DRAFT – uncomplete – Work in progress! HR 06.08.2020

Raspberry Pi Agent V4.2 - MindSphere V3 Version

## Content

[Content 1](#_Toc522281977)

[Introduction 2](#_Toc522281978)

[Prerequisites 2](#_Toc522281979)

[Hardware 2](#_Toc522281980)

[SD card 3](#_Toc522281981)

[Get connected 3](#_Toc522281982)

[Raspberry Pi Desktop and Directories 6](#_Toc522281983)

[Data Input File, generated by Python Scripts 6](#_Toc522281984)

[Commissioning / Steps on MindSphere Tenant 8](#_Toc522281985)

[Agent configuration / Entries in the agent configuration file 9](#_Toc522281986)

[Quick Start Guide for the RPi Agent to get Data uploaded to MindSphere 12](#_Toc522281987)

[Start with the 8GB RPi Image 13](#_Toc522281988)

[Procedure for MindSphere V3 13](#_Toc522281989)

[Raspbian / RPi Agent Installation from Scratch (at least 8GB formatted, empty SD Card required) 14](#_Toc522281990)

## Introduction

MCL V4.2 (MindConnect Library) based Software, running on Raspberry Pi V4/ V3B+ with Raspbian OS.

Lightweight and easy transportable.

Configuration via JSON File.

## Prerequisites

1. Raspberry Pi Hardware V4/V3 B+ , (V3 B also )
2. The SD card image from H.Rieger
3. Empty 8, 16 or 32GB SD Card for the RPi. Hi speed (e.g. Transcend Ultimate 633x)
4. Access to a MindSphere V3 tenant with admin rights
5. Internet access for the PI
6. A SD Card reader/writer with respective software on your PC

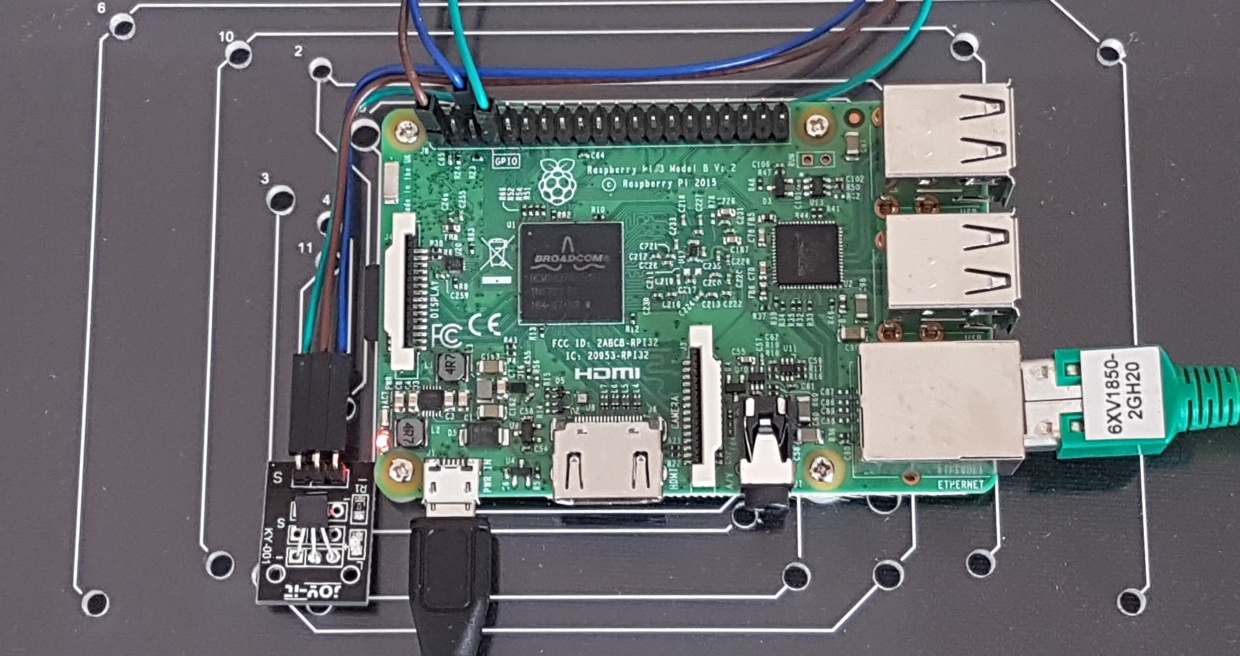
(e.g.: <https://sourceforge.net/projects/win32diskimager/>)

