HOW TO SETUP PI FOR MATLAB/SIMULINK

BONUS MATERIAL: ACADOS

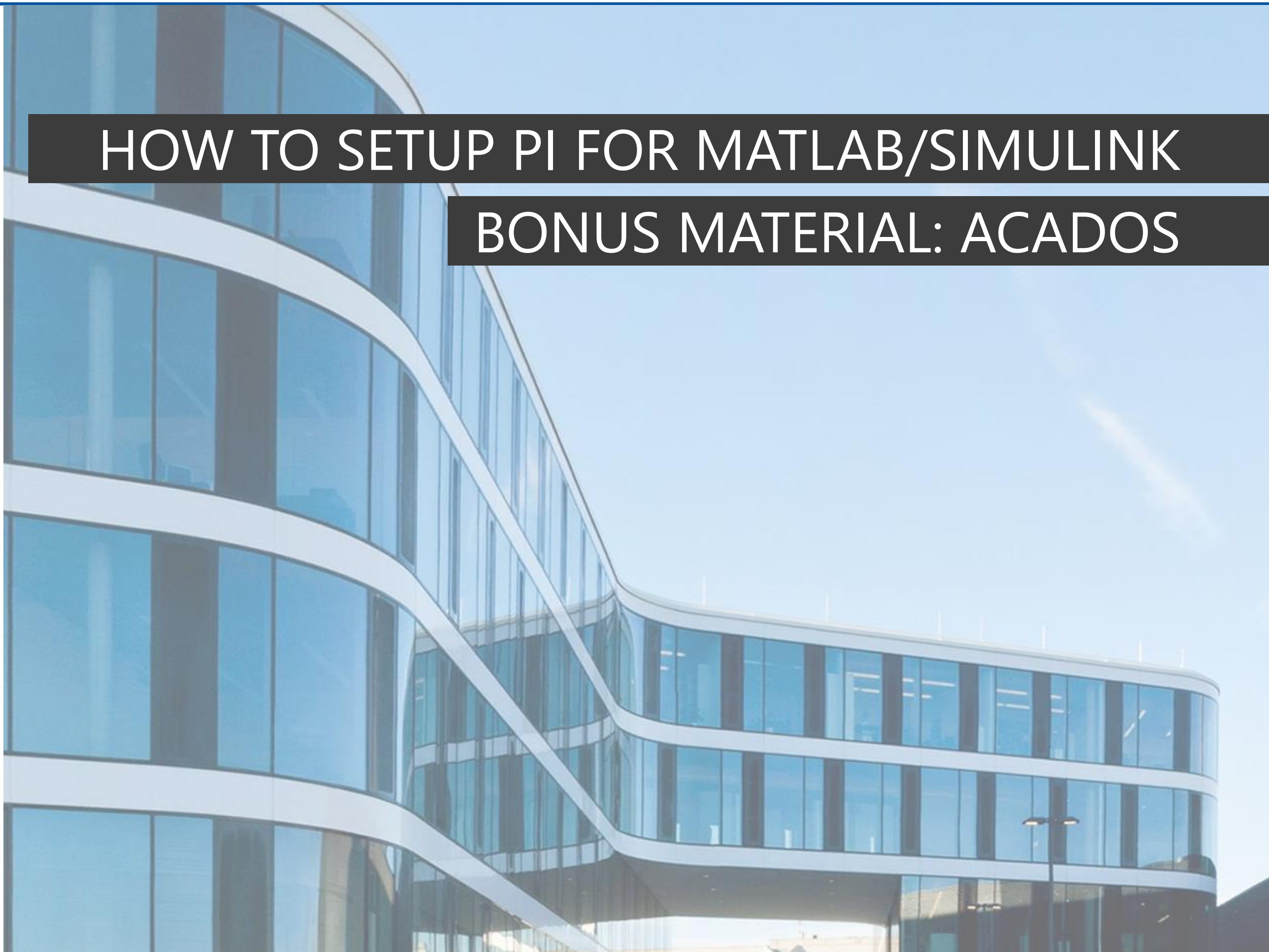
EDMONTON, AB, CA. 28.06.2022

ALEXANDER WINKLER

LEHR- UND FORSCHUNGSGEBIET
MECHATRONIK IN MOBILEN ANTRIEBEN

PREPARED FOR

UOFA AND RWTH STUDENT



AGENDA

Setup matlab and simulink for raspberry pi

Build and run the model on pi

Clone and build **acados** on raspi

Clone and build **tensorflow lite** on raspi (new 5.7.2022)

Setup simulink models for use with pi

Add **custom code** for acados binaries

Build and **run the model on pi**

Run **simulink** model with **tensorflow on pi** (new 5.7.2022)

Setup **CAN** communication on **Pi** (new 18.7.2022)

Setup **CAN** communication on **MABX** (new 18.7.2022)

Setup **UDP** communication on **Pi** (new 18.7.2022)

Setup **UDP** communication on **MABX** (new 18.7.2022)

Linux / pi **basics**

Overclock raspi

Additional information

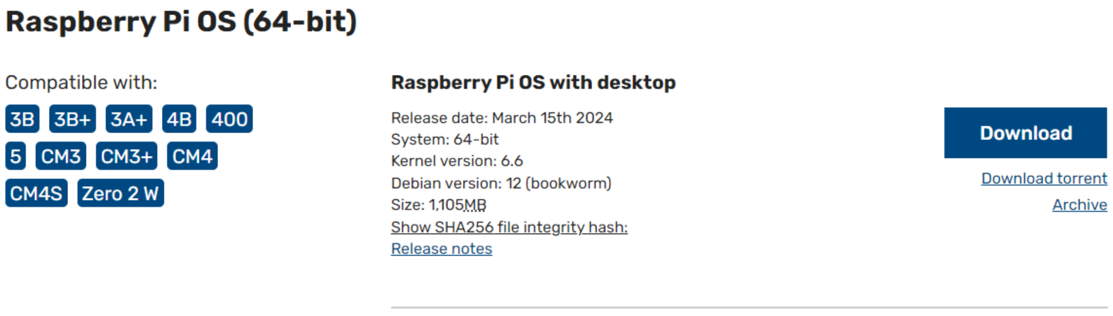
Setup Matlab and Simulink for Raspberry Pi – Step 1

Choose 32 or 64 bit! Both work! Bookworm version 12 tested.

https://www.raspberrypi.com/software/operating-systems/#raspberry-pi-os-64-bit

Flash SD Card with Tool of choice, e.g.: IMAGE USB

https://www.osforensics.com/tools/write-usb-images.html



Raspberry Pi OS (64-bit)

Compatible with:

3B 3B+ 3A+ 4B 400
5 CM3 CM3+ CM4
CM4S Zero 2 W

Raspberry Pi OS with desktop

Release date: March 15th 2024
System: 64-bit
Kernel version: 6.6
Debian version: 12 (bookworm)
Size: 1,105MB
Show SHA256 file integrity hash:
Release notes

Download

Download torrent

Archive

Run **mathworks setup** (next slides, step 3)

If this setup fails: check libraries individually on the Pi itself!

https://github.com/mathworks/Raspbian_OS_Setup

Either way works.

The matlab prepared image is 32 bits though

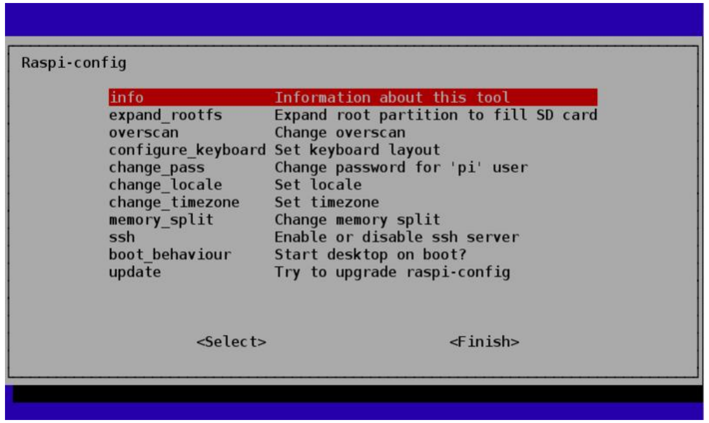
64 bits works, also with acados!

Setup Matlab and Simulink for Raspberry Pi – Step 2

After flash **enable SSH and VNC** in interface of raspi-config

-> access the Pi with SSH (matlab and cmd) or VNC (VNC Viewer, grapghical interface)

Terminal: sudo raspi-config

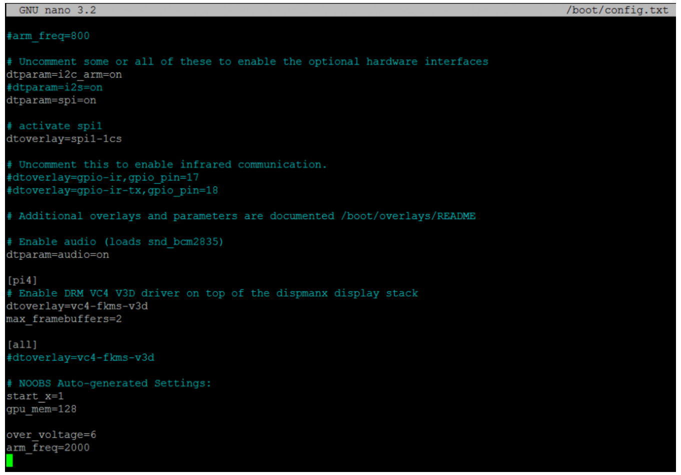


```
Raspi-config

        info                Information about this tool
        expand_rootfs       Expand root partition to fill SD card
        overscan            Change overscan
        configure_keyboard  Set keyboard layout
        change_pass         Change password for 'pi' user
        change_locale       Set locale
        change_timezone     Set timezone
        memory_split        Change memory split
        ssh                 Enable or disable ssh server
        boot_behaviour      Start desktop on boot?
        update              Try to upgrade raspi-config


              <Select>                     <Finish>
```

Don’t forget to manipulate the boot config for fixing the cpu speed, if you want to use the pi as a controller!

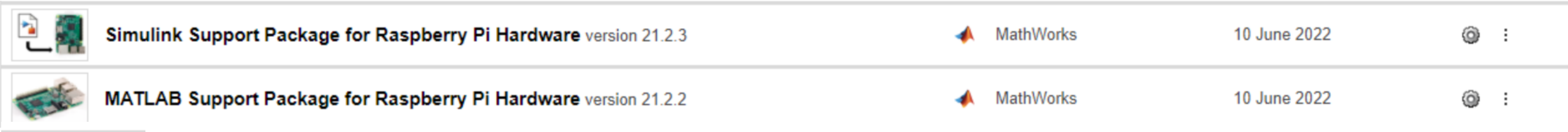
Change Frequency:
sudo nano /boot/config.txt
over_voltage=6
arm_freq=2000
force_turbo=1 (fixes the speed to the overclocked speed above)
Ctrl+X plus Y and Enter to save and overwrite.

See other slides in the back for more information!



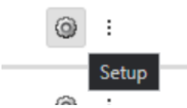
```
  GNU nano 3.2                                                        /boot/config.txt

#arm_freq=800

# Uncomment some or all of these to enable the optional hardware interfaces
dtparam=i2c_arm=on
#dtparam=i2s=on
dtparam=spi=on

# activate spi1
dtoverlay=spi1-1cs

# Uncomment this to enable infrared communication.
#dtoverlay=gpio-ir,gpio_pin=17
#dtoverlay=gpio-ir-tx,gpio_pin=18

# Additional overlays and parameters are documented /boot/overlays/README

# Enable audio (loads snd_bcm2835)
dtparam=audio=on

[pi4]
# Enable DRM VC4 V3D driver on top of the dispmanx display stack
dtoverlay=vc4-fkms-v3d
max_framebuffers=2

[all]
#dtoverlay=vc4-fkms-v3d

# NOOBS Auto-generated Settings:
start_x=1
gpu_mem=128

over_voltage=6
arm_freq=2000
```

Setup Matlab and Simulink for Raspberry Pi – Step 3

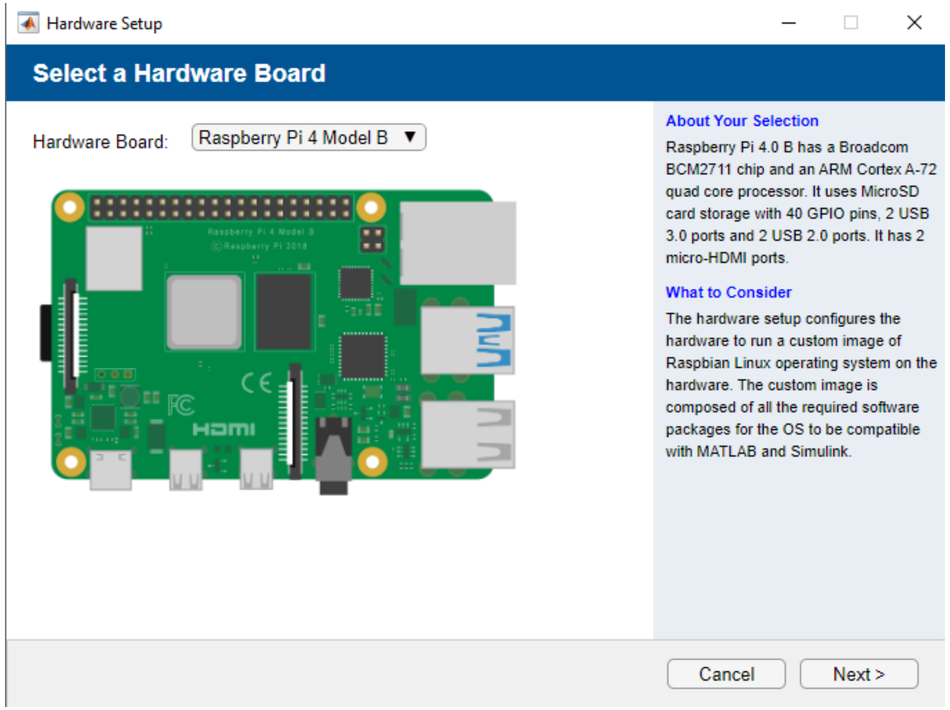


Simulink Support Package for Raspberry Pi Hardware version 21.2.3 MathWorks 10 June 2022

