Sensorimotor

Schlussbericht

Zuwendungsempfänger:

JustScheunemann Sensorimotor GbR

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS18S35 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Wir entwickeln Open-Source-free-Design Hardware und Software für Robotik-Antriebe und wollen so die vielen kleinen Robotik-Projekte weltweit fördern und durch kostengünstige und modifikationsfreundliche Hardware die Kreativität der Entwickler*innen, Forscher*innen und Bastler*innen anregen und sie ein Stück weit unabhängiger von bestehenden industriellen Lösungen machen. Wir haben unsere Wurzeln in der universitären Robotik-Forschung und in der Maker-Szene und möchten unsere Entwicklungen den Robotikenthusiasten weltweit zugänglich machen.

Roboter und KI halten Einzug in unseren Alltag. Wir glauben, dass die Entwicklung für die Menschheit vorteilhaft und sicher sein kann - wenn die zugrundeliegende Technologie frei von Interessen der Wenigen ist und in die Hände von Forscher*innen und Entwickler*innen weltweit gelegt wird. Hierfür sind freie Hardware-Designs entscheidend, denn sie stärken die Freiheit der Nutzer*innen und geben ihnen wieder Kontrolle über ihre Hardware. Mit freien Hardware-Designs können Maschinen auch endlich wieder repariert und modifiziert werden. Wir möchten zur Demokratisierung der Robotik beitragen, indem wir ein völlig freies Hardware-Design und die dazugehörige Software für Robotik-Antriebe entwickeln.

Ziel unseres Projektes war es, die Nutzer*innen in die Lage zu versetzen, einfache Motoren in SmartServos zu verwandeln und somit für beliebige Robotikaufgaben einsetzbar zu machen. Hierfür haben wir eine Platine entwickelt, die mit zahlreichen DC-Motoren kompatibel ist und mit minimalem Aufwand angeschlossen werden kann. Mit ihr erhalten Nutzer*innen ein reichhaltiges sensorisches Feedback vom Motor (Winkel, Temperatur, Stromstärke, Spannung) und es gibt eine Vielzahl an Control-Modi, die eine exakte Regelung von Position, Geschwindigkeit und Drehmoment ermöglichen.

Die Kommunikation zu den Motoren erfolgt über ein standardisiertes Bus-System. Dadurch können bis zu 256 Motoren über eine Strecke von bis zu einem Kilometer miteinander gekoppelt werden und von einem einzigen Host-Controller gesteuert werden, was insbesondere für Automatisierungs- und

Robotik-Projekte von großem Vorteil ist. Die Firmware ist in C/C++ geschrieben und wir verwenden ein open-source build/test system.

Ziel des Sensorimotor während der Förderungsphase war es, die fertige Hardware um notwendige Software zu ergänzen und somit für alle nutzbar zu machen. Mit einer Arduino-kompatiblen C/C++ Bibliothek und einer Raspberry Pi-kompatiblen Python-Bibliothek können nun alle Motoren im Bus für viele unterschiedliche Projekte komfortabel angesteuert werden.

Meilensteine des Projekts waren:

1. C/C++ Bibliothek inklusive Unittests, Beispielcode und Tutorials für Arduino

2. Python-Bibliothek inklusive Unittests, Beispielcode und Tutorials für Raspberry Pi

3. Erstellung einer umfassenden Dokumentation inkl. weiteren Anwendungsbeispielen.

Beitrag des Projektes zu den Zielen der Förderinitative „Software-Sprint“

Unsere Zielgruppe sind Wissenschaftler*innen, Ausbilder*innen, Maker und Entwickler*innen kleinerer Unternehmen, insbesondere in strukturell schwächeren Regionen. Durch free-design Hardware besteht die Möglichkeit zur Modifikation und Eigenproduktion und damit einhergehend die Förderung unabhängiger Entwicklungen. Durch ausführliche Dokumentation, Tutorials und Veröffentlichung aller Schaltpläne auf einer Platform wie GitHub, wird ein weltweiter Wissenaustausch ermöglicht.