1. At least basic knowledge about RPi hardware and Raspbian OS
2. Optional: Temperature sensor DS1820B , distance sensor , rotary encoder , camera with Cables to connect the sensors / camera to the GPIO of the RPi

Sensors from JoyIT kit : http://sensorkit.joy-it.net

## Hardware

Raspberry Pi V3B+ with temperature sensor



## SD card

To find a fast SD card you may check:

<https://havecamerawilltravel.com/photographer/fastest-microsd-cards/>

Unzip the Image that you received. Copy your Image of the SD Card onto your empty SD card.

Format your card and make sure that there are no other (hidden) partitions:

https://www.sdcard.org/downloads/formatter\_4/

Then use the win32diskimager to put the image on the SD Card

https://sourceforge.net/projects/win32diskimager/

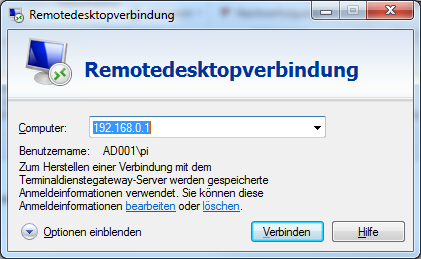
Plug the card into your RPi.

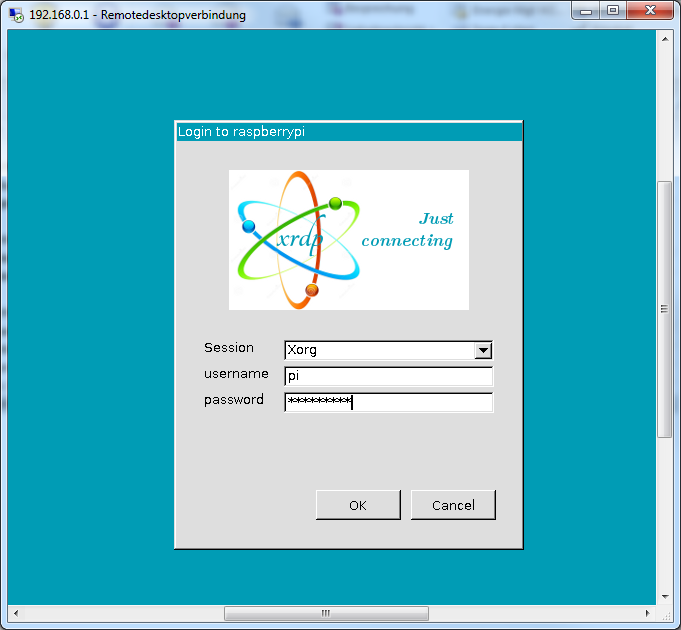
## Get connected

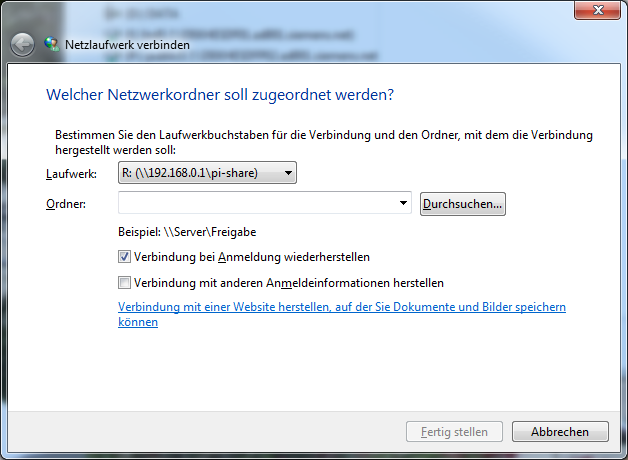
1. Connect your RPi with HDMI and USB to Screen, Keyboard and Mouse.
2. Power on the RPi, it will start with user pi.
3. Or you may connect it with a network cable with **Remote Desktop Connection** to your windows PC.

The address for connecting your pc to the RPi is **192.168.0.4**

**Username** on the logon screen is: pi

**Password** is: raspberry ( keyboard layout on Pi and PC must be the same ? else y /z may be changed on transmission to the pi !)



1. Connect to the shared directory (/home/pi) on the Raspberry

## Raspberry Pi Desktop and Directories

The RPi starts with its home directory: /home/pi

There you will find the executable RPiAgentV4 for MindSphere V3 Default is RPiAgentV4 with MCL library V4.2x.