MATLAB Support Package for Raspberry Pi Hardware version 21.2.2 MathWorks 10 June 2022



If this setup fails: check libraries individually on the Pi itself! https://github.com/mathworks/Raspbian_OS_Setup



Select a Hardware Board

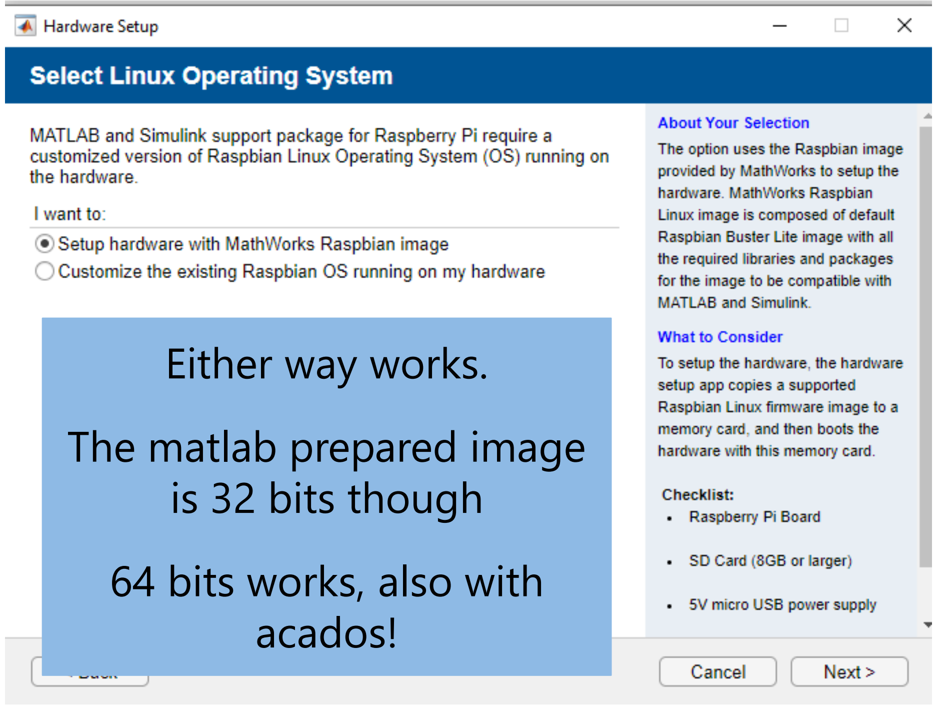
Hardware Board: Raspberry Pi 4 Model B ▼

About Your Selection

Raspberry Pi 4.0 B has a Broadcom BCM2711 chip and an ARM Cortex A-72 quad core processor. It uses MicroSD card storage with 40 GPIO pins, 2 USB 3.0 ports and 2 USB 2.0 ports. It has 2 micro-HDMI ports.

What to Consider

The hardware setup configures the hardware to run a custom image of Raspbian Linux operating system on the hardware. The custom image is composed of all the required software packages for the OS to be compatible with MATLAB and Simulink.



Select Linux Operating System

MATLAB and Simulink support package for Raspberry Pi require a customized version of Raspbian Linux Operating System (OS) running on the hardware.

I want to:

● Setup hardware with MathWorks Raspbian image

○ Customize the existing Raspbian OS running on my hardware

About Your Selection

The option uses the Raspbian image provided by MathWorks to setup the hardware. MathWorks Raspbian Linux image is composed of default Raspbian Buster Lite image with all the required libraries and packages for the image to be compatible with MATLAB and Simulink.

What to Consider

To setup the hardware, the hardware setup app copies a supported Raspbian Linux firmware image to a memory card, and then boots the hardware with this memory card.

Checklist:

- Raspberry Pi Board
- SD Card (8GB or larger)
- 5V micro USB power supply

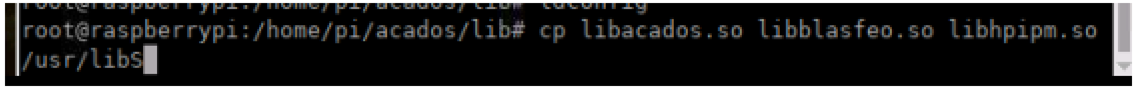
Either way works.

The matlab prepared image is 32 bits though

64 bits works, also with acados!

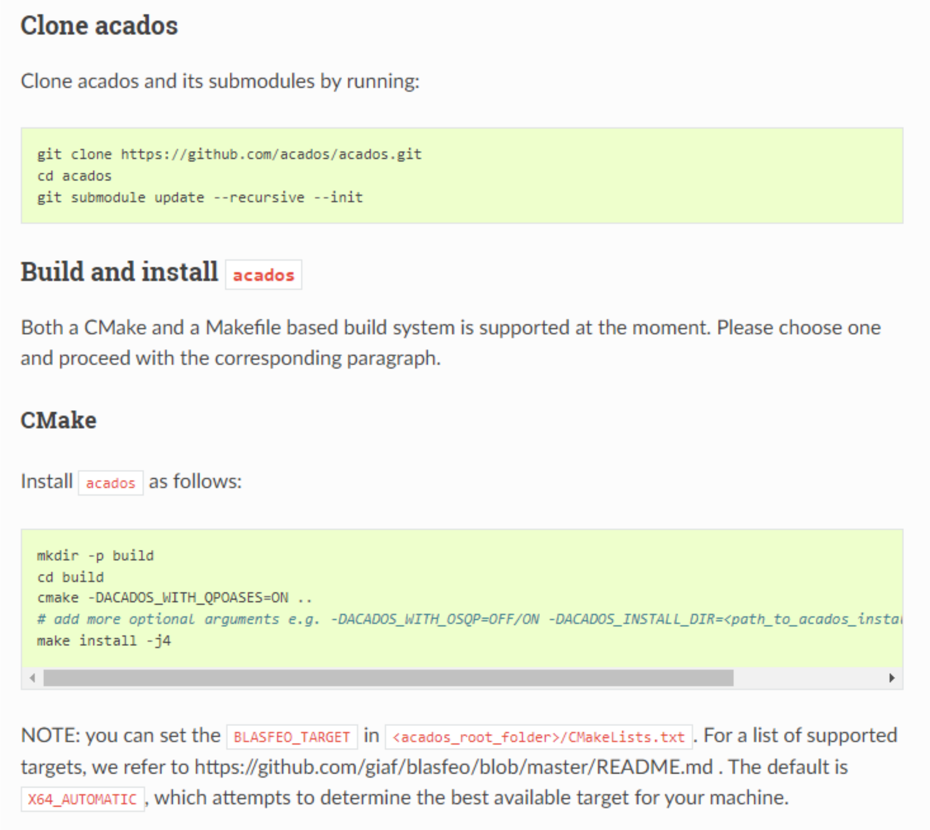
Clone and build acados on RasPi

- Enable filesystem expand to enable writing on SD card with terminal (open at specific folder with “F4”) “sudo raspi-config”: Navigate to Advanced Options and to Expand Filesystem
- Follow the inctructions: https://docs.acados.org/installation/index.html
- Use “sudo su” in the terminal to be the root user with all rights
- After successfully installing the libraries, copy the libs to standard folder for libs (/usr/lib) and add the path to the Pi’s system library path for shared libraries:
  - In the lib folder with root: Cp libacados.so libblasfeo.so libhpipm.so /usr/lib



```
root@raspberrypi:/home/pi/acados/lib# ldconfig
root@raspberrypi:/home/pi/acados/lib# cp libacados.so libblasfeo.so libhpipm.so
/usr/lib$
```

  - In the same folder with root rights: ldconfig



Clone acados

Clone acados and its submodules by running:

```
git clone https://github.com/acados/acados.git
cd acados
git submodule update --recursive --init
```

Build and install `acados`

Both a CMake and a Makefile based build system is supported at the moment. Please choose one and proceed with the corresponding paragraph.

CMake

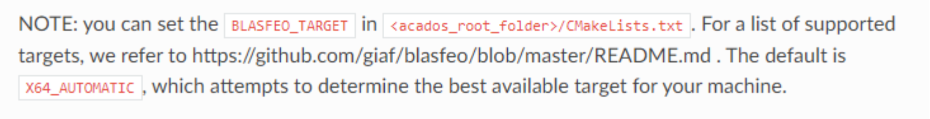
Install `acados` as follows:

```
mkdir -p build
cd build
cmake -DACADOS_WITH_QPOASES=ON ..
# add more optional arguments e.g. -DACADOS_WITH_OSQP=OFF/ON -DACADOS_INSTALL_DIR=<path_to_acados_insta
make install -j4
```

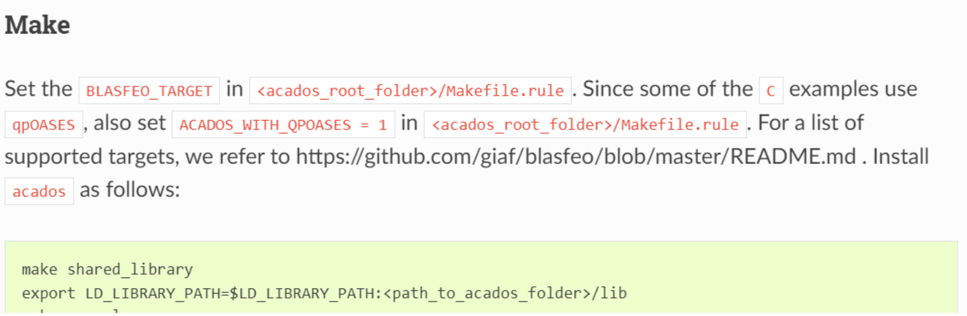
NOTE: you can set the `BLASFEO_TARGET` in `<acados_root_folder>/CMakeLists.txt`. For a list of supported targets, we refer to https://github.com/giaf/blasfeo/blob/master/README.md . The default is `X64_AUTOMATIC`, which attempts to determine the best available target for your machine.

Clone and build acados on RasPi - Debugging

- Set the Blasefeo_Target or HPIPM_Target manually if needed and the errors appear:
  - Set Blasfeo_Target to ARMV8A_ARM_CORTEX_A57 for Raspberry Pi 4, 4B, 400 or 5
  - Set HPIPM_Target to GENERIC for Raspberry Pi 4, 4B, 400 or 5
- Execute CMAKE steps in the installation guide. If Cmake doesn't work, try make:



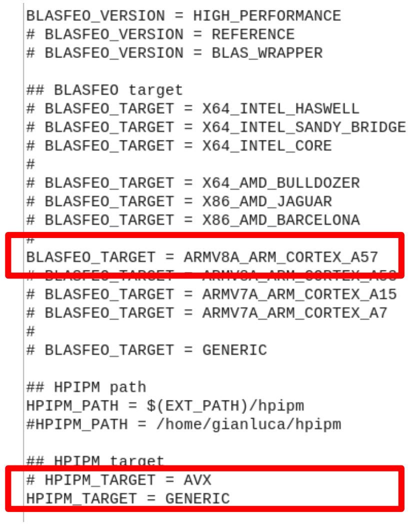
NOTE: you can set the `BLASFEO_TARGET` in `<acados_root_folder>/CMakeLists.txt`. For a list of supported targets, we refer to https://github.com/giaf/blasfeo/blob/master/README.md . The default is `X64_AUTOMATIC`, which attempts to determine the best available target for your machine.



Make

Set the `BLASFEO_TARGET` in `<acados_root_folder>/Makefile.rule`. Since some of the `c` examples use `qpOASES`, also set `ACADOS_WITH_QPOASES = 1` in `<acados_root_folder>/Makefile.rule`. For a list of supported targets, we refer to https://github.com/giaf/blasfeo/blob/master/README.md . Install `acados` as follows:

```
make shared_library
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:<path_to_acados_folder>/lib
```



```
BLASFEO_VERSION = HIGH_PERFORMANCE
# BLASFEO_VERSION = REFERENCE
# BLASFEO_VERSION = BLAS_WRAPPER

## BLASFEO target
# BLASFEO_TARGET = X64_INTEL_HASWELL
# BLASFEO_TARGET = X64_INTEL_SANDY_BRIDGE
# BLASFEO_TARGET = X64_INTEL_CORE
#
# BLASFEO_TARGET = X64_AMD_BULLDOZER
# BLASFEO_TARGET = X86_AMD_JAGUAR
# BLASFEO_TARGET = X86_AMD_BARCELONA
#
BLASFEO_TARGET = ARMV8A_ARM_CORTEX_A57
# BLASFEO_TARGET = ARMV8A_ARM_CORTEX_A53
# BLASFEO_TARGET = ARMV7A_ARM_CORTEX_A15
# BLASFEO_TARGET = ARMV7A_ARM_CORTEX_A7
#
# BLASFEO_TARGET = GENERIC

## HPIPM path
HPIPM_PATH = $(EXT_PATH)/hpipm
#HPIPM_PATH = /home/gianluca/hpipm

## HPIPM target
# HPIPM_TARGET = AVX
HPIPM_TARGET = GENERIC
```

For Raspbian on Raspberry Pi 5 with Bookworm version and gcc 11/12 only the make file worked, not the Cmake workflow.

