SIP4.0

ERGEBNISBERICHT

Task / Meilenstein x.y

Portierung des Python Projekts in C++

Autor(en): Maximilian Papa, BSc

Version: 1.0

Datum: 02.02.2018

# Einleitung

Nachdem nun wie vorher beschrieben auch eine Bahnplanung und somit eine Art der künstlichen Intelligenz in das Programm eingebaut werden muss wird das Programm immer mächtiger und durch python leidet die Einfachheit der Struktur, Vererbung und die Datenkapselung. Deshalb soll das python Programm auf C++ umgeschrieben werden. Es wurde nach einer geeigneten Entwicklungsumgebung unter Raspbian gesucht. Dabei wurde CodeLite ausgewählt, da es sehr einfach und gut funktioniert. Alternativ könnte auch Eclipse verwendet werden. <https://developer-blog.net/raspberry-pi-cplusplus-entwicklungsumgebung/>

Die serielle Kommunikation und die Kommunikation mittels MQTT Protokoll wurden unter python schon umgesetzt, müssen aber auch in das neue C++ Projekt mit importiert werden.

# Serielle Kommunikation

In dieser Arbeit wurde die WiringPi-Bibliothek schon einmal für die Steuerung der GPIO-Pins über die Konsole verwendet und installiert. Nach einer Recherche hat sich herausgestellt, dass diese Bibliothek auch die Möglichkeit einer seriellen Kommunikation bietet und wurde deshalb genau für diese Aufgabe getestet: <https://raspberrypi.stackexchange.com/questions/38830/serial-communication-c-c>

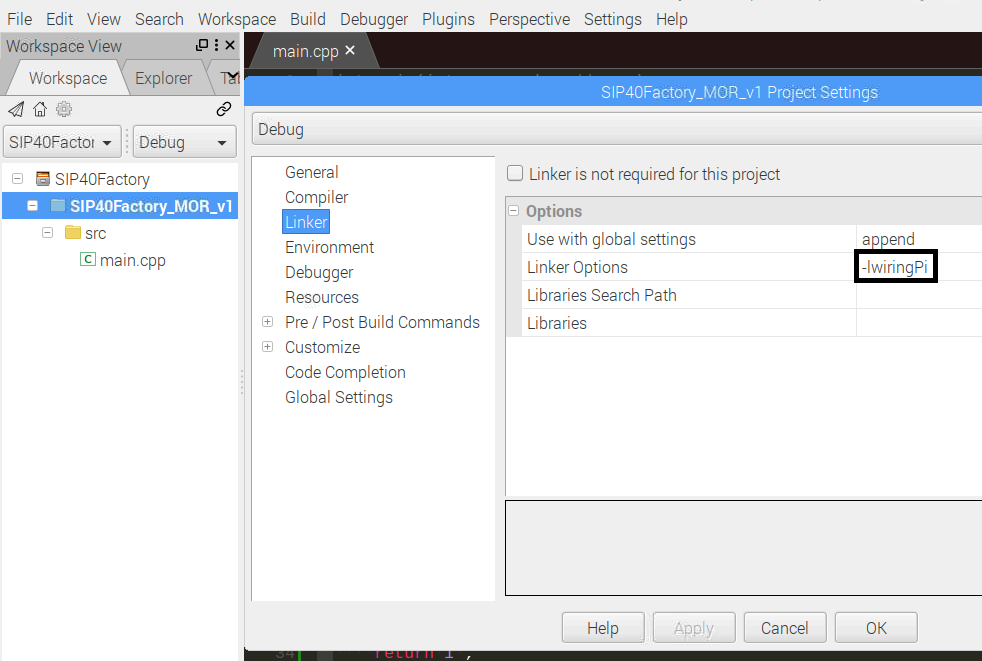
<https://www.youtube.com/watch?v=g-YVWNxSW2U>

Die verschiedenen Methoden dieser Bibliothek sind unter folgendem Link einsehbar: <http://wiringpi.com/reference/serial-library/> und falls die Installation nicht mehr vorhanden sein sollte, kann die Bibliothek mittels <http://wiringpi.com/download-and-install/> wieder installiert werden.

Ein Testprogramm wurde in diesem Fall ebenfalls unter <https://github.com/WiringPi/WiringPi/blob/master/examples/serialTest.c> gefunden.

Beim Testen im Programmcode wurde die Bibliothek zwar mit #include gefunden, das Programm konnte jedoch noch nicht ausgeführt werden, da dem Linker die Verknüpfungen zu den richtigen Methoden fehlten. Um dieses Problem zu lösen, musste die Bibliothek manuell zum linker hinzugefügt werden. <https://stackoverflow.com/questions/34121370/how-do-i-add-libraries-to-codelite-projects>

Im unteren Bild ist die vorgenommene Einstellung in der schwarzen Markierung zu sehen. Um dorthin zu gelangen, muss mit einem rechtsklick auf das Projekt die Einstellungen dieses Projekts geöffnet werden. Unter dem Tab “Linker” sind die Einstellungen dann zu finden.



# MQTT Protokoll

Für das MQTT Protokoll war es schon schwerer eine passende Umsetzung zu finden. Es gibt mehrere Bibliotheken, welche in Betracht gezogen wurden. Der Großteil in Ubuntu ist mit Eclipse umgesetzt, aber es sollte auch mit dem einfacheren CodeLite funktionieren.

<https://github.com/mqtt/mqtt.github.io/wiki/libraries>

<https://www.eclipse.org/paho/clients/cpp/>

<https://debihiga.wordpress.com/using-mqtt-in-python-c-and-c-for-embedded-and-non-embedded-systems/>

<https://github.com/debihiga/mqtt_paho_cpp_subscriber>

Da aber für den MQTT-Broker eh schon Mosquitto installiert war, wurde eine Möglichkeit gesucht gleich diese Bibliothek zu verwenden. Die Idee dafür kam von einer Bibliothek, welche auf die mosquitto-Bibliothek aufsetzt: <https://github.com/OpenSensorsIO/raspberry-pi-mqtt>

Auf dem Raspberry Pi sind die meisten verfügbaren Bibliotheken im Ordner “/usr/include/ zu finden. Zum Programmieren mit der mosquitto-Bibliothek wird eine “mosquitto.h” verwendet. Diese war aber noch nicht in diesem Ordner. Also musste diese Bibliothek nachinstalliert werden. Um das zu machen, wurde ähnlich wie schon einmal bei der Mosquitto Installation “sudo aptitude install libmosquitto-dev” verwendet. Damit werden die Abhängigkeiten zu anderen Paketen ignoriert und es sollte alles installiert werden. Nach der ausführung dieses Befehls war die “mosquitto.h”-Datei auch wirklich im “/usr/include/”-Ordner vorhanden und konnte nun mit “#include <mosquitto.h>” im Programm benutzt werden. Auch bei dieser Bibliothek musste “-lmosquitto” in den Linker-Einstellungen gesetzt werden. Wichtige Methoden dieser Bibliothek sind unter folgendem Link einsehbar: <https://mosquitto.org/api/files/mosquitto-h.html>

Und ein Testprogramm wurde unter folgendem Link festgehalten: <http://kevinboone.net/mosquitto-test.html>