Temporary directory: /tmp

This directory is already mounted to the RAM of the RPi. There the data input file gets stored / updated / deleted to save the SD card of the RPi from getting faulty too soon as the count of write cycles to an SD card is not endless.

## Data Input File generated by Python Scripts

You can edit and start the python scripts by opening Geany (Python Development tool) from the desktop. All examples are located in “/home/pi/DataInputScripts”.

As an example, open DataSimulation Input project by pointing on it , rightclick -> geany.

Press F5 to start the script. The script will be run in a terminal window. Alternatively, you may start the script directly using python V3 (python3) in a terminal window.

The DataSimulation script will provide datapoints with sinus, cosinus and sawtooth values each 5 second. Sinus and cosinus range is from -1 to +1, the sawtooth range is from 0 <x < 4\*pi\*10. Increment for x is 0.01 per 5s.

If you want to use the temperature sensor, you must use the TempSensor script and install the sensor to the GPIO of the RPi (in German, English installation guide is also easy to find)

<https://www.raspiprojekt.de/machen/basics/schaltungen/9-1wire-mit-temperatursensor-ds18b20.html?showall=1&limitstart>=

Later on, open the temperature sensor project with Geany. The 1-wirebus is already activated.

The agent checks every 10s for the data input file. If it is available its contents get read and are sent to MindSphere. After all data from the data input file has been sent, the data input file is deleted by the agent and the agent sleeps for 10s.

**Example of Input lines for data (RPiAgentDataInput.txt):**

12341123456789011234567890112345678901234567890123456

1001|0000000000| 0.0600|2018-05-29 10:27:29.073430

1002|0000000000| 0.9982|2018-05-29 10:27:29.073430

2001|0000000000| 0.6000|2018-05-29 10:27:29.073430

12345678901234567890123456789012345678901234567890123

1 2 3 4 5

**Description of fields:**

1001 = Datasource 1, Datapoint 001 (of Datasource 1) -> 4 chars

1002 = Datasource 1, Datapoint 002 (of Datasource 1)

2001 = Datasource 2, Datapoint 001 (of Datasource 2)

| = separator -> 1 char

0000000000 = OPC Quality code (0 = OK) -> 10 chars

| = separator -> 1 char

00000.0600 = numeric Value -> 10 chars including dot (10.4 format)

| = separator -> 1 char

2018-05-29 10:27:29.073430

= UTC time (YYYY-MM-DD\_HH:MM: SS. XXXXXX)

MindSphere is using max 3 digits after the dot (resolution = 1 ms) The agent cuts off the other digits.

**New:**

From V3.1 the agent splits the input lines by the separators. Now the lines are not fixed input position anymore. The parts may now differ from length, but the new entry is expected direct after the pipes “|”.

**Example of Input lines for events (RPiAgentEventInput.txt):**

0001|2018-08-17 11:05:42.598129|+|n

0001|2018-08-17 11:05:47.603823|-|n

**Description of fields:**

0001 = Event 1

| = separator -> 1 char

2018-08-17 11:05:42.598129 = UTC Timestamp of the event

| = separator -> 1 char

+ = coming event. (see explanation below)

| = separator -> 1 char

n = new -> not acknowledged, A = acknowledged

You may use “+”/ “-“ or your own text / value , it will be put in front of the description of the event on uploading the event. This may be useful if you want to indicate a coming / going event and a certain value connected to/ triggering this event.

## Commissioning / Steps on MindSphere Tenant

To transfer data to MindSphere you must prepare at least an asset that will be connected to the agent. This asset should be of an MCL type. Afterwards, you get the IAT and the credentials the agent needs to connect to MindSphere. This information must be transferred to the config file of the agent. It is convenient to use the chromium browser on the RPi to do that. In this case you can simply transfer the data from chromium to the config file with copy n’ paste.