Clone and build acados on RasPi - Debugging

- Be sure to use the identical acados commit / version on Pi and Host PC / Windows!

Clone and build TF Lite on RasPi

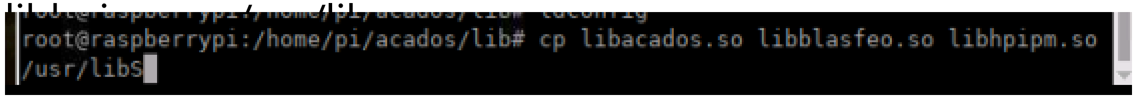
- Enable filesystem expand to enable writing on SD card with terminal “sudo raspi-config”:
  Navigate to Advanced Options and to Expand Filesystem

- Follow the inctructions: https://qengineering.eu/install-tensorflow-2-lite-on-raspberry-pi-4.html

- Use “sudo su” in the terminal to be the root user with all rights

- After successfully installing the libraries, copy the libs to standard folder for libs (/usr/lib) and add the path to the Pi’s system library path for shared libraries:

  - In the lib folder with root: Cp libacados.so libblasfeo.so



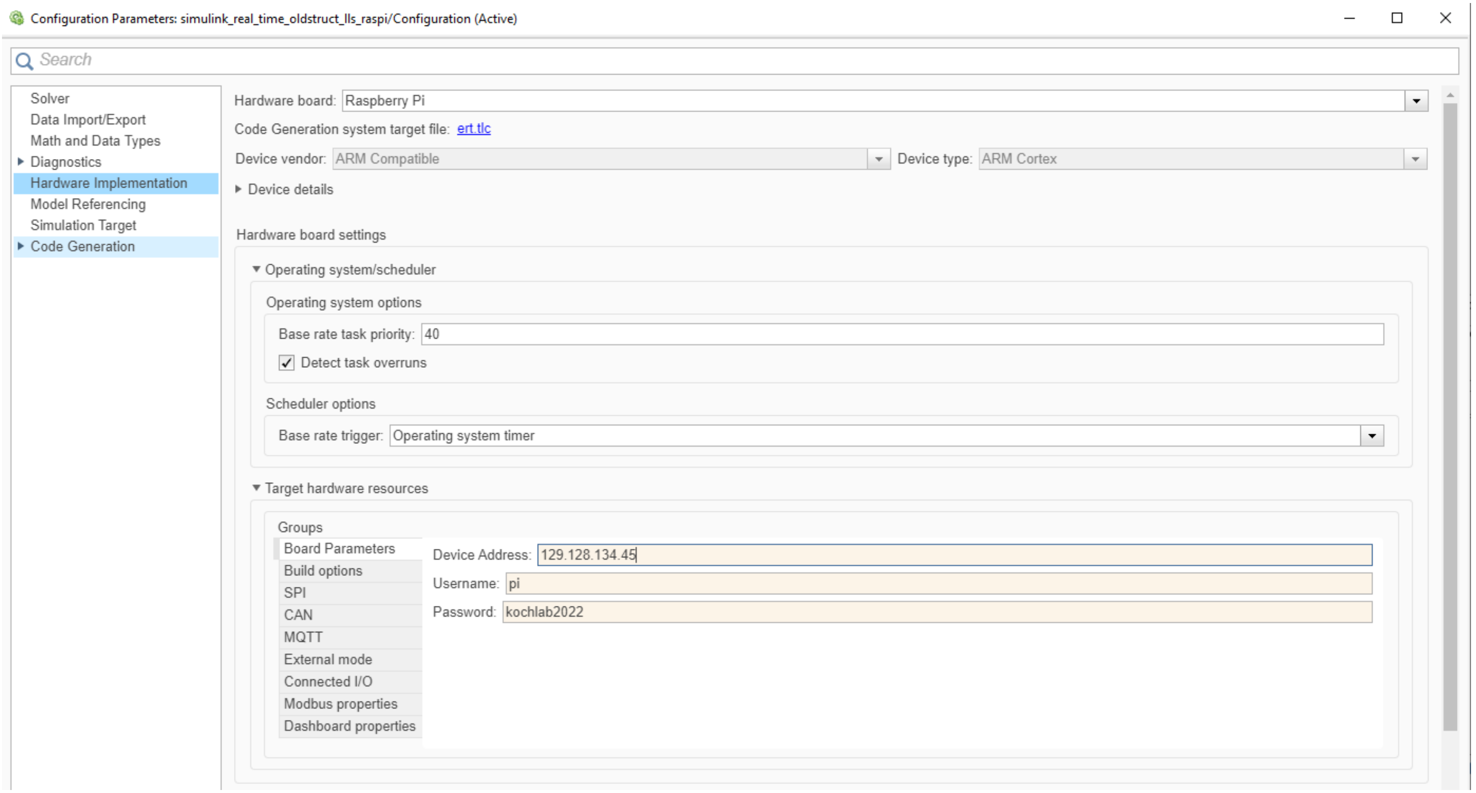
```
root@raspberrypi:/home/pi/acados/lib# ldconfig
root@raspberrypi:/home/pi/acados/lib# cp libacados.so libblasfeo.so libhpipm.so
/usr/libS
```

  - In the same folder with root rights: ldconfig

Be sure to be in the c_generated_code folder on the matlab host-pc with matlab root

Setup Simulink Models for Use With Pi 1

- Get IP address of the Pi with “ifconfig” in the terminal
- Configure the Simulink model with “Ctrl+E”. Choose the Hardware board in Hardware Implementation



Configuration Parameters: simulink_real_time_oldstruct_lls_raspi/Configuration (Active)

Hardware board: Raspberry Pi

Code Generation system target file: ert.tlc

Device vendor: ARM Compatible

Device type: ARM Cortex

▸ Device details

Hardware board settings

▾ Operating system/scheduler

Operating system options

Base rate task priority: 40

☑ Detect task overruns

Scheduler options

Base rate trigger: Operating system timer

▾ Target hardware resources

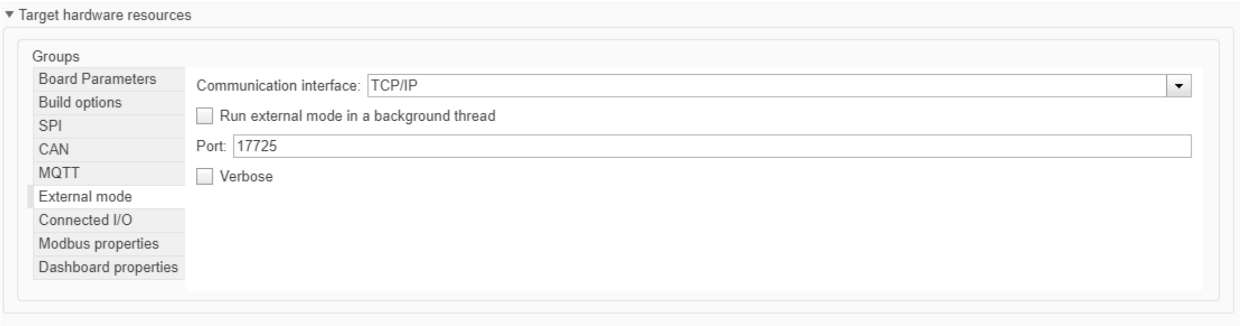
Device Address: 129.128.134.45

Username: pi

Password: kochlab2022

Be sure to be in the c_generated_code folder on the matlab host-pc with matlab root

Setup Simulink Models for Use With Pi 2

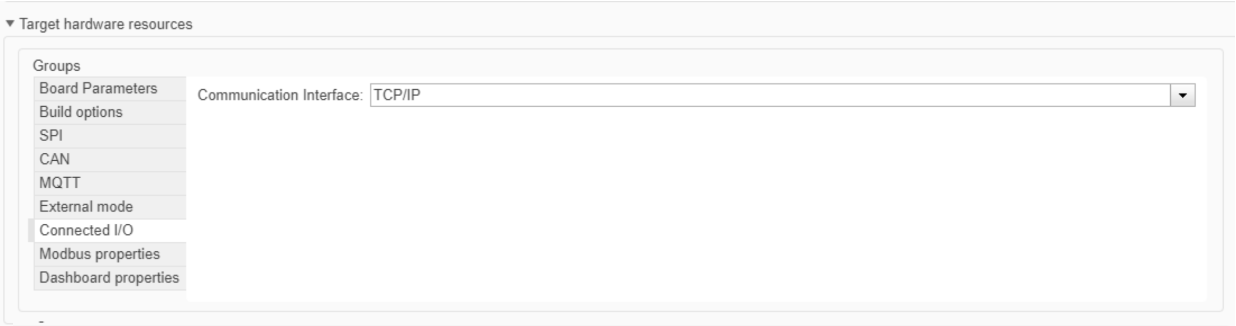


Communication interface: TCP/IP

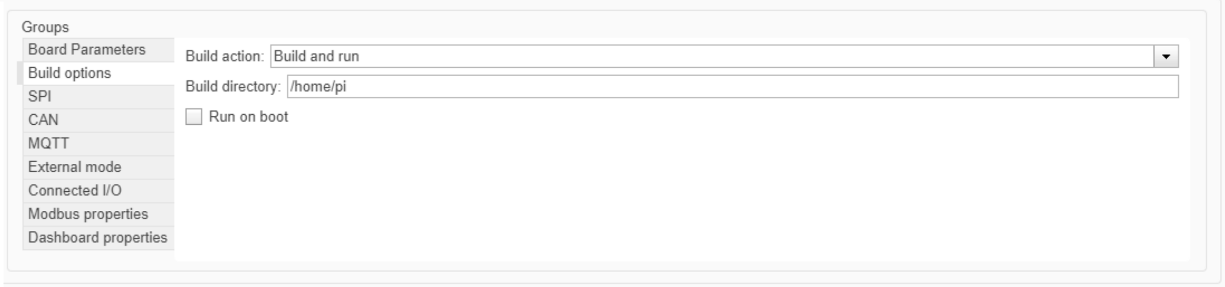
☐ Run external mode in a background thread

Port: 17725

☐ Verbose



Communication Interface: TCP/IP



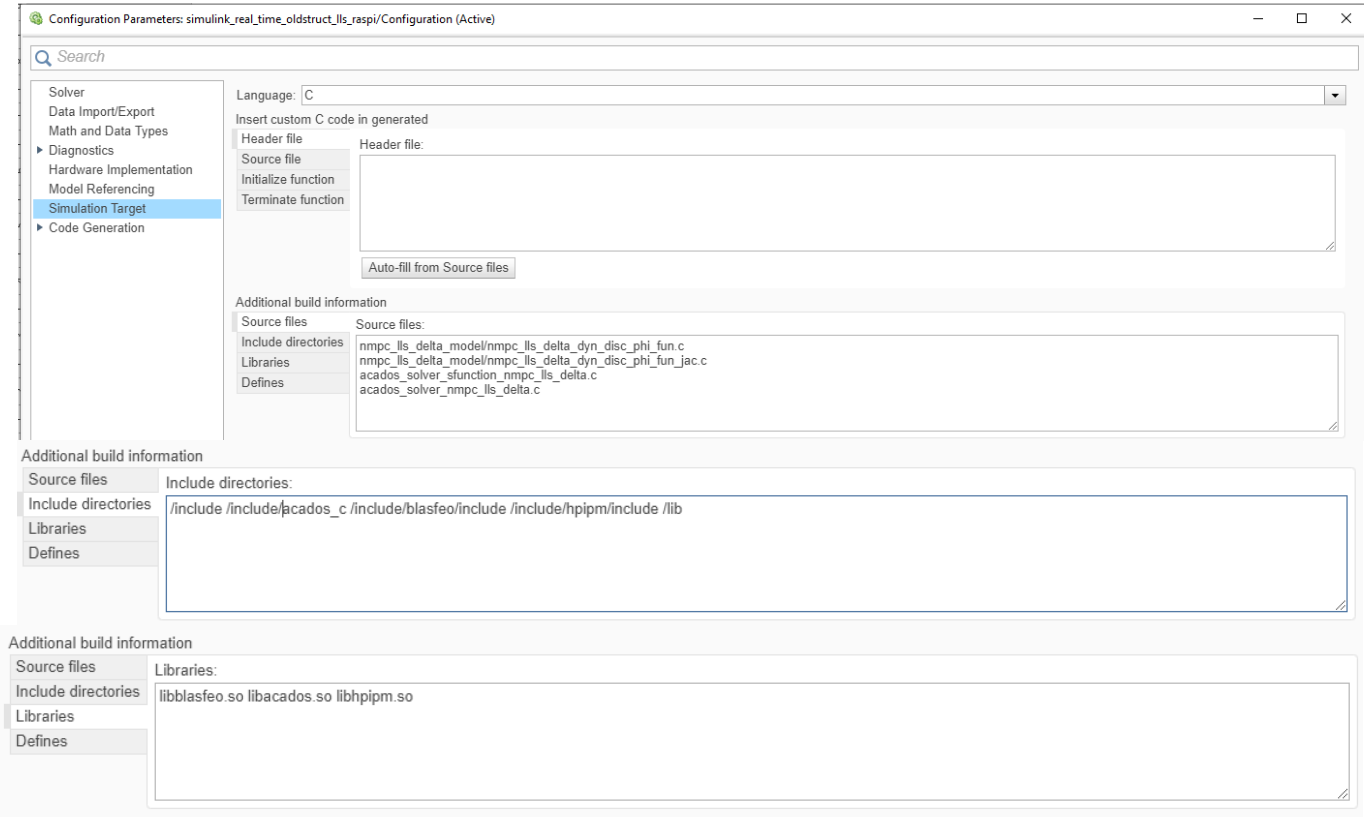
Build action: Build and run

Build directory: /home/pi

☐ Run on boot

Add Custom Code for acados binaries

Be sure to be in the c_generated_code folder on the matlab host-pc with matlab root



