List of commands

This page lists all of the SPERTE commands.

Users

`SPERTE_Bin_To_Mat(file_list,N_signals,delete_file,precision)`

INPUTS

- (required) file_list = list of filename, e.g., 'filescope0.bin' or cell array of [N_names x 1] this is an output of `SPERTE_Get_Latest_Measurement`.
- (required) N_signals = number of signals, e.g., N_signals = 3. This is the amount of signals that you were logging in the model.
- (optional) delete_file = delete the .bin file? [1] Yes [0] No. default delete_file = 0
- (optional) precision = numerical precision of the .bin file, default precision = 'single' should ordinarily not be changed.

OUTPUTS

- result = matrix of [N_samples,N_signals,N_measurements] of measurement data. This only works if all measurements in `file_list` are of equal size.

Remarks

This function is used to convert the on demand binary measurements to .mat files that are ready to be loaded into Matlab. If you convert multiple measurements at the same time, make sure they are all of equal size.

`SPERTE_Forced_Restart(IP)`

INPUTS

- (required) IP = ip address of the pi you want to force restart.

OUTPUTS

- None

Remarks

This function uses the `PLINK.exe` file to establish an ssh with the pi located at IP and forces it to `sudo reboot now`. Usefull if a student connection gets stuck.

`file_list = SPERTE_Get_Latest_Measurement(p,filetype,N,folder)`

INPUTS

- (optional) p = raspi() object
- (optional) filetype = requested filetype, 'mat' or 'bin'
- (optional) N = Number of measurement files to retrieve, default N = 1
- (optional) folder = Path to place the files, default = current folder.

OUTPUTS

- file_list = names of files that were retrieved.

Remarks

This function is used to obtain the measurement files from the SPERTE setup and transfer them to the user PC. You can use it to obtain `.mat` files from the `To File` block. Or to obtain `.bin` files from the on demand measurement block. The output `file_list` is used as an input for `SPERTE_Bin_To_Mat` to covert the files to .mat files. **Note this is only required for `.bin` files.**

`SPERTE_Install_User()`

INPUTS

None

OUTPUTS

None

Remarks

This function is the main installation file for the SPERTE Tools. Run this function after obtaining a copy of the tools from Gitlab to install it to the local system.

`SPERTE_Measure_And_Collect(p,N_signals,N_samples,ModelName,ShowFigure)`

INPUTS

- (required) p = raspi() object

- (required) N_signals = number of signals, e.g., N_signals = 3
- (optional) N_samples = number of sampels to measure, default N_samples = 4000
- (optional) ModelName = model to get measurements form, default ModelName = gcs
- (optional) ShowFigure = show the new measurement in a new figure, default ShowFigure = 0

OUTPUTS

- result = matrix of [N_samples,N_signals,N_measurements] of measurement data

Remarks

This function performs all 3 steps to perform, retrieve and convert a measurement. This should mostly be used to perform a single measurement on the system.

`SPERTE_Model_Settings(name,Fs_m,tsim_m)`

INPUTS

- (required) name - modelname to change settings on.
- (required) Fs_m - New sample rate of the model [hz]
- (required) tsim_m - New simulation time of the model, [s] or Inf

OUTPUTS

None

Remarks

This function changes the model settings of the `modelname` model. It does so by changing the internal `initfnc()` callback of the model. This is visible in model properties.

`SPERTE_New_Model(name,Fs_m,tsim_m)`

INPUTS

- (optional) name - modelname
- (optional) Fs_m - Sample rate of the model [hz]
- (optional) tsim_m - Simulation time of the model, [s] or Inf

OUTPUTS

None

Remarks

This function creates a SPERTE compatible model with the name `modelname` in the current directory. Use this as a template to build more complex models. It initializes all the settings to their correct values.

`SPERTE_Perform_Measurement(N_samples,ModelName)`

INPUTS

- (optional) N_samples = number of sampels to measure, default N_samples = 4000
- (optional) ModelName = model to get measurements form, default ModelName = current model

OUTPUTS

None

Remarks

Use this function to initiate a new on demand measurement using the block in the model `ModelName`. Ensure that the model is running and currently open. Afterwards retrieve the measurement using `Sperte_Get_Latest_Measurement`.

`SPERTE_Restart(p)`

INPUTS

- (optional) p = raspi() object

OUTPUTS

None

Remarks

This function restarts the system located at `IP`.

This function also removes all user models and measurement files from the SPERTE system. Please ensure you have downloaded any relevant measurements first.

`SPERTE_Run_Basic_Test(Fs_m,p)`

INPUTS

- (optional) Fs_m = sampling rate of the model
- (optional) p = raspi() object

OUTPUTS

None

Remarks

This function creates a new SPERTE model in the current directory and compiles and runs it on the system located at `IP`. It then obtains the measurement file and analyzes the delay.

Please ensure the ADC1 and DAC1 of the Ebox are connected using a BNC Cable.

`SPERTE_Shutdown(p)`

INPUTS

- (optional) p = raspi() object

OUTPUTS

None

Remarks

This function shuts down the system located at `IP`.

This function also removes all user models and measurement files from the SPERTE system. Please ensure you have downloaded any relevant measurements first.

SPERTE_Stop_All_models(ip,user,pass)

Stop all models currently running on the Raspberry Pi

INPUTS

- (optional) ip = ip address of the pi
- (optional) user = username of the pi, default (pi)
- (optional) pass = Password of the pi, default (raspberry)

OUTPUTS

None

Remarks

This function stops all models that are currently running on the Raspberry Pi.

Developers

`SPERTE_Install_Ethercat_Libraries_Libraries(ip,user,pass)`

Note: Developers only!

INPUTS

- (optional) ip = ip address of the pi
- (optional) user = username of the pi, default (pi)
- (optional) pass = Password of the pi, default (raspberry)

OUTPUTS

None

Remarks

This function installs the libraries to communicate with the Ebox on the Raspberry Pi.

`SPERTE_Install_RT_Kernel(ip,user,pass)`

Note: Developers only!

INPUTS

- (optional) ip = ip address of the pi
- (optional) user = username of the pi, default (pi)
- (optional) pass = Password of the pi, default (raspberry)

OUTPUTS

None

Remarks

This function installs the RT kernel on the Raspberry Pi. It also asks to overclock the system and change the password.

This should only be run once on a fresh SD card image.

Internal

`SPERTE_Install_Wifi_Static_IP(p,sperte_nr)`

Install the wifi information and change the hostname and static ip

INPUTS

- (required) p = raspi() object
- (required) sperte_nr = nr of the sperte device

OUTPUTS

- IPS = New static IP on the wlan interface, default 192.168.201:255
- HOSTN = Hostname of the sperte, by default SPERTE1, SPERTE2 etc.

Remarks

`SPERTE_Install_Simulink_Library()`

Internal use only, do not call directly.

INPUTS

None

OUTPUTS

None

Remarks

This function is used internally in the installation to install the Simulink library.

`SPERTE_Mex_All_Lib()`

Internal use only, do not call directly.

INPUTS

None

OUTPUTS

None

Remarks

Function to compile all required .c files for the SPERTE installation.

`SPERTE_postGenFunc(h,buildInfo)`

Internal use only, do not call directly.

INPUTS

- (required) h
- (required) buildInfo

OUTPUTS

None

Remarks

This function adds the required `'-lec'' linke