## Agent configuration / Entries in the agent configuration file

Agent\_DebugFile = "";

MCL-Config:

{

MindSphereBaseUrl = "https:// Your MindSphere V3 Base Url ";

Port = 443;

MindSphereCertificate = "-----BEGIN CERTIFICATE-----\nMIIFYDCCA0igAwIBAgIURFc0JFuBiZs18s64KztbpybwdSgwDQYJKoZIhvcNAQEL\nBQAwSDELMAkGA1UEBhMCQk0xGTAXBgNVBAoTEFF1b1ZhZGlzIExpbWl0ZWQxHjAc\nBgNVBAMTFVF1b1ZhZGlzIFJvb3QgQ0EgMiBHMzAeFw0xMjAxMTIxODU5MzJaFw00\nMjAxMTIxODU5MzJaMEgxCzAJBgNVBAYTAkJNMRkwFwYDVQQKExBRdW9WYWRpcyBM\naW1pdGVkMR4wHAYDVQQDExVRdW9WYWRpcyBSb290IENBIDIgRzMwggIiMA0GCSqG\nSIb3DQEBAQUAA4ICDwAwggIKAoICAQChriWyARjcV4g/Ruv5r+LrI3HimtFhZiFf\nqq8nUeVuGxbULX1QsFN3vXg6YOJkApt8hpvWGo6t/x8Vf9WVHhLL5hSEBMHfNrMW\nn4rjyduYNM7YMxcoRvynyfDStNVNCXJJ+fKH46nafaF9a7I6JaltUkSs+L5u+9ym\nc5GQYaYDFCDy54ejiK2toIz/pgslUiXnFgHVy7g1gQyjO/Dh4fxaXc6AcW34Sas+\nO7q414AB+6XrW7PFXmAqMaCvN+ggOp+oMiwMzAkd056OXbxMmO7FGmh77FOm6RQ1\no9/NgJ8MSPsc9PG/Srj61YxxSscfrf5BmrODXfKEVu+lV0POKa2Mq1W/xPtbAd0j\nIaFYAI7D0GoT7RPjEiuA3GfmlbLNHiJuKvhB1PLKFAeNilUSxmn1uIZoL1NesNKq\nIcGY5jDjZ1XHm26sGahVpkUG0CM62+tlXSoREfA7T8pt9DTEceT/AFr2XK4jYIVz\n8eQQsSWu1ZK7E8EM4DnatDlXtas1qnIhO4M15zHfeiFuuDIIfR0ykRVKYnLP43eh\nvNURG3YBZwjgQQvD6xVu+KQZ2aKrr+InUlYrAoosFCT5v0ICvybIxo/gbjh9Uy3l\n7ZizlWNof/k19N+IxWA1ksB8aRxhlRbQ694Lrz4EEEVlWFA4r0jyWbYW8jwNkALG\ncC4BrTwV1wIDAQABo0IwQDAPBgNVHRMBAf8EBTADAQH/MA4GA1UdDwEB/wQEAwIB\nBjAdBgNVHQ4EFgQU7edvdlq/YOxJW8ald7tyFnGbxD0wDQYJKoZIhvcNAQELBQAD\nggIBAJHfgD9DCX5xwvfrs4iP4VGyvD11+ShdyLyZm3tdquXK4Qr36LLTn91nMX66\nAarHakE7kNQIXLJgapDwyM4DYvmL7ftuKtwGTTwpD4kWilhMSA/ohGHqPHKmd+RC\nroijQ1h5fq7KpVMNqT1wvSAZYaRsOPxDMuHBR//47PERIjKWnML2W2mWeyAMQ0Ga\nW/ZZGYjeVYg3UQt4XAoeo0L9x52ID8DyeAIkVJOviYeIyUqAHerQbj5hLja7NQ4n\nlv1mNDthcnPxFlxHBlRJAHpYErAK74X9sbgzdWqTHBLmYF5vHX/JHyPLhGGfHoJE\n+V+tYlUkmlKY7VHnoX6XOuYvHxHaU4AshZ6rNRDbIl9qxV6XU/IyAgkwo1jwDQHV\ncsaxfGl7w/U2Rcxhbl5MlMVerugOXou/983g7aEOGzPuVBj+D77vfoRrQ+NwmNtd\ndbINWQeFFSM51vHfqSYP1kjHs6Yi9TM3WpVHn3u6GBVv/9YUZINJ0gpnIdsPNWNg\nKCLjsZWDzYWm3S8P52dSbrsvhXz1SnPnxT7AvSESBT/8twNJAlvIJebiVDj1eYeM\nHVOyToV7BjjHLPj4sHKNJeV3UvQDHEimUF+IIDBu8oJDqz2XhOdT+yHBTw8imoa4\nWSr2Rz0ZiC3oheGe7IUIarFsNMkd7EgrO3jtZsSOeWmD3n+M\n-----END CERTIFICATE-----";

