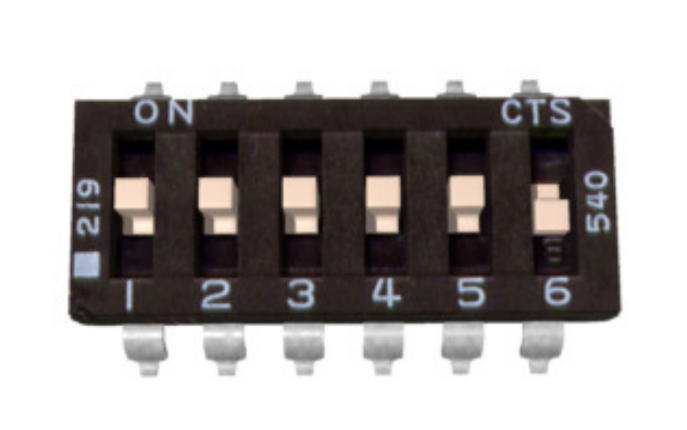
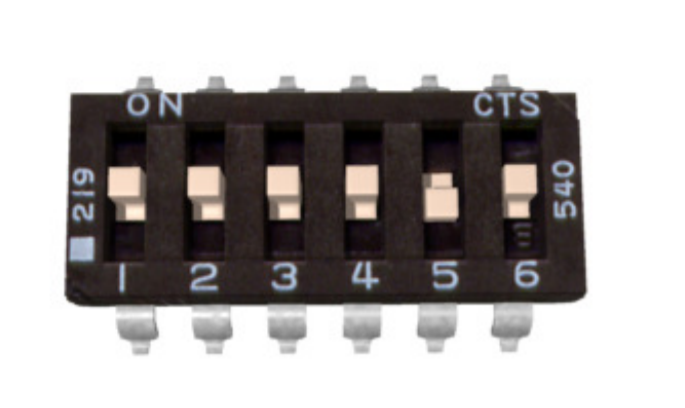
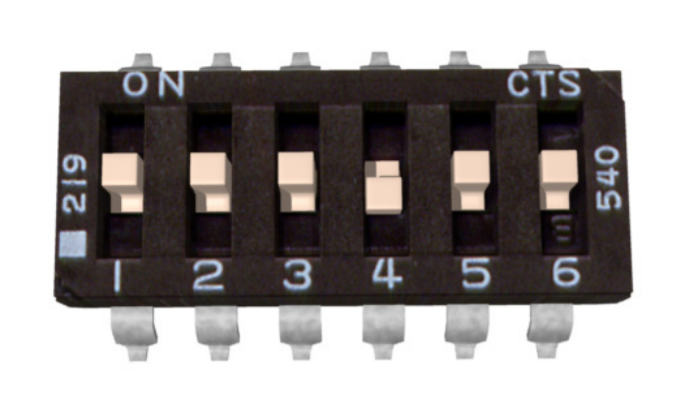
The hardware contains the following components:

### Sabertooth2x12 V 1.00 motor control board

The sabertooth 2x12 V 1.00 is a brushless dc motor controller able to provide two motors with power in two directions. The board is supplied with a battery power source (but a bench power supply can be provided in parallel with the battery) between and . The board features 6 dip switches which have to be set in a certain order to function properly:

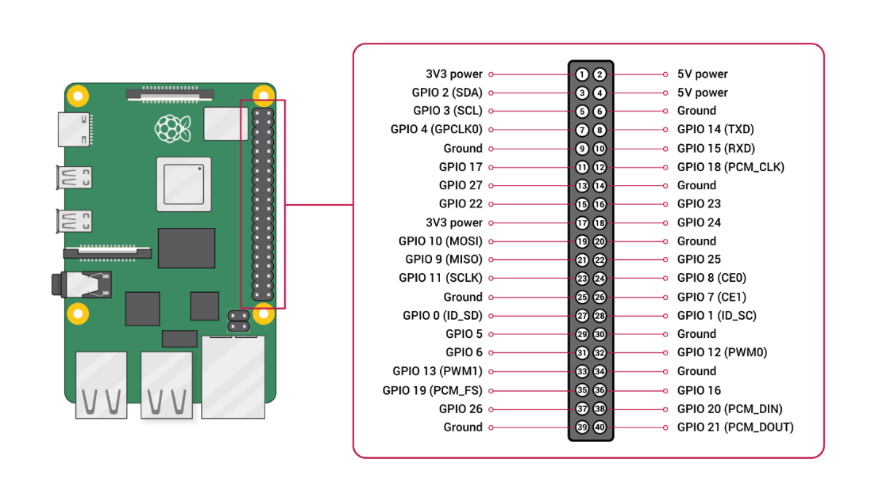
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Dip 1 | Dip 2 | Dip 3 | Dip 4 | Dip 5 | Dip 6 |
| Up (Analog input mode) | Up (Analog input mode) | Up (Power supply mode) | Down (independent mode) | Down (Exponential mode) | Down (4x sensitivity) |





### Raspberry Pi 4 B

The Raspberry Pi 4 B is at the moment the most recent version of the Raspberry Pi. The Raspberry Pi features a Quad core Cortex-A72 processor, making it a powerful tiny computer. The Raspberry Pi contains many input/outputs making it very useful for electronic prototyping. Also, MATLAB supports the Raspberry Pi with their Simulink software. To make use of the functionality of the Raspberry Pi here is the pinout:



We will be using two pins from the Raspberry namely: GPIO 13 to control the motor encoder board with PWM. And GPIO 15, the UART receiver to receive the encoder data from the Arduino.