Configuration Parameters: simulink_real_time_oldstruct_lls_raspi/Configuration (Active)

Language: C

Insert custom C code in generated

Header file:

Additional build information

Source files:

```
nmpc_lls_delta_model/nmpc_lls_delta_dyn_disc_phi_fun.c
nmpc_lls_delta_model/nmpc_lls_delta_dyn_disc_phi_fun_jac.c
acados_solver_sfunction_nmpc_lls_delta.c
acados_solver_nmpc_lls_delta.c
```

Additional build information

Include directories:

```
/include /include/acados_c /include/blasfeo/include /include/hpipm/include /lib
```

Additional build information

Libraries:

```
libblasfeo.so libacados.so libhpipm.so
```

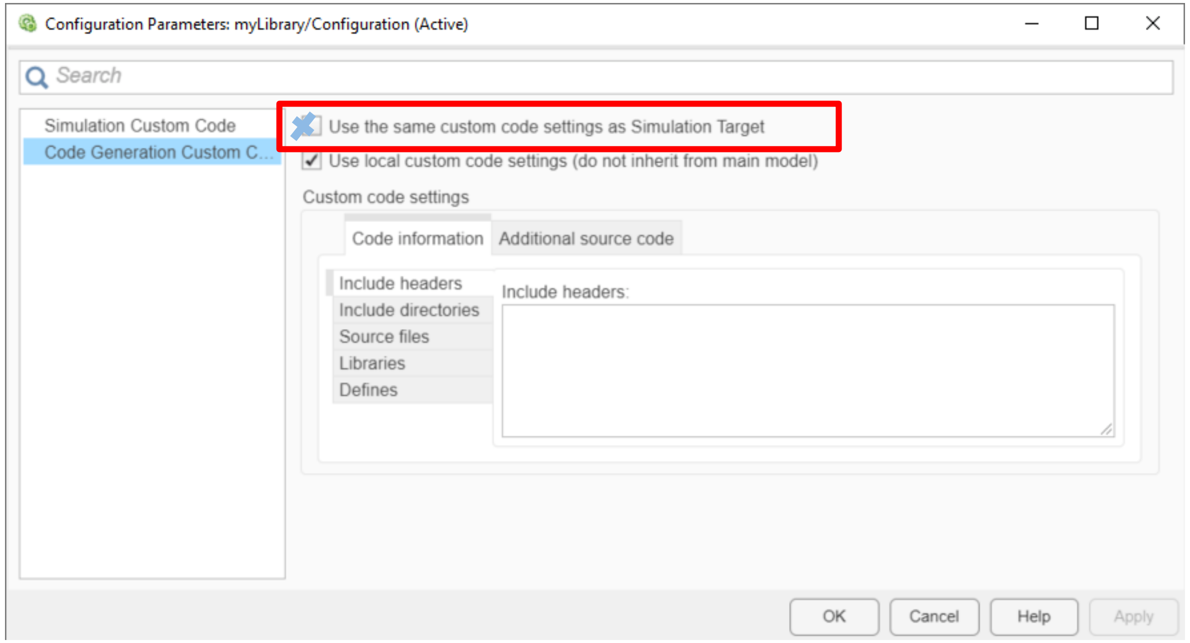
Sources:
nmpc_lls_delta_model/nmpc_lls_delta_dyn_disc_phi_fun.c
nmpc_lls_delta_model/nmpc_lls_delta_dyn_disc_phi_fun_jac.c
acados_solver_sfunction_nmpc_lls_delta.c
acados_solver_nmpc_lls_delta.c
-> get this information from the autogenerated "make_sfun.m" file from acados

Directories (relative to matlab working directory):
/include/acados_c (acados)
/include/blasfeo/include /include/hpipm/include /lib

Libs:
libblasfeo.so libacados.so libhpipm.so
-> the libs compiled on the RaspberryPi, but you have to transfer them to the respective lib folder you are pointing to on the windows host machine

Add Custom Code for acados binaries

Be sure to be in the c_generated_code folder on the matlab host-pc with matlab root



Configuration Parameters: myLibrary/Configuration (Active)

☒ Use the same custom code settings as Simulation Target

☑ Use local custom code settings (do not inherit from main model)

Custom code settings

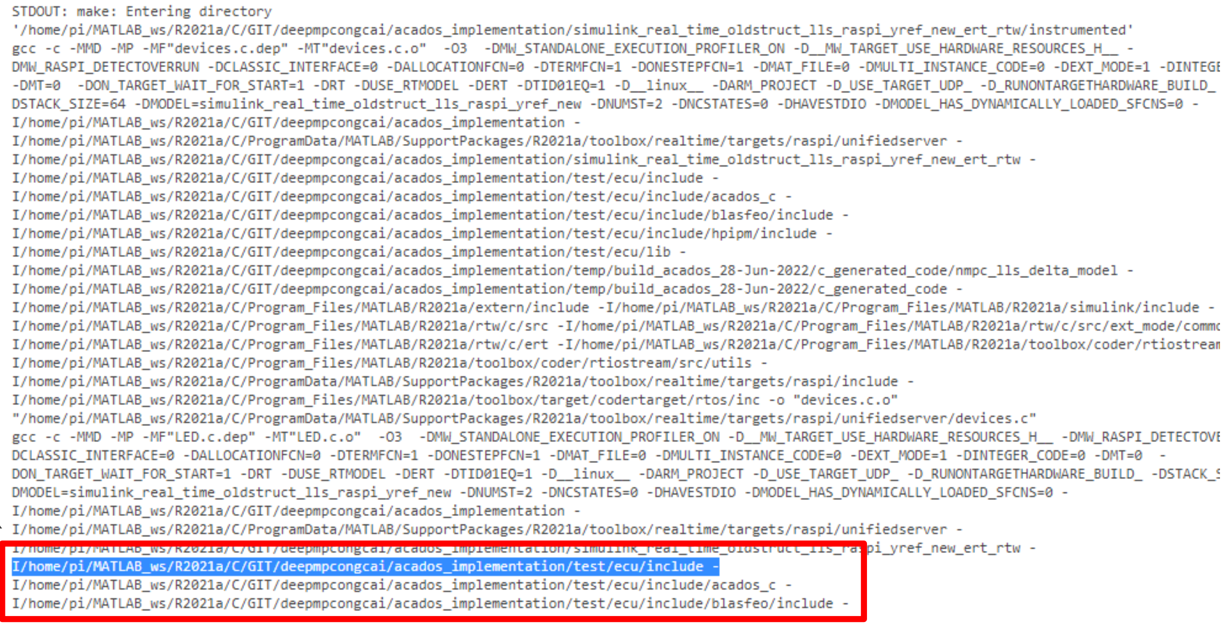
Include headers:

Attention:
Check the Box “use same custom code as in simulation target!”

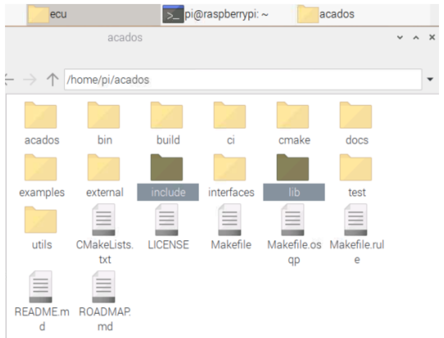
Be sure to be in the c_generated_code folder on the matlab host-pc with matlab root

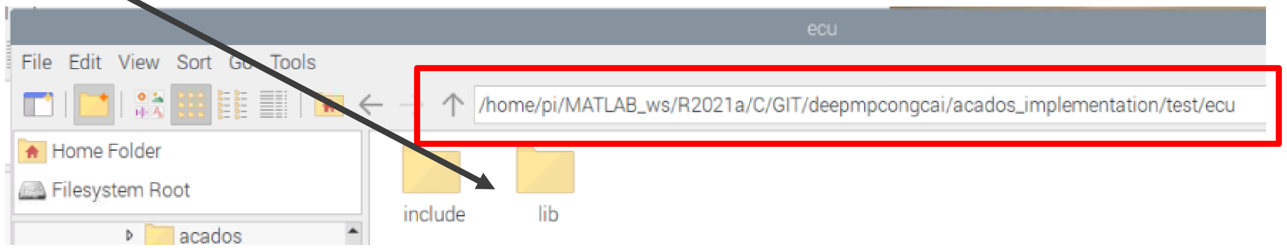
Build and Run the Model on Pi 1

- Press Monitor / Build
- See the diagnostic viewer for the path on the Pi, where the include folder and libraries should be copied.
- In this case the **matlab root folder** is in home/pi/matlab_ws/R2021a. The desktop PC's structure is copied into there.
- Add **include** folder and **lib** folder from acados into the respective folder, starting from the matlab working directory. Libs are being pulled tp the Pi by matlab.



```
STDOUT: make: Entering directory
'/home/pi/MATLAB_ws/R2021a/C/GIT/deepmpcongcai/acados_implementation/simulink_real_time_oldstruct_lls_raspi_yref_new_ert_rtw/instrumented'
gcc -c -MMD -MP -MF"devices.c.dep" -MT"devices.c.o"  -O3  -DMW_STANDALONE_EXECUTION_PROFILER_ON -D__MW_TARGET_USE_HARDWARE_RESOURCES_H__ -
DMW_RASPI_DETECTOVERRUN -DCLASSIC_INTERFACE=0 -DALLOCATIONFCN=0 -DTERMFCN=1 -DONESTEPFCN=1 -DMAT_FILE=0 -DMULTI_INSTANCE_CODE=0 -DEXT_MODE=1 -DINTEGE
-DMT=0  -DON_TARGET_WAIT_FOR_START=1 -DRT -DUSE_RTMODEL -DERT -DTID01EQ=1 -D__linux__ -DARM_PROJECT -D_USE_TARGET_UDP_ -D_RUNONTARGETHARDWARE_BUILD_
DSTACK_SIZE=64 -DMODEL=simulink_real_time_oldstruct_lls_raspi_yref_new -DNUMST=2 -DNCSTATES=0 -DHAVESTDIO -DMODEL_HAS_DYNAMICALLY_LOADED_SFCNS=0 -
I/home/pi/MATLAB_ws/R2021a/C/GIT/deepmpcongcai/acados_implementation -
I/home/pi/MATLAB_ws/R2021a/C/ProgramData/MATLAB/SupportPackages/R2021a/toolbox/realtime/targets/raspi/unifiedserver -
I/home/pi/MATLAB_ws/R2021a/C/GIT/deepmpcongcai/acados_implementation/simulink_real_time_oldstruct_lls_raspi_yref_new_ert_rtw -
I/home/pi/MATLAB_ws/R2021a/C/GIT/deepmpcongcai/acados_implementation/test/ecu/include -
I/home/pi/MATLAB_ws/R2021a/C/GIT/deepmpcongcai/acados_implementation/test/ecu/include/acados_c -
I/home/pi/MATLAB_ws/R2021a/C/GIT/deepmpcongcai/acados_implementation/test/ecu/include/blasfeo/include -
I/home/pi/MATLAB_ws/R2021a/C/GIT/deepmpcongcai/acados_implementation/test/ecu/include/hpipm/include -
I/home/pi/MATLAB_ws/R2021a/C/GIT/deepmpcongcai/acados_implementation/test/ecu/lib -
I/home/pi/MATLAB_ws/R2021a/C/GIT/deepmpcongcai/acados_implementation/temp/build_acados_28-Jun-2022/c_generated_code/nmpc_lls_delta_model -
I/home/pi/MATLAB_ws/R2021a/C/GIT/deepmpcongcai/acados_implementation/temp/build_acados_28-Jun-2022/c_generated_code -
I/home/pi/MATLAB_ws/R2021a/C/Program_Files/MATLAB/R2021a/extern/include -I/home/pi/MATLAB_ws/R2021a/C/Program_Files/MATLAB/R2021a/simulink/include -
I/home/pi/MATLAB_ws/R2021a/C/Program_Files/MATLAB/R2021a/rtw/c/src -I/home/pi/MATLAB_ws/R2021a/C/Program_Files/MATLAB/R2021a/rtw/c/src/ext_mode/commo
I/home/pi/MATLAB_ws/R2021a/C/Program_Files/MATLAB/R2021a/rtw/c/ert -I/home/pi/MATLAB_ws/R2021a/C/Program_Files/MATLAB/R2021a/toolbox/coder/rtiostream
I/home/pi/MATLAB_ws/R2021a/C/Program_Files/MATLAB/R2021a/toolbox/coder/rtiostream/src/utils -
I/home/pi/MATLAB_ws/R2021a/C/ProgramData/MATLAB/SupportPackages/R2021a/toolbox/realtime/targets/raspi/include -
I/home/pi/MATLAB_ws/R2021a/C/Program_Files/MATLAB/R2021a/toolbox/target/codertarget/rtos/inc -o "devices.c.o"
"/home/pi/MATLAB_ws/R2021a/C/ProgramData/MATLAB/SupportPackages/R2021a/toolbox/realtime/targets/raspi/unifiedserver/devices.c"
gcc -c -MMD -MP -MF"LED.c.dep" -MT"LED.c.o"  -O3  -DMW_STANDALONE_EXECUTION_PROFILER_ON -D__MW_TARGET_USE_HARDWARE_RESOURCES_H__ -DMW_RASPI_DETECTOVE
DCLASSIC_INTERFACE=0 -DALLOCATIONFCN=0 -DTERMFCN=1 -DONESTEPFCN=1 -DMAT_FILE=0 -DMULTI_INSTANCE_CODE=0 -DEXT_MODE=1 -DINTEGER_CODE=0 -DMT=0  -
DON_TARGET_WAIT_FOR_START=1 -DRT -DUSE_RTMODEL -DERT -DTID01EQ=1 -D__linux__ -DARM_PROJECT -D_USE_TARGET_UDP_ -D_RUNONTARGETHARDWARE_BUILD_ -DSTACK_S
DMODEL=simulink_real_time_oldstruct_lls_raspi_yref_new -DNUMST=2 -DNCSTATES=0 -DHAVESTDIO -DMODEL_HAS_DYNAMICALLY_LOADED_SFCNS=0 -
I/home/pi/MATLAB_ws/R2021a/C/GIT/deepmpcongcai/acados_implementation -
I/home/pi/MATLAB_ws/R2021a/C/ProgramData/MATLAB/SupportPackages/R2021a/toolbox/realtime/targets/raspi/unifiedserver -
I/home/pi/MATLAB_ws/R2021a/C/GIT/deepmpcongcai/acados_implementation/simulink_real_time_oldstruct_lls_raspi_yref_new_ert_rtw -
I/home/pi/MATLAB_ws/R2021a/C/GIT/deepmpcongcai/acados_implementation/test/ecu/include -
I/home/pi/MATLAB_ws/R2021a/C/GIT/deepmpcongcai/acados_implementation/test/ecu/include/acados_c -
I/home/pi/MATLAB_ws/R2021a/C/GIT/deepmpcongcai/acados_implementation/test/ecu/include/blasfeo/include -
```