Security\_Profile = 0;

Proxy\_Hostname = "";

Proxy\_Port = 0;

Proxy\_Type = -1;

Proxy\_Username = "";

Proxy\_Password = "";

Proxy\_Domain = "";

Max\_Http\_Payload\_Size = 65536;

Http\_Request\_Timeout = 300;

TimeStamp\_Precision = 0;

User\_Agent = "custom agent v1 0";

Initial\_Access\_Token = "Get the IAT from MindSphere and put in here";

Tenant = "Your MindSphere tenant name";

Agent\_SleepTime = 10;

Agent\_StartDelay = 10;

MCL\_DebugLevel = 4;

MCL\_DebugFile = "";

DataLakeCertificate= …

};

Data-Model:

{

DataConfig\_ID = "WillBeDoneByAgent";

TimeSeriesVersion = "1.0";

DataSourceVersion = "1.0";

DataInputFile = "/tmp/RPiAgentDataInput.txt";

DataSource-Count = 1;

DataSource1-DataPointsCount = 1;

DataSource1-Description = "ToBeDone DataSrc Description 1";

DataSource1-Name = "ToBeDone DataSrc Name 1";

DataPoint1001-Name = "ToBeDone DataPt Name 1";

DataPoint1001-DataType = "ToBeDone DataPt DataType 1";

DataPoint1001-Unit = "ToBeDone DataPt Unit 1";

DataPoint1001-Description = "ToBeDone DataPt Description 1";

DataPoint1001-MinDeltaVal = "0%";

DataPoint1001-MaxDeltaTime = 0;

DataPoint1001-ID = "ToBeDone";

Event-Count = 1;

EventInputFile = "/tmp/RPiAgentEventInput.txt";

Event0001-Source = "Motor 1";

Event0001-Description = "High Temperature";

Event0001-Severity = 2;

FileUploadDirectory = "/tmp/RPiAgentFiles2Upload/";

};

**Agent\_DebugFile** = "";

If you need debug information from the agent, enter a valid path/filename. If left empty no debug info is written. Example: “/tmp/AgentDebugTrace.txt”;

**MindSphereBaseUrl** = "https:// Your MindSphere V3 Base Url ";

Example: "https://southgate.eu1.mindsphere.io";

**MindSphereCertificate** = … nothing to do , in case you should get a certificate error contact SIOS support or extract the certificate from <https://southgate.eu1.mindsphere.io> website in base64 format.

DataLakeCertificate= … nothing to do. Use the certificate as it comes with the agent.

**Proxy entries** – only to be done if you are using one.

**Security\_Profile** = 0; - don’t change

0 = MCL\_SECURITY\_SHARED\_SECRET, //!< Shared secret.

1 = MCL\_SECURITY\_RSA\_3072 //!< RSA 3072.

**Max\_Http\_Payload\_Size** = 10485760; // 1024\*1024\*10 = 10MB i.e Files up to 10MB can be uploaded .Increase if you get messages that data was lost due to memory problems.

**Initial\_Access\_Token** = "Get the IAT from MindSphere and put in here"; // from asset manager

**Tenant** = "Your MindSphere tenant name"; // from asset manager

**Agent\_SleepTime** = 10; // in s, before checking for new input

**Agent\_StartDelay** = 10; // in s, gives time to read the disclaimer on start

**MCL\_DebugLeve**l = 4; // if you need to debug the underlying MCL functions

MCL\_LOG\_UTIL\_LEVEL\_VERBOSE 1

MCL\_LOG\_UTIL\_LEVEL\_DEBUG 2

MCL\_LOG\_UTIL\_LEVEL\_INFO 3

MCL\_LOG\_UTIL\_LEVEL\_WARN 4

MCL\_LOG\_UTIL\_LEVEL\_ERROR 5

MCL\_LOG\_UTIL\_LEVEL\_FATAL 6

MCL\_LOG\_UTIL\_LEVEL\_NONE 255 (0xFF)

**MCL\_DebugFile** = "";

If you need debug information from the MCL, enter a valid path/filename. If left empty, no debug info is written. Example:"/tmp/MclDebugTrace.txt";

**Data / Timeseries Upload:**

**DataConfig\_ID** = "WillBeDoneByAgent"; If a Data configuration is transmitted to Mindsphere for the first time or because of change in the data set (data points / data sources) then the entry must be = "WillBeDoneByAgent" . Following, the agent fills a UUID in this entry.