### Raspberry Pi Arduino motor encoder readout protoshield

Initially the motor encoder was planned on being read out by the Raspberry Pi. Unfortunately I did not get this working. Therefore, I created a Arduino Protoshield to have the Arduino read out the encoder and pass the value on to the Raspberry Pi using UART. Design files of the protoboard can be found under “ElevatorControlSysFontys\6. Electrical Engineering\PCB design”. The Protoboard is powered either through the Raspberry Pi, or through the Motor control board (never both!). When powering the protoboard and Raspberry Pi from the Motor control board connect the red power cable to the PCB as shown in Figure 1.

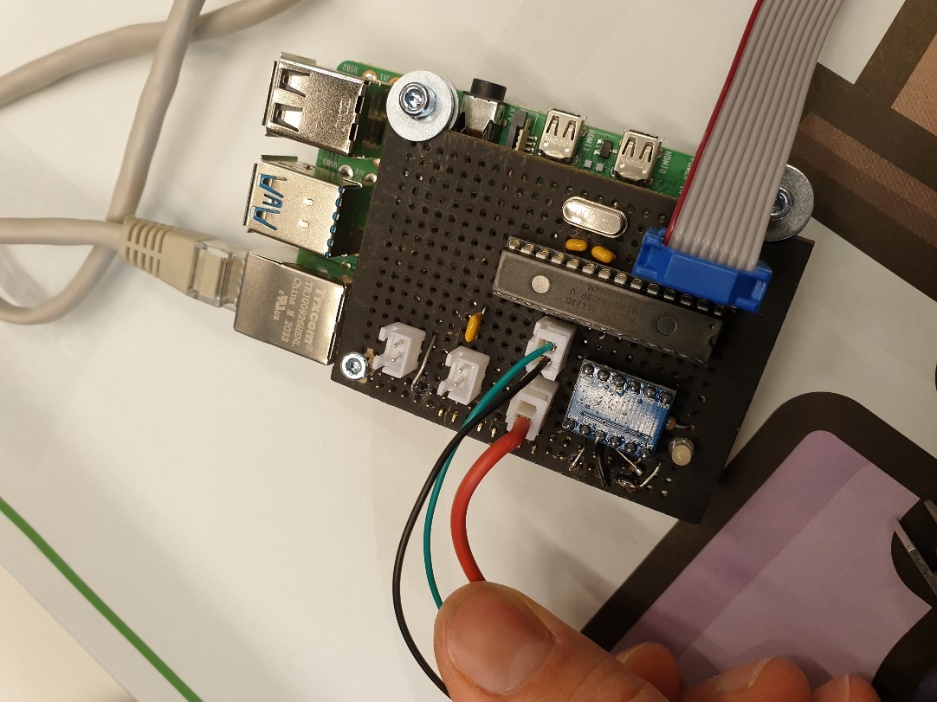


Figure 1: Raspberry Pi 4 with the Arduino motor encoder readout protoshield

### Maxon Motor (exact specifications unsure)

More details about the Maxon Motor to be added here.

### Wiring diagram

Figure 2 shows an overview of all the electrical connections made in the system.

The system is split up into 4 main parts:

* The Raspberry Pi
* Arduino Protoshield
* Sabertooth motor control board
* The Maxon Motor with Encoder.

