

Anleitung C++ AMIS CI Client mit MBUS-Library

18.11.2019, Dr. Markus Flohberger

1. Die für C++ Entwicklung notwendigen Programme und Bibliotheken installieren (z.B. build-essential). Unter Raspbian oder Ubuntu benötigt der Client folgende Pakete: `libstdc++x-x.x-dev`, `libconfig++-dev` und `libsqlite3-dev`. Unter Manjaro sind es folgende Pakete: `libstdc++x`, `libconfig` und `sqlite3`.
2. Die M-Bus Bibliothek extrahieren (`libmbus-0.8.0`), in das entpackte Verzeichnis wechseln und diese mittels `./configure`, `make` und `make install` installieren. Die Library muss in den Pfad hinzugefügt werden. Dazu in der `.bashrc` im Home-Verzeichnis am Ende folgende Zeilen hinzufügen: `LD_LIBRARY_PATH=/usr/local/lib` und `export LD_LIBRARY_PATH`.
3. Das Archiv mit dem C++ AMIS CI Client (`mbus-slave-ima-cpp-webdemo.7z`) extrahieren. In dem Ordner `make` ausführen um das Programm zu kompilieren.
4. Standardmäßig schreibt der Client die aufgezeichneten Werte in eine Sqlite-Datenbank (Echtzeit und Historisch) im Verzeichnis des Clients. In der C++ Datei `mbus-slave.cpp` können durch die Variabel `FILENAME_DATABASE` der Dateiname und Speicherort geändert werden. Das Programm muss dann neu kompiliert und gegebenenfalls die notwendigen Berechtigungen für das Zielverzeichnis gesetzt werden.
5. Bei Manjaro: Zugriffsberechtigung (Gruppe, z.B. `uucp`) auf USB-Devices prüfen z.B. `ls -l /dev/ttyUSB*`. Gruppenzugehörigkeit user prüfen `id -Gn user`. User zu Gruppe hinzufügen `sudo usermod -a -G uucp user`. Aus- und Einloggen.
6. Zahlreiche Parameter können in der Konfigurationsdatei `config.ini` im Programmordner angepasst werden.
7. Den AES-Schlüssel in der Datei `config.ini` eingeben, z.B. `aeskey = "00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF";`
8. Bestimmen welches USB-Gerät der optische Lesekopf ist, z.B. durch aufrufen von `tail -f /var/log/messages` mit anschließendem Abziehen und Anstecken des USB-Steckers des Lesekopfs oder mit `dmesg`. Es ergibt sich z.B. `/dev/ttyUSB0`. Dieses Device in der Datei `config.ini` (z.B. `device1 = "/dev/ttyUSB0";`) eingeben.
9. Das Programm im Programmordner mit `./mbus-slave` aufrufen.
10. Das Programm schreibt standardmäßig jede Sekunde in die Echtzeit-Datenbank und alle 15min in die historische Datenbank, dies kann durch editieren der entsprechenden Variablen in der Datei `config.ini` geändert werden.
11. Das Programm kann beim Booten automatisch gestartet werden z.B mit einem Init-Script oder einem `rc.local` Script.