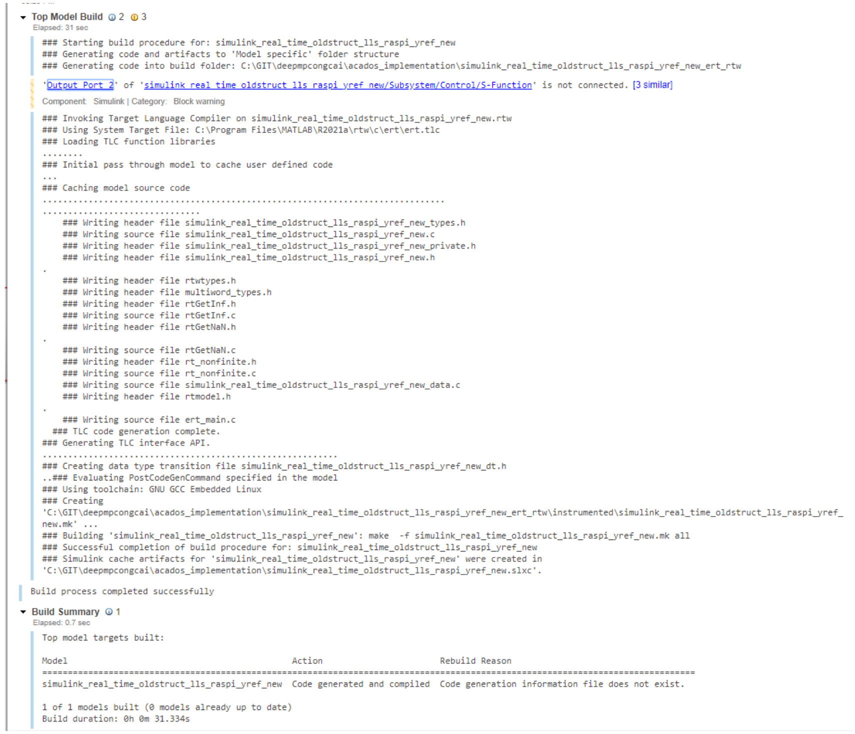




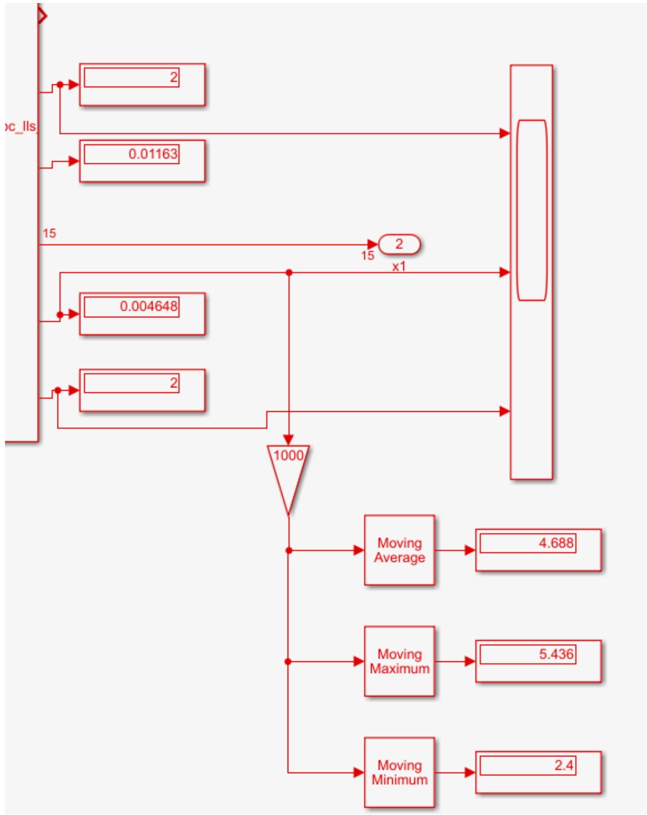
/home/pi/MATLAB_ws/R2021a/C/GIT/deepmpcongcai/acados_implementation/test/ecu

Build and Run the Model on Pi 2

- Press Monitor / Build:



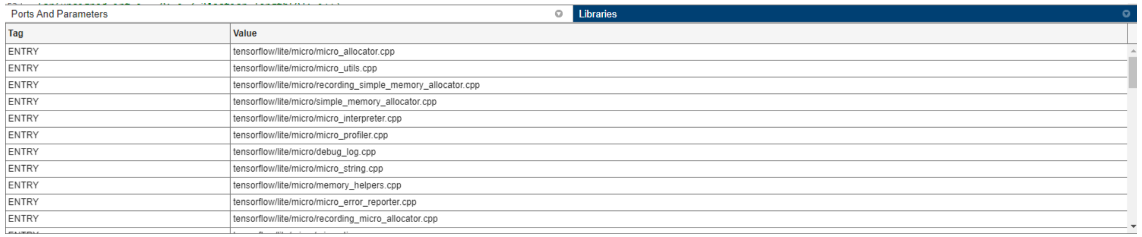
```
▾ Top Model Build ⓘ 2 ⓘ 3
  Elapsed: 31 sec
   ### Starting build procedure for: simulink_real_time_oldstruct_lls_raspi_yref_new
   ### Generating code and artifacts to 'Model specific' folder structure
   ### Generating code into build folder: C:\GIT\deepmpcongcai\acados_implementation\simulink_real_time_oldstruct_lls_raspi_yref_new_ert_rtw

   'Output Port 2' of 'simulink_real_time_oldstruct_lls_raspi_yref_new/Subsystem/Control/S-Function' is not connected. [3 similar]
   Component:  Simulink | Category:  Block warning

   ### Invoking Target Language Compiler on simulink_real_time_oldstruct_lls_raspi_yref_new.rtw
   ### Using System Target File: C:\Program Files\MATLAB\R2021a\rtw\c\ert\ert.tlc
   ### Loading TLC function libraries
   ........
   ### Initial pass through model to cache user defined code
   ...
   ### Caching model source code
   ...............................................................................
   ...............................
       ### Writing header file simulink_real_time_oldstruct_lls_raspi_yref_new_types.h
       ### Writing source file simulink_real_time_oldstruct_lls_raspi_yref_new.c
       ### Writing header file simulink_real_time_oldstruct_lls_raspi_yref_new_private.h
       ### Writing header file simulink_real_time_oldstruct_lls_raspi_yref_new.h
   .
       ### Writing header file rtwtypes.h
       ### Writing header file multiword_types.h
       ### Writing header file rtGetInf.h
       ### Writing source file rtGetInf.c
       ### Writing header file rtGetNaN.h
   .
       ### Writing source file rtGetNaN.c
       ### Writing header file rt_nonfinite.h
       ### Writing source file rt_nonfinite.c
       ### Writing source file simulink_real_time_oldstruct_lls_raspi_yref_new_data.c
       ### Writing header file rtmodel.h
   .
       ### Writing source file ert_main.c
     ### TLC code generation complete.
   ### Generating TLC interface API.
   ..........................................................
   ### Creating data type transition file simulink_real_time_oldstruct_lls_raspi_yref_new_dt.h
   ..### Evaluating PostCodeGenCommand specified in the model
   ### Using toolchain: GNU GCC Embedded Linux
   ### Creating
   'C:\GIT\deepmpcongcai\acados_implementation\simulink_real_time_oldstruct_lls_raspi_yref_new_ert_rtw\instrumented\simulink_real_time_oldstruct_lls_raspi_yref_
   new.mk' ...
   ### Building 'simulink_real_time_oldstruct_lls_raspi_yref_new': make  -f simulink_real_time_oldstruct_lls_raspi_yref_new.mk all
   ### Successful completion of build procedure for: simulink_real_time_oldstruct_lls_raspi_yref_new
   ### Simulink cache artifacts for 'simulink_real_time_oldstruct_lls_raspi_yref_new' were created in
   'C:\GIT\deepmpcongcai\acados_implementation\simulink_real_time_oldstruct_lls_raspi_yref_new.slxc'.

 Build process completed successfully

▾ Build Summary ⓘ 1
  Elapsed: 0.7 sec
   Top model targets built:

   Model                                             Action                         Rebuild Reason
   =================================================================================================================================
   simulink_real_time_oldstruct_lls_raspi_yref_new  Code generated and compiled  Code generation information file does not exist.

   1 of 1 models built (0 models already up to date)
   Build duration: 0h 0m 31.334s
```



- Watch your model running (while on Desktop PC)!

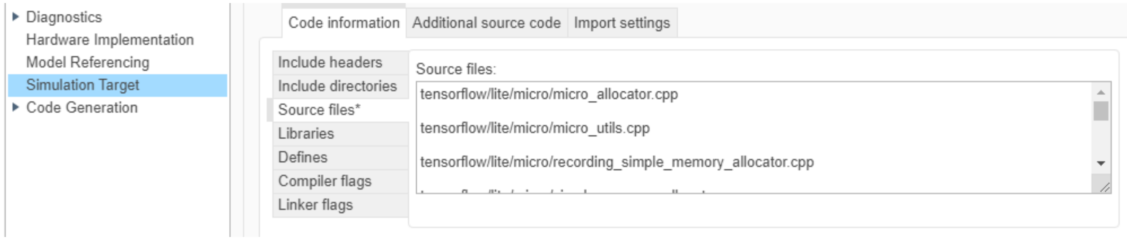
Run Simulink Model with TensorFlow on Pi

- Build S-Function locally to test the model offline.
- Add all the source files, which are in the S-Function builder also to the Simulink model configuration (Ctrl.+E)



Ports And Parameters | Libraries

Tag	Value
ENTRY	tensorflow/lite/micro/micro_allocator.cpp
ENTRY	tensorflow/lite/micro/micro_utils.cpp
ENTRY	tensorflow/lite/micro/recording_simple_memory_allocator.cpp
ENTRY	tensorflow/lite/micro/simple_memory_allocator.cpp
ENTRY	tensorflow/lite/micro/micro_interpreter.cpp
ENTRY	tensorflow/lite/micro/micro_profiler.cpp
ENTRY	tensorflow/lite/micro/debug_log.cpp
ENTRY	tensorflow/lite/micro/micro_string.cpp
ENTRY	tensorflow/lite/micro/memory_helpers.cpp
ENTRY	tensorflow/lite/micro/micro_error_reporter.cpp
ENTRY	tensorflow/lite/micro/recording_micro_allocator.cpp



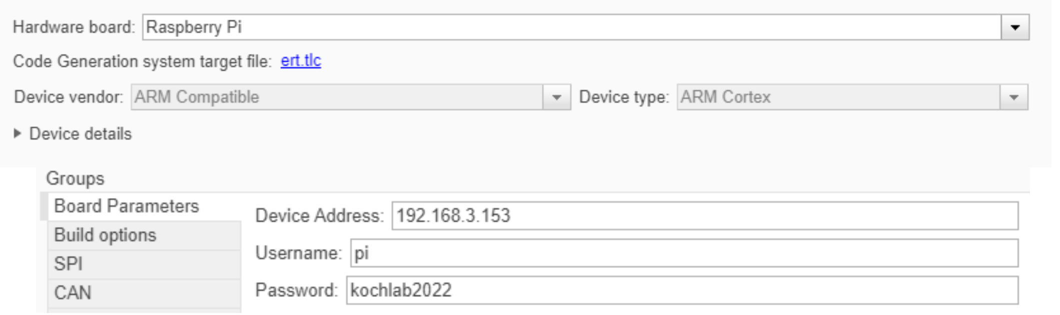
Source files:

tensorflow/lite/micro/micro_allocator.cpp

tensorflow/lite/micro/micro_utils.cpp

tensorflow/lite/micro/recording_simple_memory_allocator.cpp

- Set hardware board to Raspberry Pi in the Simulink model configuration (Ctrl.+E). Set the board parameters



Hardware board: Raspberry Pi

Code Generation system target file: ert.tlc

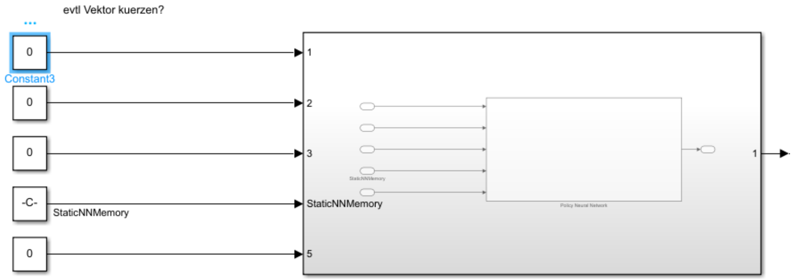
Device vendor: ARM Compatible Device type: ARM Cortex