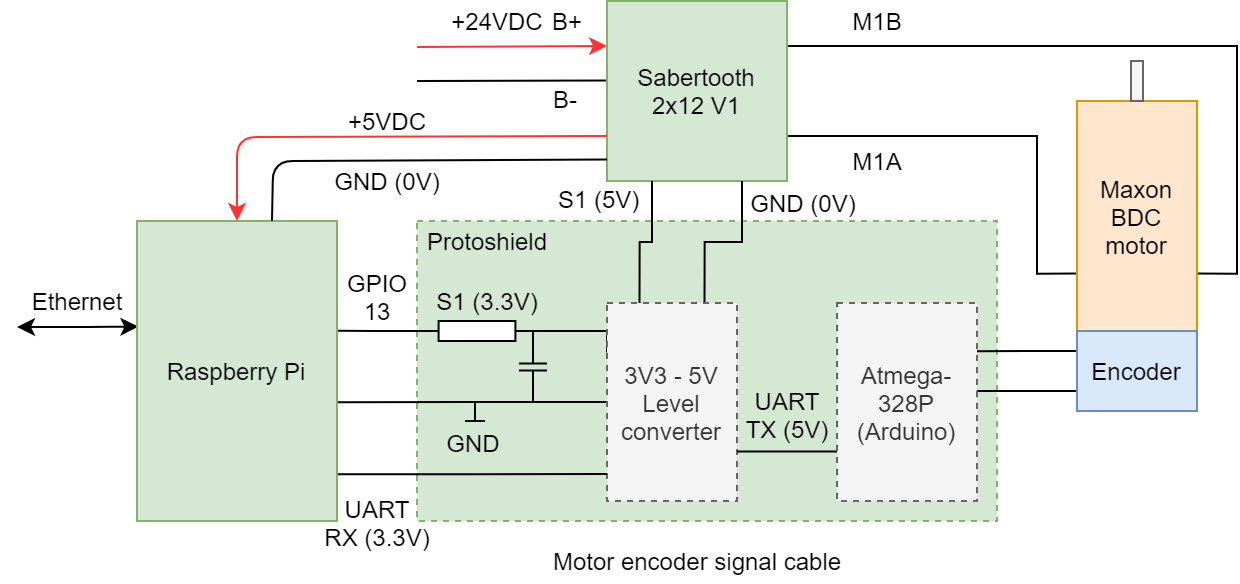


Figure 2: System electrical overview

### Setting up the system

To setup the system please refer to Appendix 3.1 How to use Simulink Support package for Rapsberry Pi.

Then power the system using a bench power supply, launch the Simulink RpiTestV5.slx file

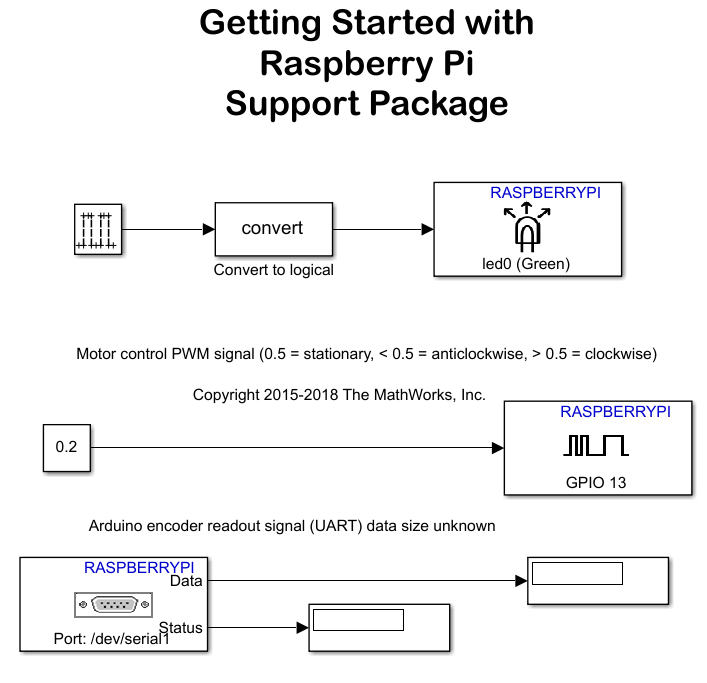
The Arduino is programmed with a simple encoder read and throughput program (1Hz, increase in future). The Atmega328p chip may be removed from the protoboard and inserted into an Arduino Uno to be programmed seperately. The Arduino software can be found in the Electrical engineering folder.

### Suggested improvements

* Increase the writing speed of the Arduino in the future when the data is correctly received by the Simulink program.
* To change from analog control to serial control a level converter is required to turn the 3.3V raspberry to 5V logic for the sabertooth (<https://www.tinytronics.nl/shop/en/dc-dc-converters/level-converters/i2c-uart-bi-directional-logic-level-converter-5v-3.3v-2-channel-with-supply>)

## Software

The software is made with Simulink and currently looks like this:



The most important functionality is the PWM control signal. A signal between 0 – 1 creates a PWM signal with a duty cycle of 0 – 100% respectively. The PWM signal is then filtered and instructs the motor control board what speed and direction to control the motors.

The Serial read block should receive the data from the Arduino, currently it does seem to receive the data at the receiver but the data does not represent the correct encoder value yet.

