Temperature Measurement App on ESP32-devkitC

This guide introduces how to create and commission a temperature measurement app in esp-matter.

Prerequisites

1. A PC with WSL/virtual computer installed or a MacBook.
2. esp-idf and esp-matter installed on the platform of choice.
3. ESP32-devkitC V4.
4. Raspberry Pi with installation according to the instructions in raspi.docx.

This guide follows the WSL method as the author has used it.

On Windows Machine with WSL Installed

1. Install and enable Windows Subsystem for Linux 2 (WSL2).
2. Install Ubuntu 20.04 or 22.04 from the Windows App Store.
3. Start Ubuntu (search in the start menu) and run the command `uname -a`. It should report a kernel version of 5.10.60.1 or later. If not, upgrade WSL2 by running `wsl --upgrade` from Windows PowerShell.
4. Windows does not support exposing COM ports to WSL distros. Install `usbipd-win` on Windows and WSL (`usbipd-win WSL Support`).
5. From this point, the process for setting up esp-matter and building examples is the same as on other hosts.
6. Clone the repositories from inside the WSL environment and not inside a mounted directory.

Set-up at WSL

Install prerequisites

```
sudo apt update
sudo apt upgrade
sudo apt-get install git wget flex bison gperf python3 python
```

Download esp-idf repository (run this at home dir)

```
mkdir -p ~/esp
cd ~/esp
git clone --recursive https://github.com/espressif/esp-idf.gi
```

Install tools

```
cd ~/esp/esp-idf
./install.sh esp32
```

Download esp-matter repository

```
cd esp-idf
source ./export.sh
cd ..

git clone --recursive https://github.com/espressif/esp-matter
cd esp-matter
./install.sh
cd ..
```

Do this every time when opening a new session (Creates venv)

```
cd esp-idf; source ./export.sh; cd ..
cd esp-matter; source ./export.sh; cd ..
```

Usage at WSL

Do this every time when opening a new session (Creates venv)

```
cd esp-idf; source ./export.sh; cd ..
cd esp-matter; source ./export.sh; cd ..
```

Go to the temperature measurement app

```
cd ~/esp/esp-matter/connectedhomeip/connectedhomeip/examples/
```

Make sure that you have shared the USB with usbipd from the host computer. Then open menuconfig with

```
idf.py menuconfig
```

Turn on Wi-Fi from Rendevzeus mode and also change chip-device settings in the Wi-Fi section (`ssid` and `wpa`) to corresponding values.

Then go

```
idf.py build flash monitor
```

This should be good at the WSL end.

Setup at Raspberry Pi

Open chip-tool with

```
./chip-tool interactive start
```

Then choose

```
pairing onnetwork <this can be anything e.g. 1337> <20202021>
```

After this, the monitor at WSL should say that the connection is successful.

Then you can read values by:

```
temperaturemeasurement read measured-value 1337 1
```

Success!