Device Address: 192.168.3.153

Username: pi

Password: kochlab2022

- Copy the source files to the Pi: “MyIncludeDirs” (without the parent folder) and “tensorflow” to the model folder on the Pi, as explained in Chapter “Build and Run the Model on Pi 1”
- Do not use “from workspace” blocks. They will crash the communication to the Pi due to the huge amount of data / alignment issues. Replace them with constants:



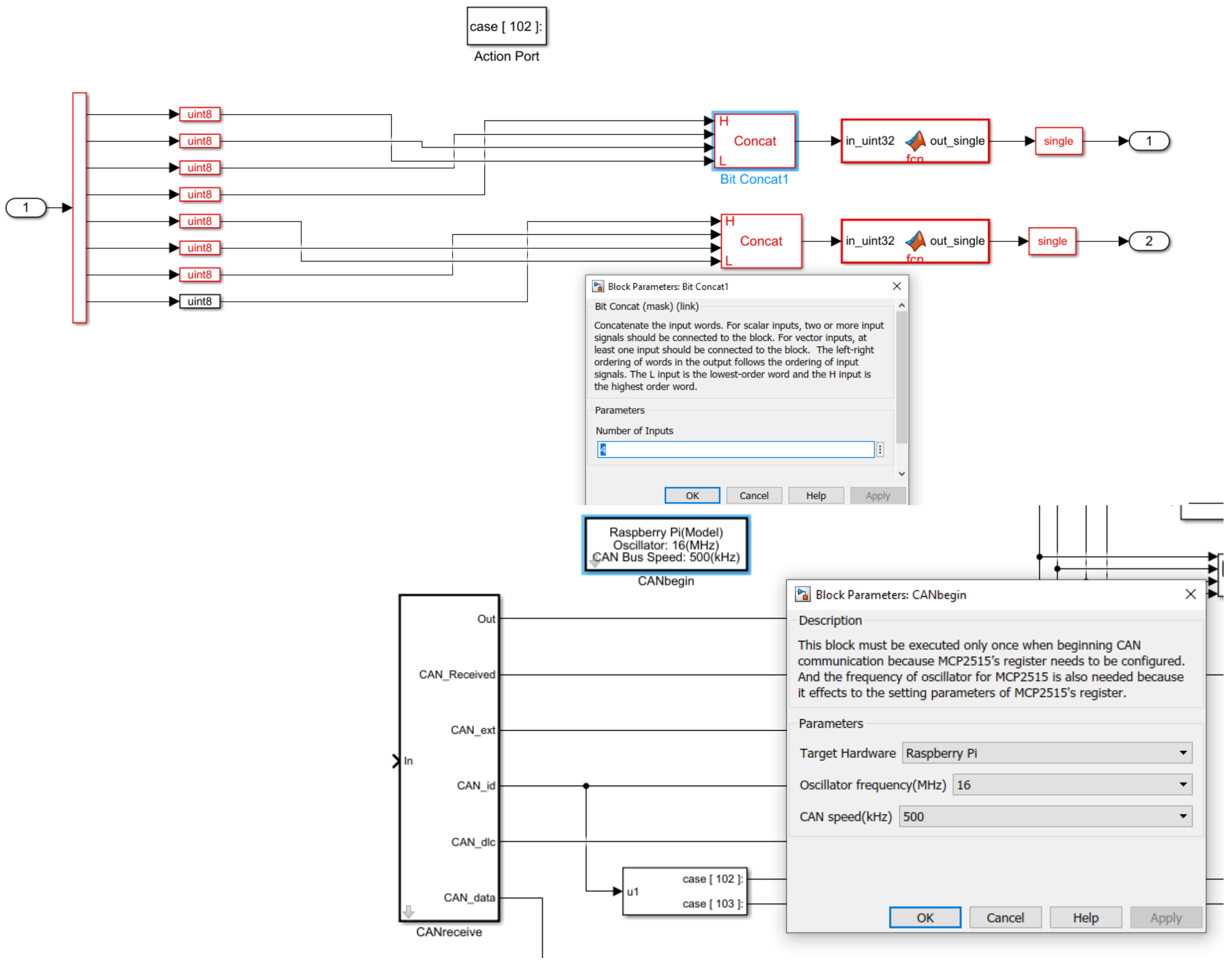
Setup CAN communication on Pi

CAN RECEIVE

- CAN receive block: download a different packacge (see above) and extract the receive block from the example.
  - Bit concat of the demuxed unit8 / CAN frame bytes
  - Byte order swap (little endian / big endian)
  - Convert data to unit 32 with matlab function: “function out_single = fcn(in_uint32) out_single = typecast(in_uint32, 'single’);”

- Put all these in action subsystems and call depending on the message ID (see pic) – when using multiple messages

- As usual: Terminate CAN network with 2x 120 Ohm resistors between High and Low



Block Parameters: Bit Concat1

Bit Concat (mask) (link)

Concatenate the input words. For scalar inputs, two or more input signals should be connected to the block. For vector inputs, at least one input should be connected to the block. The left-right ordering of words in the output follows the ordering of input signals. The L input is the lowest-order word and the H input is the highest order word.

Parameters

Number of Inputs: 4

Block Parameters: CANbegin

Description

This block must be executed only once when beginning CAN communication because MCP2515’s register needs to be configured. And the frequency of oscillator for MCP2515 is also needed because it effects to the setting parameters of MCP2515’s register.

Parameters

Target Hardware: Raspberry Pi

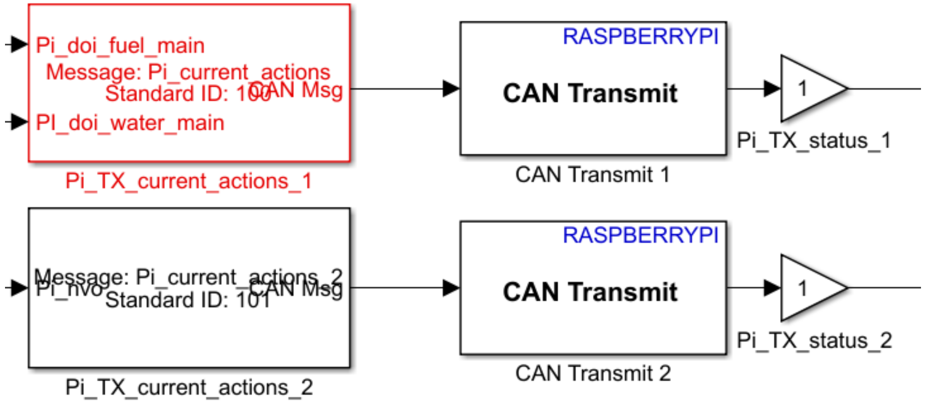
Oscillator frequency(MHz): 16

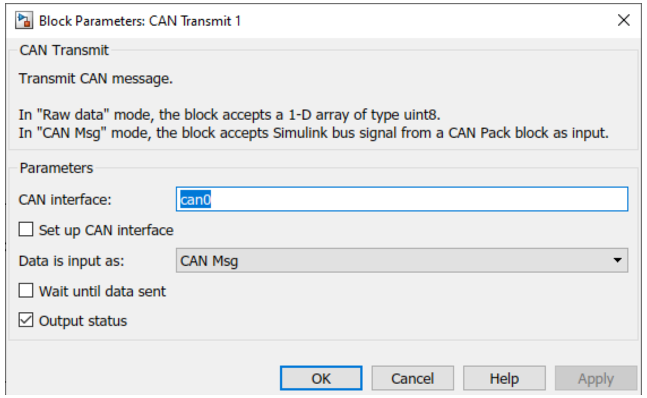
CAN speed(kHz): 500

Setup CAN communication on Pi

CAN TRANSMIT

- CAN transmit block from official mathworks support package works well
- Use normal CAN TX block from mathworks to pack the data (roght bottom pic). Set ID, data types, length, etc.
- Transmit 1: Can0 interface is channel 1 on the 2 channel shield, Can1 is channel 2.
- As usual: Terminate CAN network with 2x 120 Ohm resistors between High and Low





Block Parameters: CAN Transmit 1

CAN Transmit

Transmit CAN message.

In "Raw data" mode, the block accepts a 1-D array of type uint8.
In "CAN Msg" mode, the block accepts Simulink bus signal from a CAN Pack block as input.

Parameters

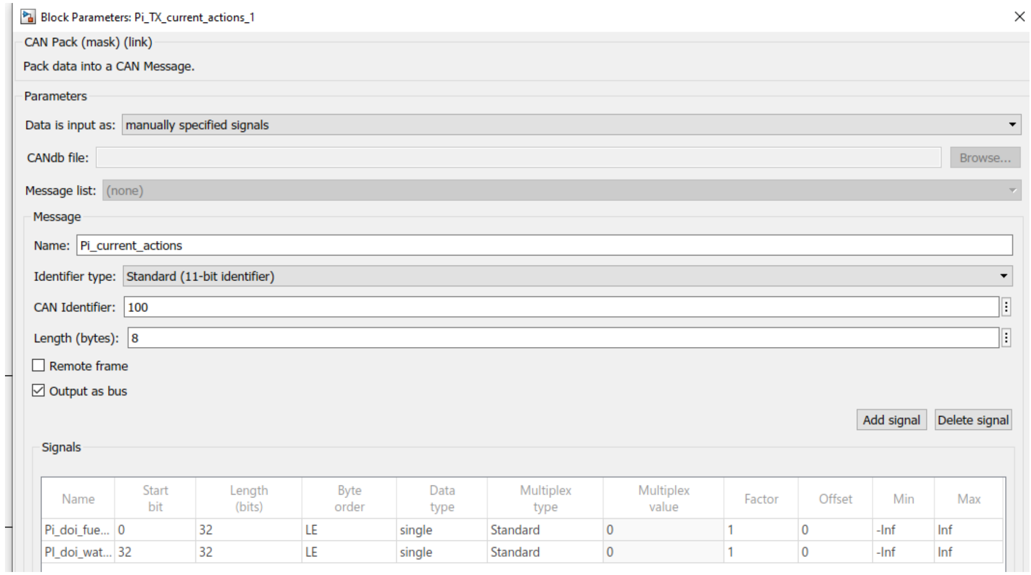
CAN interface: can0

☐ Set up CAN interface

Data is input as: CAN Msg

☐ Wait until data sent

☑ Output status



Block Parameters: Pi_TX_current_actions_1

CAN Pack (mask) (link)

Pack data into a CAN Message.

Parameters

Data is input as: manually specified signals

CANdb file:

Message list: (none)

Message

Name: Pi_current_actions

Identifier type: Standard (11-bit identifier)

CAN Identifier: 100

Length (bytes): 8

☐ Remote frame

☑ Output as bus

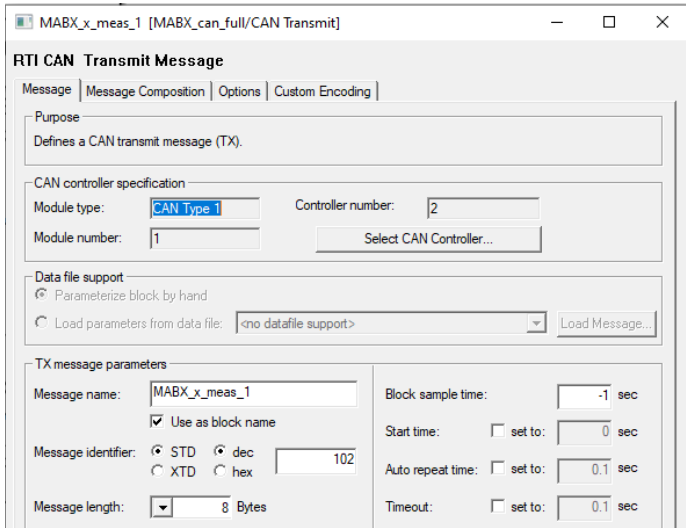
Signals

Name	Start bit	Length (bits)	Byte order	Data type	Multiplex type	Multiplex value	Factor	Offset	Min	Max
Pi_doi_fue...	0	32	LE	single	Standard	0	1	0	-Inf	Inf
PI_doi_wat...	32	32	LE	single	Standard	0	1	0	-Inf	Inf

Setup CAN communication on MABX

CAN TRANSMIT

- Use dSPACE Blocks
- Pack message manually
- Be careful with module and controller number



MABX_x_meas_1 [MABX_can_full/CAN Transmit]

RTI CAN Transmit Message

Purpose

Defines a CAN transmit message (TX).