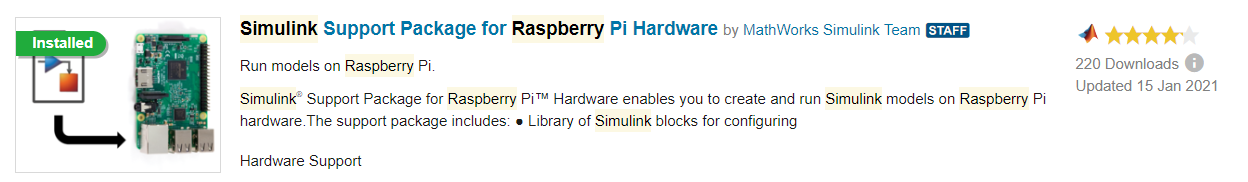
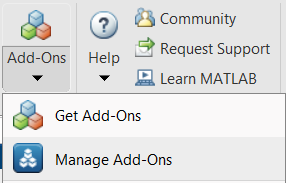
# Appendix

## How to use Simulink Support Package for Raspberry Pi Hardware

### Setup Raspberry Pi hardware

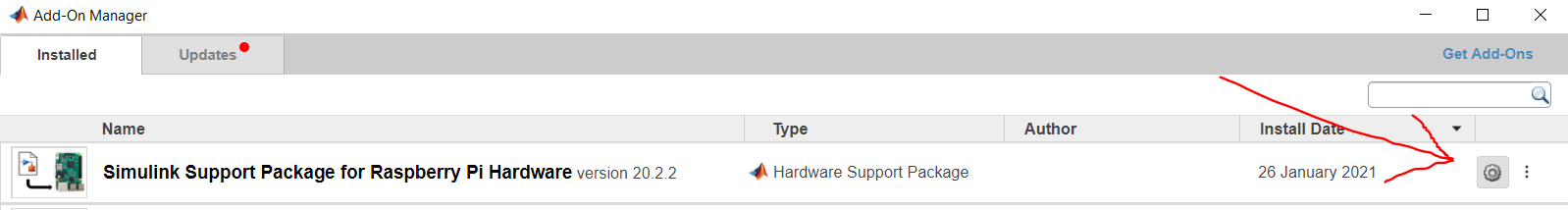
Open MATLAB and go to AddiOns:

Search for Simulink Raspberry and install the following package:

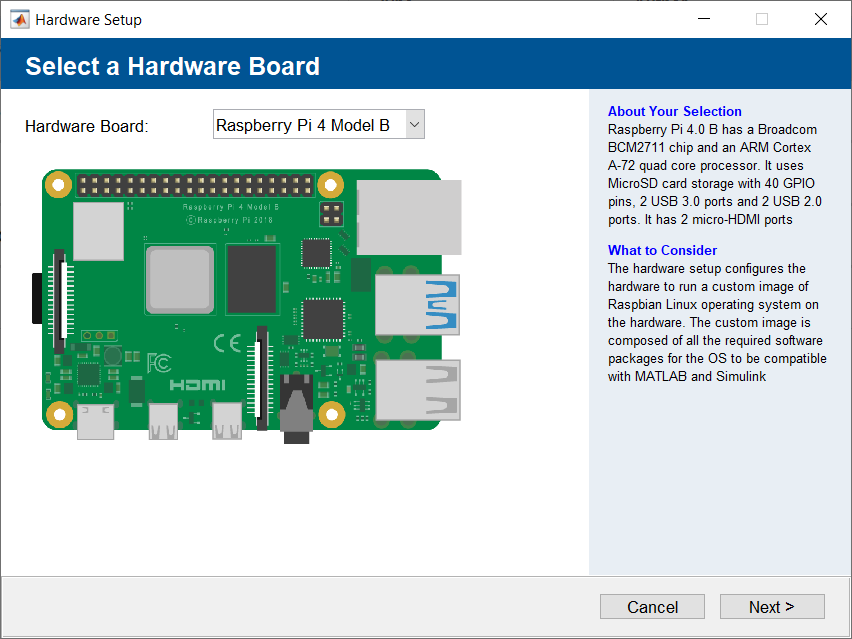


Go back to Add-Ons and select Manage Add-Ons:

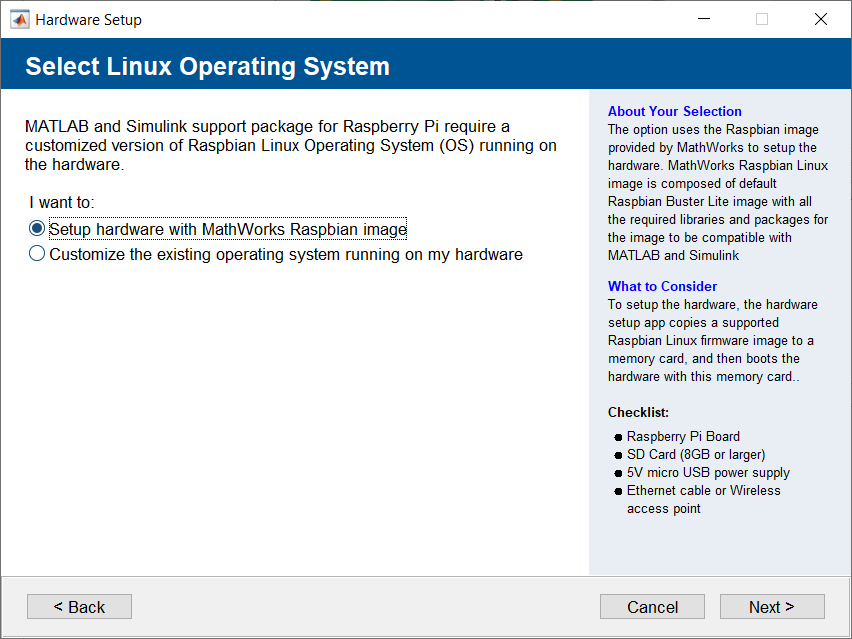
Click the setup icon on the right:



Select the Raspberry Pi 4 Model B and click Next >

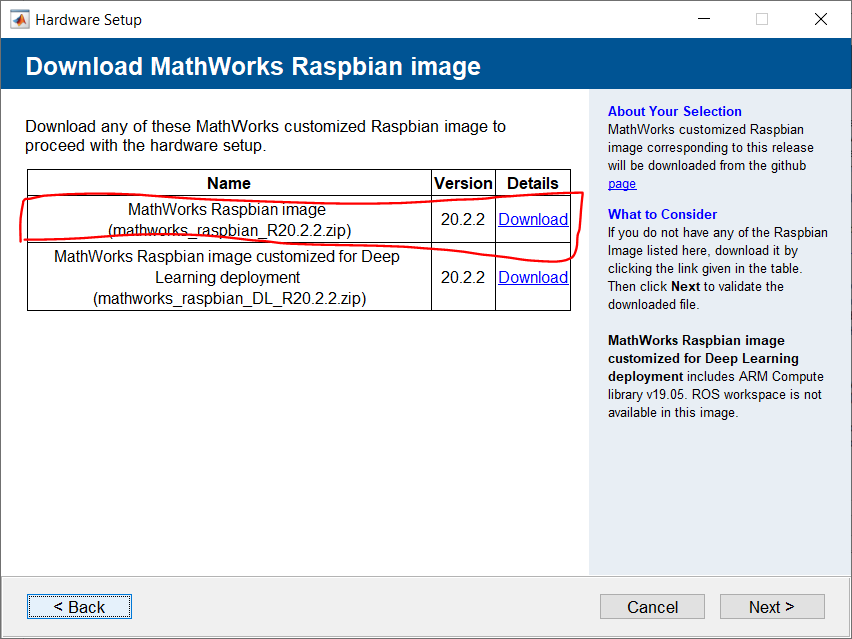


Setup hardware with MathWorks Raspbian image should be selected:

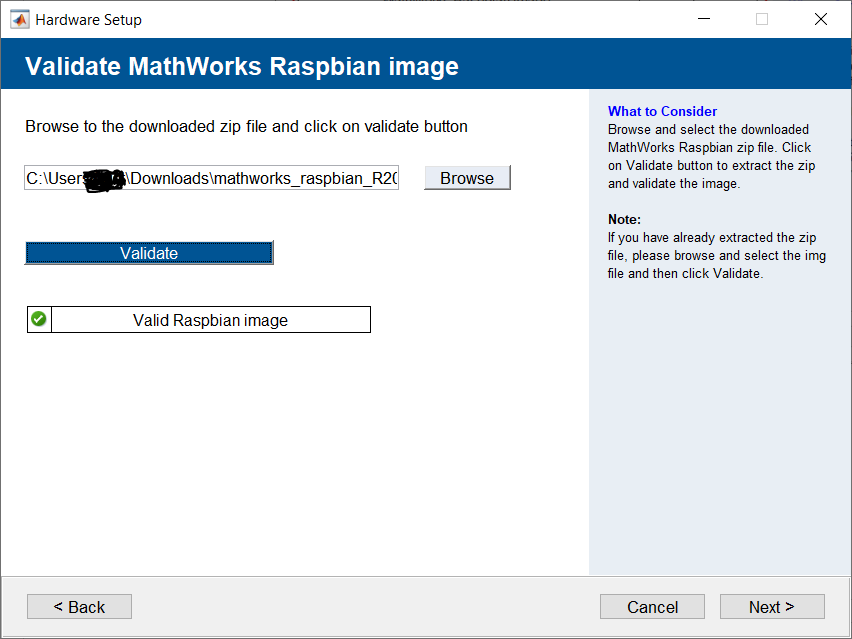


Click next >

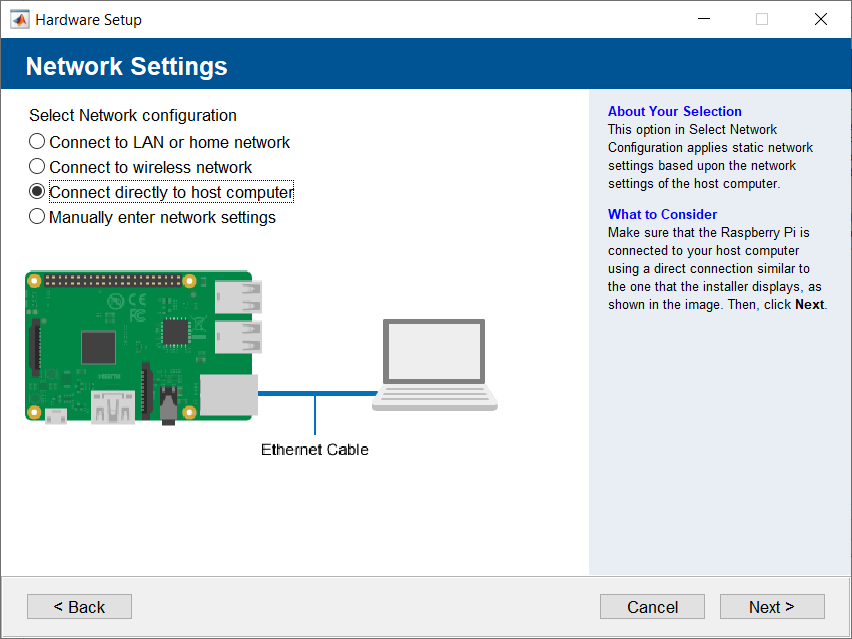
Download the regular image:



Browse for the image in your downloads and validate it:

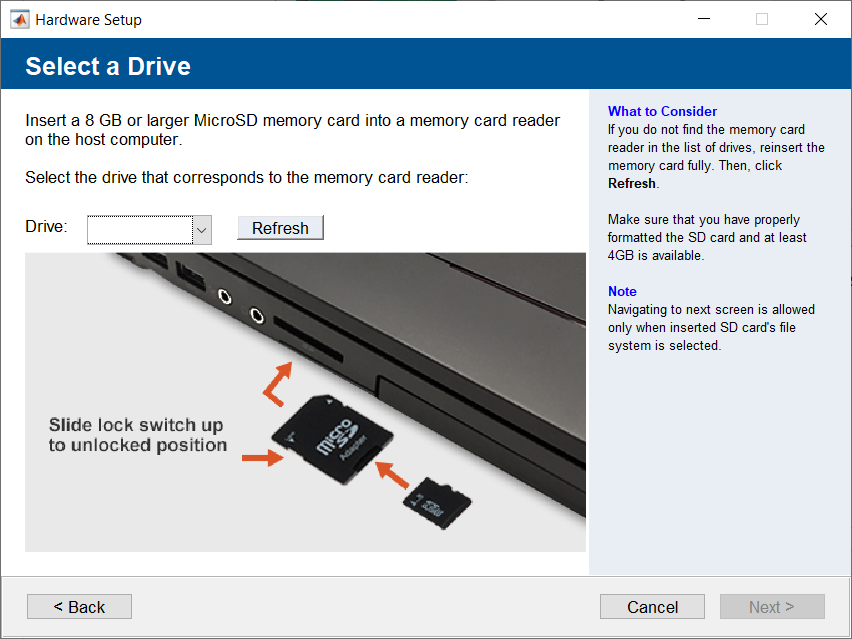


Select the Connect directly to host computer option.



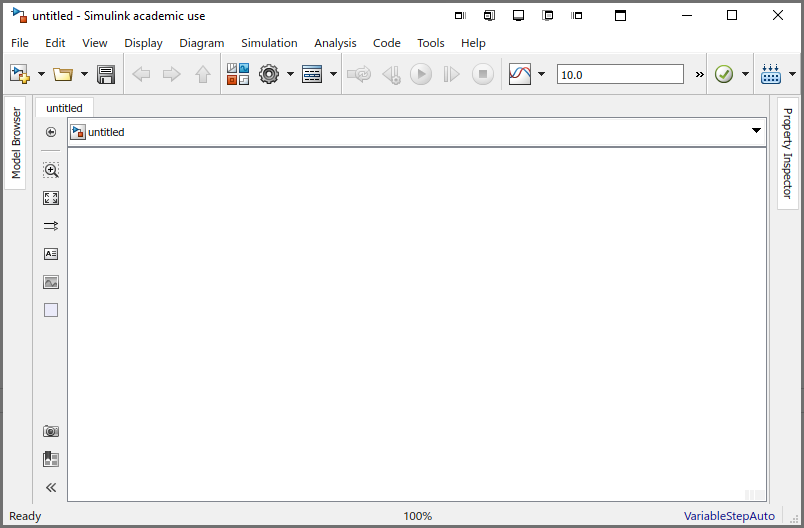
I have also successfully teted the connect to wireless network option but I advise to use an ethernet cable for debugging purposes (ssh connection).

Select the correct SD card (careful not to select the wrong one as the sd card will be formatted!)