**DataInputFile** = "/tmp/RPiAgentDataInput.txt"; // This is the file the agent checks for new input data. The same as in the script files.

**DataSource-Count** = 1; // the count of datasources Range: 1-9

**DataSource1-DataPointsCount** = 1; // The first Datasource has one Datapoint. Range: 1-999

**DataSource1-Description** = "DS1 with 2DPs"; // Description

**DataSource1-Name** = "RaspberryDataSource1"; // Name

**DataSource1-CustomField** = ""; // optional

**DataSource1-CustomValue** = ""; // optional

**DataPoint1001-Name** = "ToBeDone DataPt Name 1"; // enter the Name of the datapoint

…

**DataPoint1001-DataType** = "ToBeDone DataPt DataType 1"; // to be replaced by INT, LONG, DOUBLE, BOOLEAN, STRING

**DataPoint1001-ID** = "ToBeDone"; // if “ToBeDone” is found by the agent an UUID will be generated.

**DataPoint1001-Unit** = "ToBeDone DataPt Unit 1"; // enter the Unit

**DataPoint1001-Description** = "ToBeDone DataPt Description 1"; // enter a description

**DataPoint1001-MinDeltaVal** = "0%"; // send a datapoint to MindSphere if its value has changed more than x%

**DataPoint1001-MaxDeltaTime** = 0; // in s, if a value has not changed but you need a datapoint in a timeseries at least every x seconds, (but if there is no input for this datapoint and the time has expired no value will be sent)

If you want every datapoint to be sent, leave MinDeltaVal and MaxDeltaTime at the default (0).

**DataPoint1001-CustomField** = ""; // optional

**DataPoint1001-CustomValue** = ""; // optional

**DataPoint1001-Map2AssetID** = "4e37b1fa4f5142b0b8314528369bfc89"; // ID of the asset in which the Agent Asset is defined

**DataPoint1001-Map2AspectSet** = "RPiDemoHRTemperature"; // Aspect

**DataPoint1001-Map2AspectVar** = "Temperature"; // Aspect Variable

**DataPoint1001-KeepMapping = 1;** // if the data model gets updated this mapping will be kept

**DataPoint1001-MapStatus** = "MCL\_OK"; // Status received from Mindsphere after Mapping

If you need more data sources and data points, you can expand the config file respectively.In total you can use up to 1000 Datapoints, summed up from all Datasources.

**Events:**

Event-Count = 1; // The count of events in the config file

EventInputFile = "/tmp/RPiAgentEventInput.txt"; // the name of the file that contains the event input for the agent

Event0001-Source = "Motor 1"; // Source of Event 0001

Event0001-Description = "High Temperature"; // Description of event 0001

Event0001-Severity = 2; // 0 = Error , 1 = Warning , 2 = Information

**File Upload:**

FileUploadDirectory = "/tmp/RPiAgentFiles2Upload/"; // The directory the agent scans for file upload

**Data Lake File Upload:**

DataLakeFileUploadDirectory = "/tmp/RPiAgentDataLakeFiles2Upload/"; // The directory the agent scans for file upload

DataLakeSubTenant = "";

DataLakeUploadUrl = "RPiAgentHR"; // the directory Name to find the File.

Additional information about how to configurate the MCL can be found here:

<https://developer.mindsphere.io/resources/resources-mclib/index.html>

## Quick Start Guide for the RPi Agent to get Data uploaded to MindSphere

1. Define your data model. Define the count of your data sources and the count of datapoints per data source.
2. Define an asset for the agent in the asset manager within your tenant. Tenant Admin rights required.
3. Transfer the connection credentials from the asset manager to the config file of the agent. I.e. IAT, tenant name and MindSphereBaseUrl.
4. Implement your data model in the config file of the agent.
5. Check and test your data input scripts for the agent.
6. Start your data input scripts and let them run continuously.
7. Start the agent. Onboarding will take place . If there are Data to be uploaded the agent will start uploading.
8. Check the agent’s output for errors. If errors occur , the agent exits . You may also stop the agent anytime with Ctrl -C. If required activate the debug / trace function in the config file of the agent, correct errors and repeat.
9. Check the results in the MindSphere asset manager i.e. is the datamodel available ?
10. Do the data mapping in the asset manager to get / see values in fleetmanager.
11. Start the data input scripts, then start the agent. Timeseries / events/ Files should be successfully transmitted by the agent.
12. Check the results in the fleet manager.