CAN controller specification

Module type: CAN Type 1

Controller number: 2

Module number: 1

Data file support

● Parameterize block by hand

○ Load parameters from data file: <no datafile support>

TX message parameters

Message name: MABX_x_meas_1

☑ Use as block name

Message identifier: ● STD ○ XTD ● dec ○ hex 102

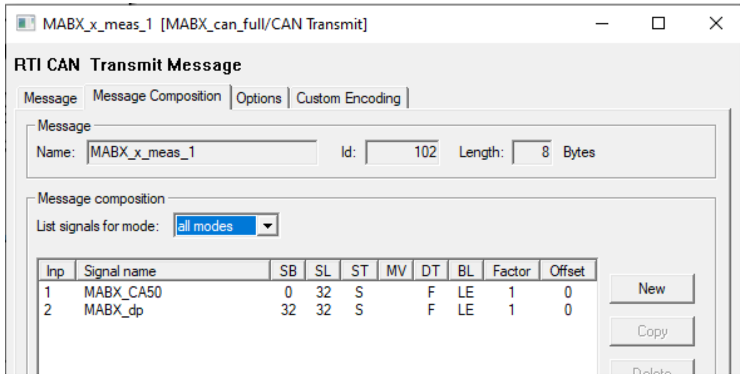
Message length: 8 Bytes

Block sample time: -1 sec

Start time: □ set to: 0 sec

Auto repeat time: □ set to: 0.1 sec

Timeout: □ set to: 0.1 sec



MABX_x_meas_1 [MABX_can_full/CAN Transmit]

RTI CAN Transmit Message

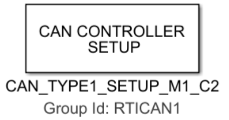
Message

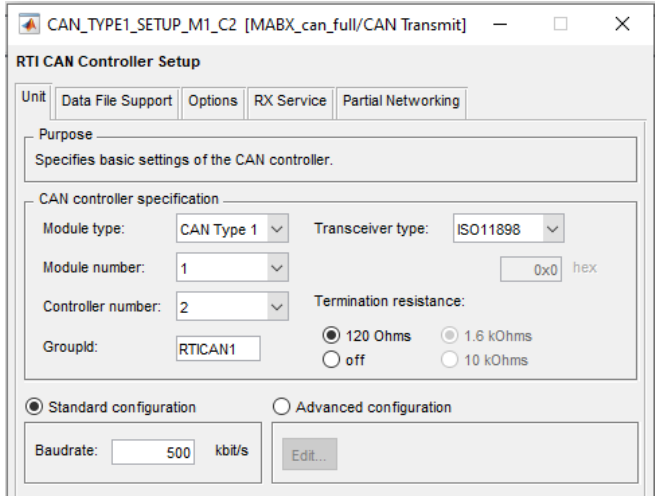
Name: MABX_x_meas_1 Id: 102 Length: 8 Bytes

Message composition

List signals for mode: all modes

Inp	Signal name	SB	SL	ST	MV	DT	BL	Factor	Offset
1	MABX_CA50	0	32	S		F	LE	1	0
2	MABX_dp	32	32	S		F	LE	1	0





CAN_TYPE1_SETUP_M1_C2 [MABX_can_full/CAN Transmit]

RTI CAN Controller Setup

Unit | Data File Support | Options | RX Service | Partial Networking

Purpose

Specifies basic settings of the CAN controller.

CAN controller specification

Module type: CAN Type 1

Transceiver type: ISO11898

Module number: 1

0x0 hex

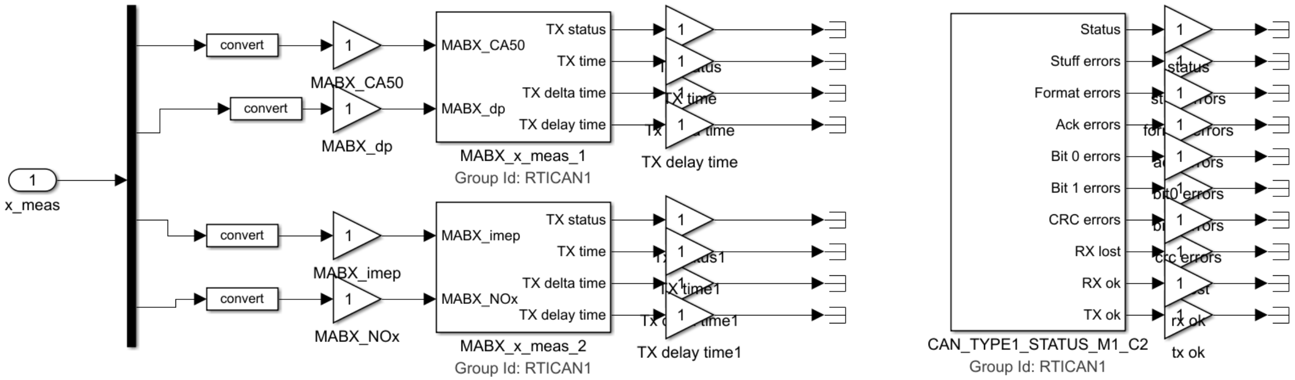
Controller number: 2

Termination resistance: ● 120 Ohms ○ off ○ 1.6 kOhms ○ 10 kOhms

GroupId: RTICAN1

● Standard configuration ○ Advanced configuration

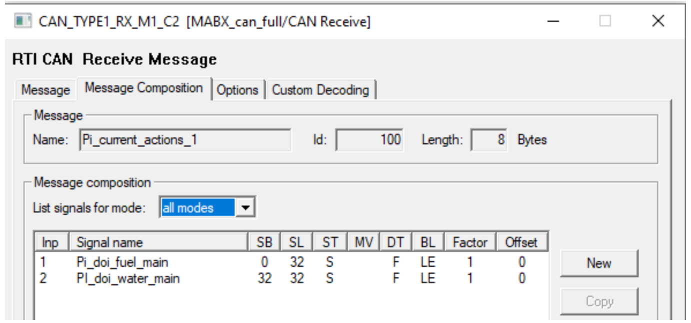
Baudrate: 500 kbit/s



Setup CAN communication on MABX

CAN RECEIVE

- Use dSPACE Blocks
- unpack message manually
- Use multiple blocks for multiple messages



CAN_TYPE1_RX_M1_C2 [MABX_can_full/CAN Receive]

RTI CAN Receive Message

Message

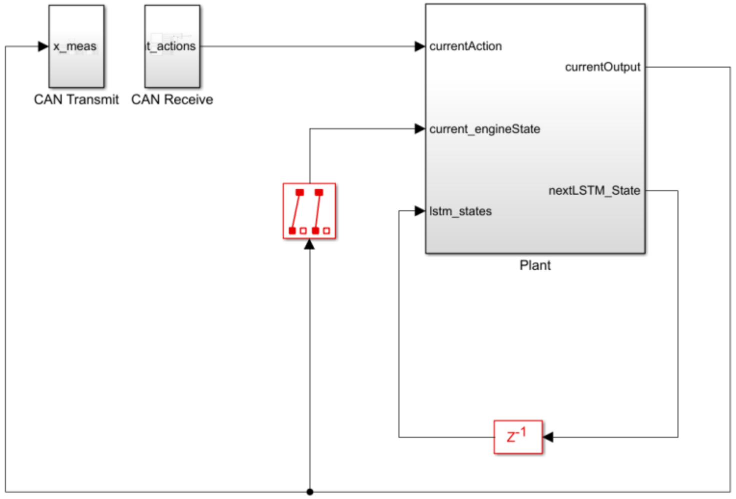
Name: Pi_current_actions_1 Id: 100 Length: 8 Bytes

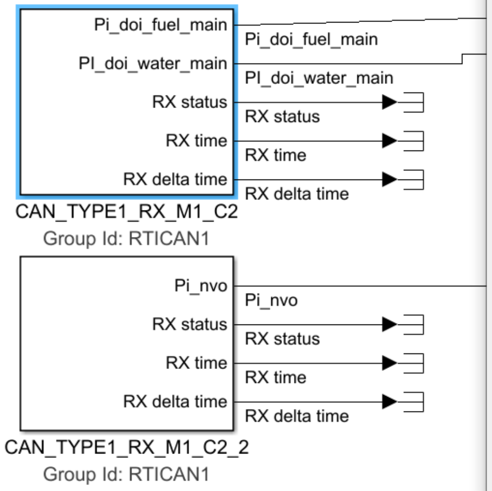
Message composition

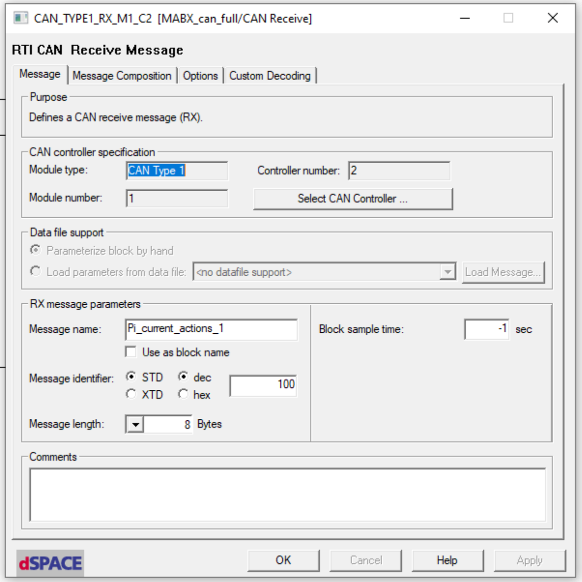
List signals for mode: all modes

Inp	Signal name	SB	SL	ST	MV	DT	BL	Factor	Offset
1	Pi_doi_fuel_main	0	32	S		F	LE	1	0
2	PI_doi_water_main	32	32	S		F	LE	1	0

- Possible model structure with Receive and Send:







CAN_TYPE1_RX_M1_C2 [MABX_can_full/CAN Receive]

RTI CAN Receive Message

Purpose

Defines a CAN receive message (RX).

CAN controller specification

Module type: CAN Type 1 Controller number: 2

Module number: 1

Data file support

● Parameterize block by hand

○ Load parameters from data file: <no datafile support>

RX message parameters

Message name: Pi_current_actions_1

□ Use as block name

Message identifier: ● STD ○ XTD ● dec ○ hex 100

Message length: 8 Bytes

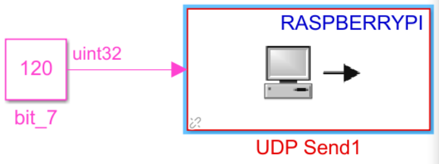
Block sample time: -1 sec

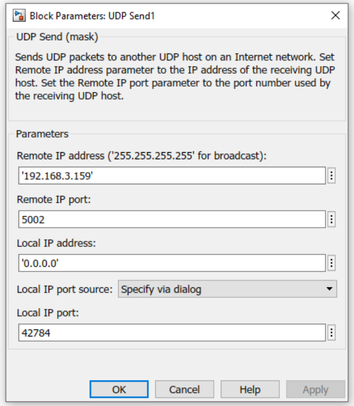
Comments

Setup UDP communication on Pi

UDP SEND

- Manipulate the standard UDP Send block by opening the mask editor (see below)
- Set the visibility of “Local IP port source” and “Local IP port” to true
- Now you can edit the mask in Simulink.
- Choose random port. Due to a bug this port is not both local and remote port.
- Give over data, look at data type and message size. Change settings on the receiving side respectively.
- UDP Send Block on Pi works same way. Set data type, message size and sample time accordingly.
- Check used port in terminal on Pi if necessary:
  $ sudo ss -ulpn




Block Parameters: UDP Send1

UDP Send (mask)

Sends UDP packets to another UDP host on an Internet network. Set Remote IP address parameter to the IP address of the receiving UDP host. Set the Remote IP port parameter to the port number used by the receiving UDP host.

Parameters

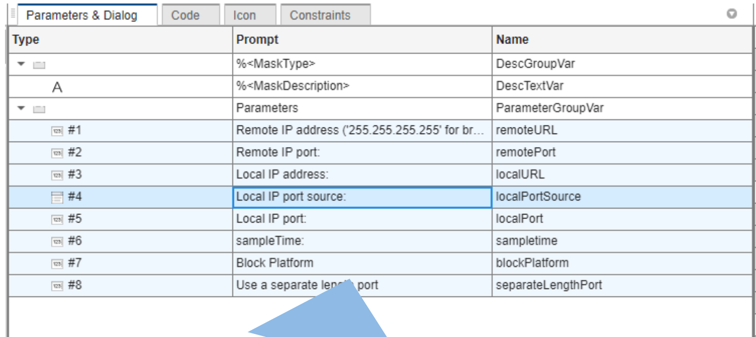
Remote IP address ('255.255.255.255' for broadcast): '192.168.3.159'

Remote IP port: 5002

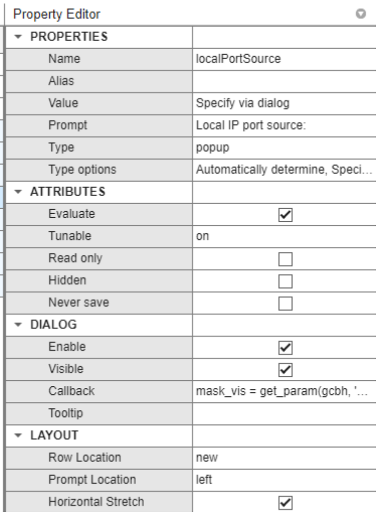
Local IP address: '0.0.0.0'

Local IP port source: Specify via dialog

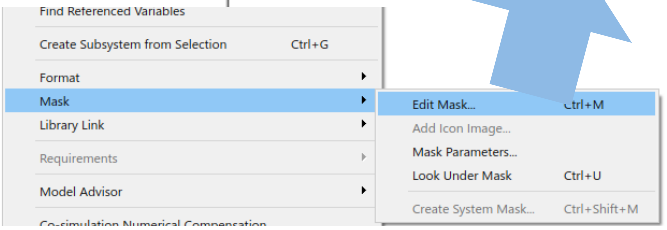
Local IP port: 42784



Type	Prompt	Name
	%<MaskType>	DescGroupVar
A	%<MaskDescription>	DescTextVar
	Parameters	ParameterGroupVar
#1	Remote IP address ('255.255.255.255' for br…	remoteURL
#2	Remote IP port:	remotePort
#3	Local IP address:	localURL
#4	Local IP port source:	localPortSource
#5	Local IP port:	localPort
#6	sampleTime:	sampletime
#7	Block Platform	blockPlatform
#8	Use a separate len[illegible] port	separateLengthPort


Property Editor

PROPERTIES	
Name	localPortSource
Alias	
Value	Specify via dialog
Prompt	Local IP port source:
Type	popup
Type options	Automatically determine, Speci…
ATTRIBUTES	
Evaluate	☑
Tunable	on
Read only	☐
Hidden	☐
Never save	☐
DIALOG	
Enable	☑
Visible	☑
Callback	mask_vis = get_param(gcbh, '…
Tooltip	
LAYOUT	
Row Location	new
Prompt Location	left
Horizontal Stretch	☑

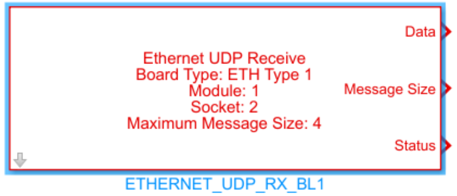


Find Referenced Variables
Create Subsystem from Selection Ctrl+G
Format
Mask
Library Link
Requirements
Model Advisor
Co-simulation Numerical Compensation
Edit Mask… Ctrl+M
Add Icon Image…
Mask Parameters…
Look Under Mask Ctrl+U
Create System Mask… Ctrl+Shift+M

Setup UDP communication on MABX

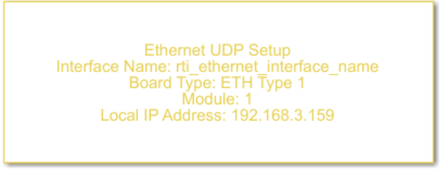
UDP RECEIVE

- If you use Ethernet Port (NOT host PC), set board type to ETH type 1
- Create new subnetwork (not the host PC connection): 192.168.3.XXX
- Assign an IP to the MABX (192.168.3.159)
- Set remote and local ports (have to be the same due to broken mathworks block) and remote IP (other device, e.g. Pi, here 192.168.3.153, 42784, 42784
- Define socket type in setup and choose the correct one in RX block.
- Define message size (in bytes. One variable of Uint32 = 4 bytes)
- UDP Send Block on MABX works same way. Set data type, message size and sample time accordingly.