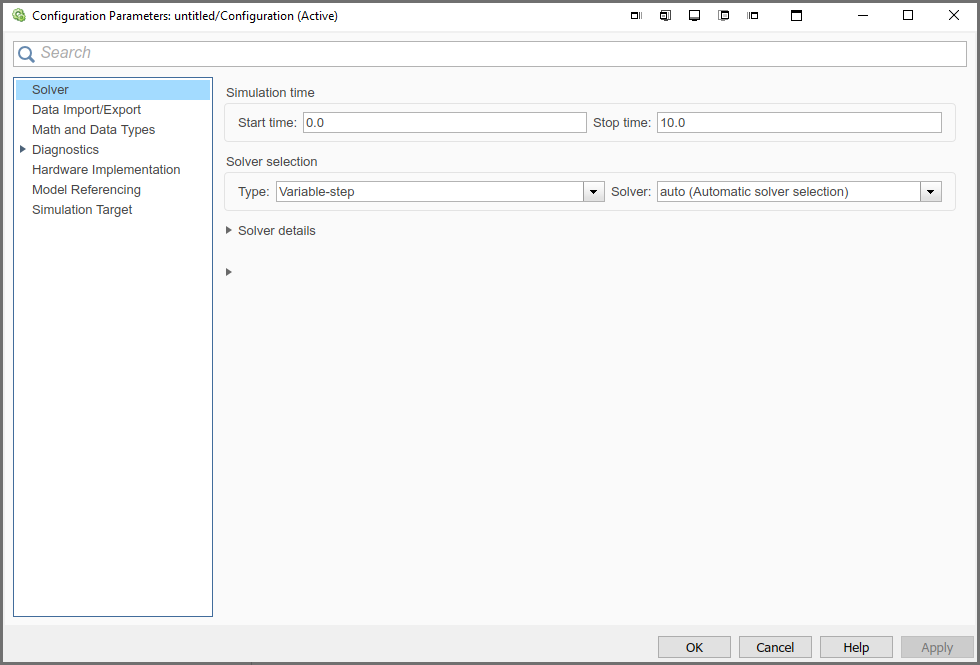
After installing the MathWorks Raspbian on the sd card you may click Next >(leave the sd card in the laptop) after clicking Next > the sd card will be done and you will be prompted to remove the sd card, insert it into the Raspberry Pi. Then power the Raspberry Pi and connect it with an ethernet cable to the laptop. The program will then detect your Raspberry Pi and the setup will be completed.

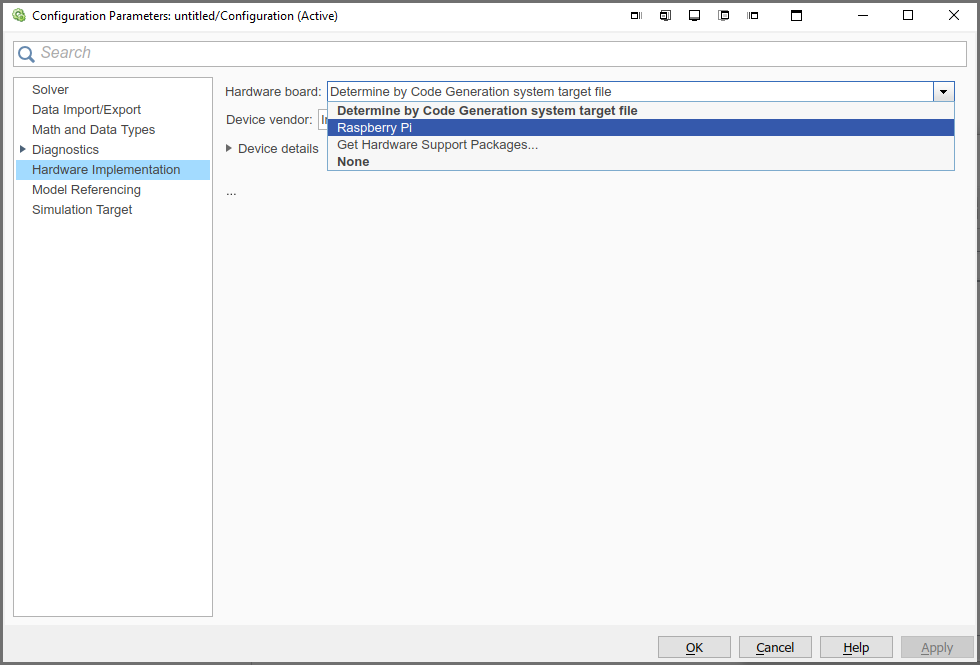
### Select hardware (after reopening project)



