Pyduino

Inhalt

[Kurzfassung 1](#_Toc123284603)

[Problem 1](#_Toc123284604)

[Quellen 1](#_Toc123284605)

# Kurzfassung

# Problem

Die Programmiersprache mit der der Arduino Microcontroller programmiert wird ist C/C++, mit einigen zusätzlichen Hilfsfunktionen, mit denen zum Beispiel die Ein- und Ausgänge des Arduinos angesteuert werden können *[Q1]*. Diese Programme können mit der Arduino Entwicklungsumgebung erstellt werden. Diese stellt Grundlegende Funktionen zur verfügung, wie zum Beispiel Syntax Highlightung und die Möglichkeit Programme zu kompilieren und auf den Arduino hochzuladen *[Q2]*. Die Arduino Entwicklungsumgebung basiert auf dem Arduino CLI, das zum Beispiel für das Kompilieruen und hochladen der Programme zuständig ist. Diese Funktionen sind aber relativ langsam, selbst das kompilieren und hochladen von einem lehren Program dauert:

|  |  |  |  |
| --- | --- | --- | --- |
| Platform *(siehe Quellen)* | Kompilieren | Hochladen | Gesamt |
| PC | 0.74s | 1.85s | 3.59s |
| Laptop 1 | 4.43s | 2.78s | 7.21s |
|  |  |  |  |
|  |  |  |  |

*(Die Zeiten wurden mit diesen Programmen ermittlet:* [*https://github.com/Bergschaf/Arduino-Benchmark*](https://github.com/Bergschaf/Arduino-Benchmark)*)*

Man kann zwar sagen, dass vor allem die Zeit, die für das Kompilieren benötigt wird, bei größeren Programmen nur leicht steigt. Die Zeit die fürs hochladen benötigt wird steigt dagegen etwas bei größeren Programmen. Diese Zeiten sind aber sehr lang, da der Nutzer jedes mal bis zu 7s warten muss, nur um den Effekt einer kleinen Änderung im Programm zu sehen. Das liegt daran, dass die Arduino Programme kompiliert und auf den Arduino Microcontroller hochgeladen werden müssen, während zum Beispiel Python Programme von einem Interpreter ausgeführt werden, was fast ohne Zeitverzögerung funktioniert.

Die Arduino IDE (Integratet Development Environment: Entwicklungsumgebung) stellt einige Features zur Verfügung, die das Programmieren des Arduino Microcontrollers einfacher machen. Ein Beispiel dafür ist Syntax Highlighting, bestimmte syntaktische Elemente werden im Editor farbig markiert, was den Code deutlich lesbarer macht. Außerdem bietet die IDE Autovervollständigung an, die allerdings etwas umständlich ist.

Da der Arduino oft von Anfängern verwendet wird, um in die Programmierung einzusteigen ist auch die Programmiersprache ein Problem. C/C++ ist eine low-level Programmiersprache, bei der dem Nutzer weniger „abgenommen“ wird als bei high-level Programmiersprachen wie zum Beispiel Python. Das führt dazu, dass Programmieranfänger nicht nur grundlegende Programmierkonzepte, sondern auch spezielle Eigenheiten von C/C++ lernen müssen. Um zum Beispiel die länge eines Array zu ermitteln muss die Länge der Gesamten Datenstuktur in Bytes durch die Länge eines Arrayelements in Bytes geteilt werden. In Python dagegen kann einfach die len() Funktion benutzt werden. Arrays sind generell ein erschwerender Faktor bei der Programmierung des Arduinos. Solche Datenstrukturen sind in Python als verlinkte liste, bei der einfach Elemente mit verschiedenen Datentypen angehängt und entfernt werden können, implementiert. Beim Arduino dagegen können Arrays nur mit fester länge und einem Vorher festgelegten Datentypen definiert werden. Dies ist zwar deutlich schneller und Effizienter als das Python Gegenstück. Für Anfänger ist es dagegen deutlich intuitver, wenn Elemente aus Listen einfach entfernt und eingefügt werden können.

# Quellen

*[Q1]* <https://en.wikipedia.org/wiki/Arduino#Sketch>

*[Q2]* [*https://github.com/arduino/arduino-ide*](https://github.com/arduino/arduino-ide)

PC: CPU: AMD Ryzen 5 5600X @3,7GHz 6-Cores 12-Threads; RAM: 16gb; OS: Windows 11

Laptop1: CPU: Intel Pentium N3710 @1,6GHz 4-Cores 4-Threads; RAM: 4gb; OS: Windows 10

Laptop2 (Linux):

Laptop2 (Windows):