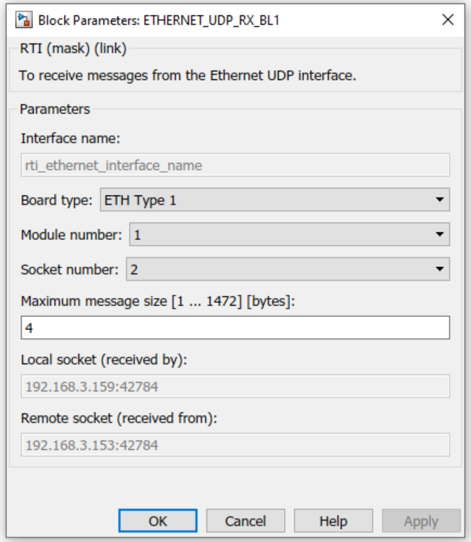


Ethernet UDP Receive
Board Type: ETH Type 1
Module: 1
Socket: 2
Maximum Message Size: 4

ETHERNET_UDP_RX_BL1



Ethernet UDP Setup
Interface Name: rti_ethernet_interface_name
Board Type: ETH Type 1
Module: 1
Local IP Address: 192.168.3.159



Block Parameters: ETHERNET_UDP_RX_BL1

RTI (mask) (link)

To receive messages from the Ethernet UDP interface.

Parameters

Interface name: rti_ethernet_interface_name

Board type: ETH Type 1

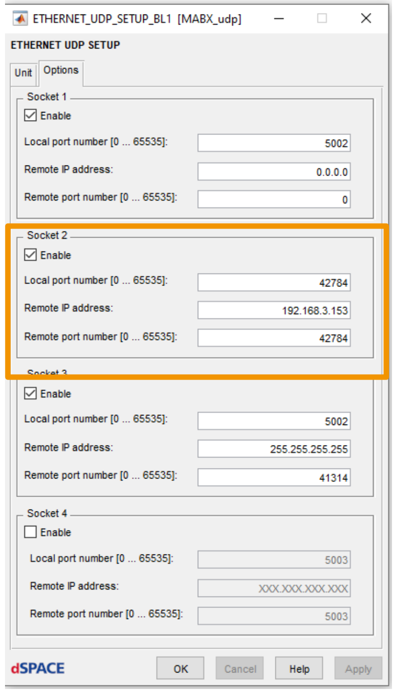
Module number: 1

Socket number: 2

Maximum message size [1 ... 1472] [bytes]: 4

Local socket (received by): 192.168.3.159:42784

Remote socket (received from): 192.168.3.153:42784



ETHERNET_UDP_SETUP_BL1 [MABX_udp]

ETHERNET UDP SETUP

Socket 1
☑ Enable
Local port number [0 ... 65535]: 5002
Remote IP address: 0.0.0.0
Remote port number [0 ... 65535]: 0

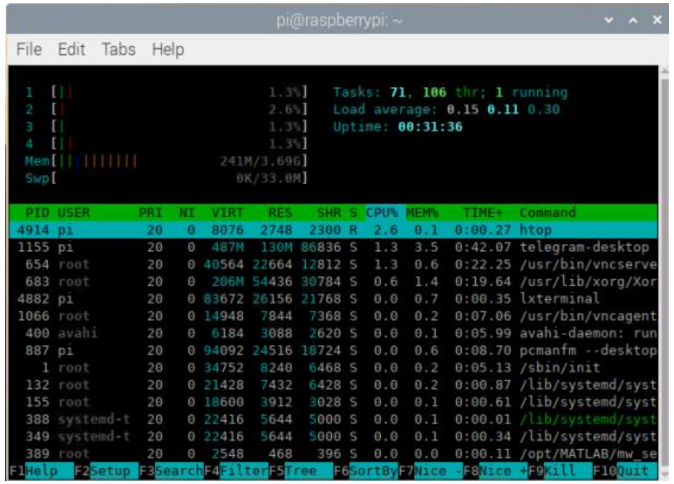
Socket 2
☑ Enable
Local port number [0 ... 65535]: 42784
Remote IP address: 192.168.3.153
Remote port number [0 ... 65535]: 42784

Socket 3
☑ Enable
Local port number [0 ... 65535]: 5002
Remote IP address: 255.255.255.255
Remote port number [0 ... 65535]: 41314

Socket 4
☐ Enable
Local port number [0 ... 65535]: 5003
Remote IP address: XXX.XXX.XXX.XXX
Remote port number [0 ... 65535]: 5003

Linux / Pi basics (feel free to add)

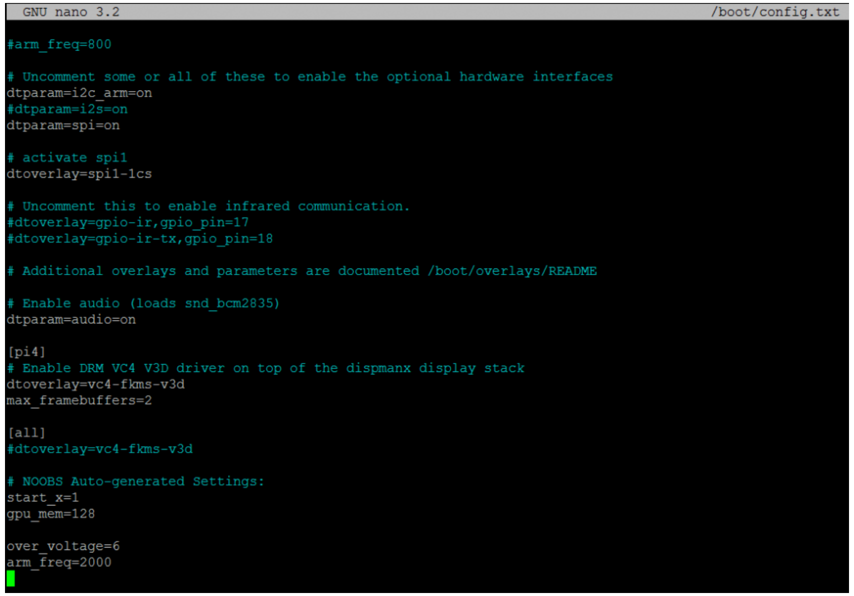
- Change passwort with passwd
- Use prefix sudo to run as root
- Change to root user with all rights with sudo su
- Update all packages with sudo apt-get upgrade
- Reboot with sudo reboot
- Move file to directory cp SOURCE1 SOURCE2 SOURCE3 SOURCEn DIRECTORY
- Open Task Manager with htop (see right)
- Open terminal in specific folder with "F4" in file manager



```
  1  [||                     1.3%]   Tasks: 71, 106 thr; 1 running
  2  [|                      2.6%]   Load average: 0.15 0.11 0.30
  3  [|                      1.3%]   Uptime: 00:31:36
  4  [||                     1.3%]
  Mem[|||||||||||      241M/3.69G]
  Swp[                   0K/33.0M]

  PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
 4914 pi         20   0  8076  2748  2300 R  2.6  0.1  0:00.27 htop
 1155 pi         20   0  487M  130M 86836 S  1.3  3.5  0:42.07 telegram-desktop
  654 root       20   0 40564 22664 12812 S  1.3  0.6  0:22.25 /usr/bin/vncserve
  683 root       20   0  206M 54436 30784 S  0.6  1.4  0:19.64 /usr/lib/xorg/Xor
 4882 pi         20   0 83672 26156 21768 S  0.0  0.7  0:00.35 lxterminal
 1066 root       20   0 14948  7844  7368 S  0.0  0.2  0:07.06 /usr/bin/vncagent
  400 avahi      20   0  6184  3088  2620 S  0.0  0.1  0:05.99 avahi-daemon: run
  887 pi         20   0 94092 24516 18724 S  0.0  0.6  0:08.70 pcmanfm --desktop
    1 root       20   0 34752  8240  6468 S  0.0  0.2  0:05.13 /sbin/init
  132 root       20   0 21428  7432  6428 S  0.0  0.2  0:00.87 /lib/systemd/syst
  155 root       20   0 18600  3912  3028 S  0.0  0.1  0:00.61 /lib/systemd/syst
  388 systemd-t  20   0 22416  5644  5000 S  0.0  0.1  0:00.01 /lib/systemd/syst
  349 systemd-t  20   0 22416  5644  5000 S  0.0  0.1  0:00.34 /lib/systemd/syst
  389 root       20   0  2548   468   396 S  0.0  0.0  0:00.11 /opt/MATLAB/mw_se
F1Help  F2Setup F3SearchF4FilterF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Quit
```

Overclock RasPi 4B/400

- Monitor Frequency: https://low-orbit.net/raspberry-pi-how-to-check-cpu-speed
  cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_cur_freq
  cat /sys/devices/system/cpu/cpu0/cpufreq/cpuinfo_max_freq
- Change Frequency:
  sudo nano /boot/config.txt
  over_voltage=6
  arm_freq=2000
  force_turbo=1 (fixes the speed to the overclocked speed above)
  Ctrl+X plus Y and Enter to save and overwrite.
- Get optimal settings in the tool on the top here:
  https://buyzero.de/blogs/news/raspberry-pi-ubertakten-pi-4-pi-400-pi-3b



```
  GNU nano 3.2                                                        /boot/config.txt

#arm_freq=800

# Uncomment some or all of these to enable the optional hardware interfaces
dtparam=i2c_arm=on
#dtparam=i2s=on
dtparam=spi=on

# activate spi1
dtoverlay=spi1-1cs

# Uncomment this to enable infrared communication.
#dtoverlay=gpio-ir,gpio_pin=17
#dtoverlay=gpio-ir-tx,gpio_pin=18

# Additional overlays and parameters are documented /boot/overlays/README

# Enable audio (loads snd_bcm2835)
dtparam=audio=on

[pi4]
# Enable DRM VC4 V3D driver on top of the dispmanx display stack
dtoverlay=vc4-fkms-v3d
max_framebuffers=2

[all]
#dtoverlay=vc4-fkms-v3d

# NOOBS Auto-generated Settings:
start_x=1
gpu_mem=128

over_voltage=6
arm_freq=2000
```

Overclock RasPi 5

- Monitor Frequency: https://low-orbit.net/raspberry-pi-how-to-check-cpu-speed

  ```
  cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_cur_freq
  cat /sys/devices/system/cpu/cpu0/cpufreq/cpuinfo_max_freq
  ```

- Change Frequency:

  ```
  sudo nano /boot/firmware/config.txt
  #over_voltage_delta=50000 (only use when going over 2800 Mhz, not tested yet)
  arm_freq=2800 (might go up to 3000 with overvoltage delta enabled)
  force_turbo=1 (fixes the speed to the overclocked speed above)
  ```

  Ctrl+X plus Y and Enter to save and overwrite.

- Get optimal settings in the tool on the top here: https://www.jeffgeerling.com/blog/2023/overclocking-and-underclocking-raspberry-pi-5

Overclock RasPi ALL

- Monitor frequency in open terminal:
  ```
  watch -n 1 cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_cur_freq
  ```
- Watch temperature CPU in terminal:
  ```
  cpu=$(</sys/class/thermal/thermal_zone0/temp)
  echo "$((cpu/1000)) c"
  ```
- Watch temperature CPU in terminal (stop with Ctrl+C):
  ```
  watch -c -b -d -n 1 -- 'vcgencmd measure_temp'
  ```
-

Additional Information

Info Matlab/SL installation:

Packages: libsdl1.2-dev libsdl2-dev alsa-utils espeak i2c-tools libi2c-dev ssmtp ntpdate git-core v4l-utils cmake snese-hat sox libsox-dev libsox-fmt-all libcurl4-openssl-dev libssl-dev libjson-c-dev lsof pigpio

Libraries: userland wiringpi pigpio mqtt-paho tornado nanomsg nnpy py-nanomsg libmodbus