Go to Hardware Implementation

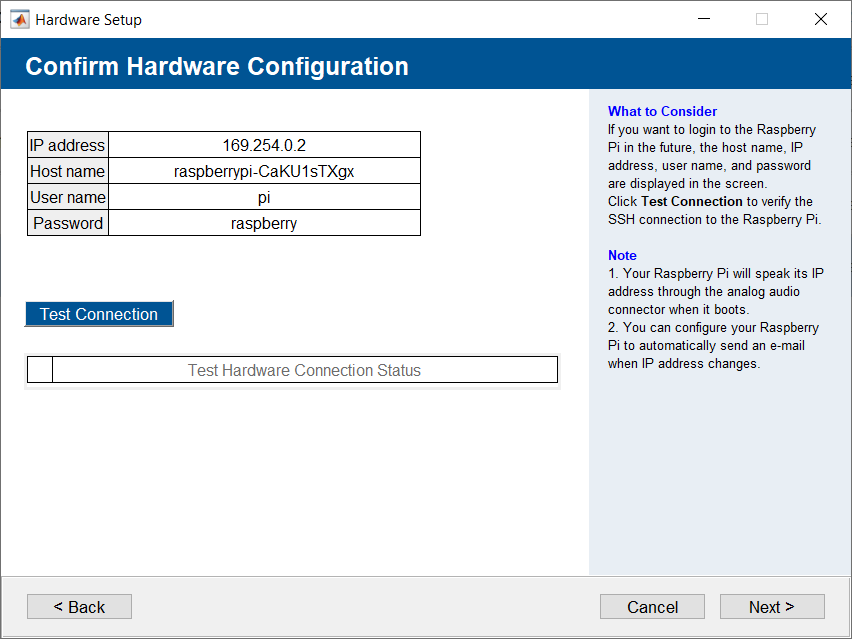
Open the configuration parameters





Select Raspberry Pi

## Hardware setup in Simulink:



Debugging the raspberry pi or clearing Matlab files:

* Launch Putty and connect with ssh to the raspberry entering the Raspberry Pi address

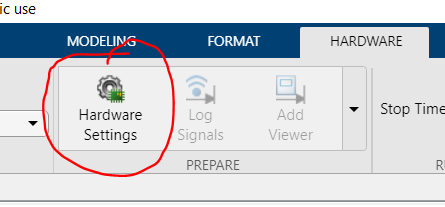
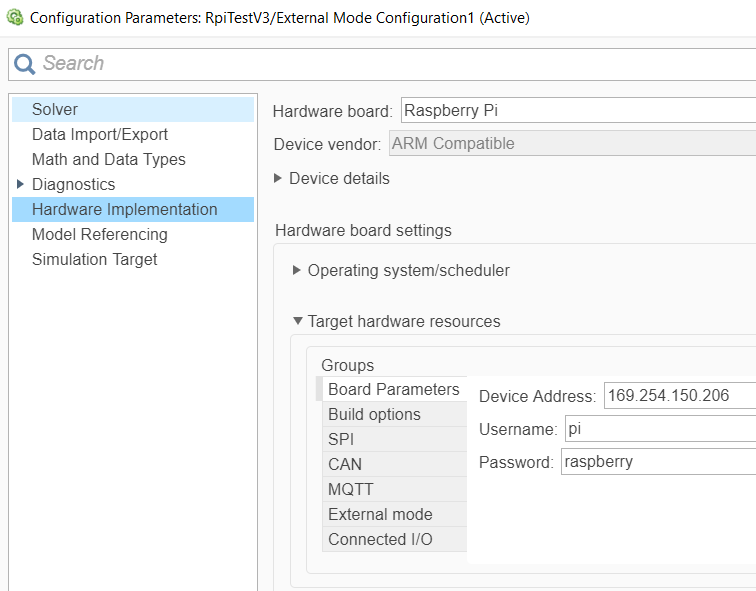
## F.A.Q.

**Q: Your current working folder is: C:\Program Files\MATLAB\R2018b\bin Simulink does not permit you to modify the MATLAB installation area. Please change to a working folder that is not in the MATLAB installation area and re-try the previous command.**

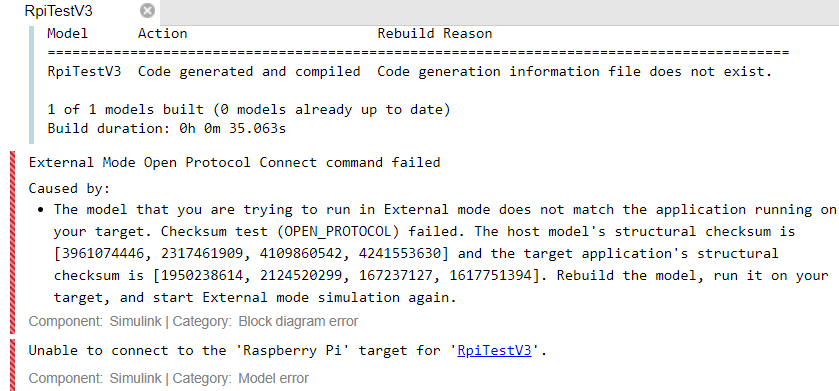
S: Your working directory in MATLAB is set to a folder that is written only. Please go to the Matlab command line and enter cd(tempdir) to change the working folder to the tempdir (don’t forget to save your files properly!) <https://nl.mathworks.com/matlabcentral/answers/443075-your-current-working-folder-is-c-program-files-matlab-r2018b-bin-simulink-does-not-permit-you-to>

**Q: Where can I find the Raspberry Pi hardware configuration settings?**

S: after you have set up the hardware as described in this document you may click the Hardware settings button under the hardware tab:

And then go to Hardware implementation 🡪 Target hardware resources.

Q: Error 

S: Reinstall Image on Raspberry Pi. (Seek better solution to solve this regularly occurring error.)