## Start with the 8GB RPi Image

This is a fully pre-configured SD Card image from an 8GB SD Card.

Containing the RPi Agent V3.1 and all required libraries (MCL, SSL, CURL, etc.)

For the first steps to get started watch the introduction on Video and read the following instructions:

<https://myvideo.siemens.com/media/RPiAgentFirstSteps/1_hu5q2327>

### Procedure for MindSphere V3

1. Define a new Asset in MindSphere for the RPi Agent of the type MindConnect Library
2. Check your internet connection.
3. Start the web browser on the RPi and try to connect to Google or MindSphere

If you want to start with the pre-installed Data Model take the sample configuration from

**/home/pi/RPiAgentV4Sample.cfg** and copy it to **RPiAgentV4.cfg**

* Open the config file with double click.
* Get your IAT from MindSphere and enter it.
* Enter your tenant name.

Start the datainput scripts for the sensors -- if connected –

else start the data simulation script with the desktop icon.

The Data Point IDs from the pre-installed input scripts are adjusted with the sample configuration script.

**How to start the RPi Agent and to keep the Terminal open:** (recommended)

open a terminal window and enter:

**/home/pi/RPiAgentV4**

Or (but terminal closes at the end...)

* start from the desktop icon and choose the Button:

(Execute in Terminal)

If everything is configured right, the RPi Agent will be onboarded to MindSphere.

You will find the data model and the mapping of the config file in the previously defined Asset for the RPi Agent.

## Raspbian / RPi Agent Installation from Scratch (at least 8GB formatted, empty SD Card required)

(Request the RPi Agent and the libraries from [Horst.Rieger@Siemens.com](mailto:Horst.Rieger@Siemens.com))

To find a fast SD card you may check:

<https://havecamerawilltravel.com/photographer/fastest-microsd-cards/>

Get the zipped NOOBS installer:

<https://www.raspberrypi.org/downloads/noobs/>

**Follow:** Extract the Zip File.

Follow the instructions in **INSTRUCTIONS-README.txt** in the Zip File to install Raspbian.

* Furthermore, copy all files from the Zip file to the (root of the) SD Card
* Connect USB Keyboard and USB Mouse to RPI.
* Connect the HDMI Video Interface to your screen.

Maybe you need an adapter depending what Video interface your screen supports (e.g. HDMI <-> VGA)

* Plug the SD card into the RPI. Power on the RPI.
* Choose Raspbian. The Installation can take a while, just wait a few minutes.
* Choose your language and keyboard in the settings of Raspbian. Preferences -> RPI Configuration.
* Set the eth0 interface to 192.168.0.1 (for remote desktop and file sharing, optional)
* Connect the wlan0 interface to your WIFI Network for internet access
* Install samba: “sudo apt-get install samba” (optional)
* Open a terminal, enter:

“sudo leafpad /etc/samba/smb.conf”

Enter the following section at the end of the file.

[pi-share]

Comment = Pi shared folder

Path = /home/pi

public = yes

only guest = yes

browseable = yes

read only = no

writeable = yes

create mask = 0644

directory mask = 0755

force create mask = 0644

force directory mask = 0755

force user = root

force group = root

See: <https://www.raspberrypi.org/magpi/samba-file-server/>

Then mount the \tmp directory in RAM to get a fast file transfer between data acquisition script and RPi agent. And extend the life of the SD Card… as it is written much less.

* Open a terminal.
* Enter “sudo leafpad /etc/fstab”.

Add following line:

“tmpfs /var/tmp tmpfs nodev,nosuid,size=100M 0 0”

Reboot and check with “df –h” if /tmp was mounted into RAM.

See: <https://www.hellojona.com/2017/06/create-a-ram-disk-tmpfs-in-raspberry-pi-3/>

* Copy the lib folder to /home/pi
* Copy all library files to the RPi system:

“sudo cp /home/pi/libs/\*.\* /usr/lib”

* Copy all agent files to /home/pi
* Set all properties of the copied files to “anyone”

To automatically create the file upload directory in the /tmp directory after each reboot of the RPi insert the following command line in the autostart file:

/home/pi/.config/lxsession/LXDE-pi/autostart

“mkdir /tmp/RPiAgentFiles2Upload”

If you need to change file properties or copy files with root rights you can start the filemanager from a console window by typing “sudo pcmanfm”