Ausführliche Darstellung der Ergebnisse

Im geförderten Projektzeitraum wurden die folgenden Coding-Projekte umgesetzt:

Protokoll, C++ Library, Python-Library, Arduino-Library, Microcontroller-Firmware, Protokoll-Plugin für Logic Analyzer, Servo Platine.

#Protokoll

Ein offenes Protokoll ist die Voraussetzung dafür, dass das Projekt als offene Plattform verwendet werden kann. Die initiale Entscheidung fiel daher auf ein möglichst einfaches, binäres Protokoll um die Verarbeitung durch leistungsschwache Plattformen wie den ATmega328 zu ermöglichen. Die zweite Priorität war die Sicherung der übertragenen Daten vor Übertragungsfehlern, was die Nutzung von einfacher und kosteneffizienter Hardware erlaubt.

Dieses Ziele führten zu der Entwicklung eines simplen und robusten Protokolls mit Prüfsumme zur Fehlererkennung, somit wurden unsere Initiellen Anforderungen an diese Komponente erfüllt.

C++ library

Bei der C++ Library handelt es sich um die Referenzimplementation des Protokolls für PC. Sie wurde als library konzipiert um sie für viele Programmiersprachen und Entwicklungsumgebungen verwendbar zu machen.

Python library

Bei der Python-library handelt es sich um einen sog. "Wrapper" für die C++ library, das heißt bereits implementierte Funktionen werden für die Python-Programmiersprache bereitgestellt. Dies erlaubt das Verwenden in einsteigerfreundlichen Python-Projekten.

Arduino-Library

Die Arduino library ist eine Implementation des Protokolls für die Arduino Platform. Sie stellt keinen Wrapper der C++ Library dar, sondern ist eine komplett unabhängige Entwicklung mit Fokus auf Kompatibilität für die Arduino-Plattform. Um die Bibliothek auch für Beginner nutzbar zu machen, wurde Wert auf eine möglichst einfache API gelegt.

Microcontroller-Firmware

Die Firmware stellt das Gegenstück zu den Bibliotheken dar, und hat die Aufgabe zwischen dem Protokoll und Steuerbefehlen auf Hardwareebene zu übersetzen.

Protokoll-Plugin für Logic Analyzer

Das Protokoll-Plugin bietet die Möglichkeit für Entwickler die Kommunikation zwischen Bibliothek und Servoplatine auf Hardwareebene zu betrachten. Dieses Projekt soll es einfacher gestalten eigene Protokollerweiterungen zu entwicklen und zu testen.

Die gesetzten Meilensteine wurden erreicht: Die genannten Komponenten sind fertig gestellt. Das Implementieren von Projekten auf Basis unserer Plattform ist möglich, und wurde bereits parallel während der Entwicklungsphase getestet.

Das Arbeiten mit einer Kombination von sowohl Software, Hardware und Firmware ist eine äußerst interessante und erfüllende Erfahrung, von der wir hoffen sie mit vielen Nutzern der Plattform teilen zu können.

Gerade aus dem Zusammenspiel der verschiedenen Komponenten ergaben sich natürlich viele potentielle Fehlerquellen die für Probleme sorgten; Im Speziellen die Hardware verhielt sich zeitweise anders als erwartet, und führte somit zu Schwierigkeiten die bei z.B. beim Entwickeln einer Webanwendung oder Handy-App sicher nicht aufgetreten wären.

Diese Probleme wurden jedoch bereits in der Planungsphase vorhergesehen, und für ihre Lösung Zeit und Ressourcen eingeplant. Wir sind überzeugt davon das sich mittels der Integration von Soft- und Hardware viele neue Problembereiche lösen und revolutionieren lassen, und wünschen uns daher vom PrototypeFund und der OKF auch in Zukunft die Bereitschaft, auch Hardware-nahe Softwareprojekte zu unterstützen.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Wir wollen mit dem Sonsorimotor eine Alternative zu bestehenden hochpreisigen closed-source SmartServos, wie z.B. Robotis/Dynamixel, anbieten. Durch unsere freie Hardware wird es ermöglicht, diese Technologie zu studieren, zu modifizieren und eigene Ideen darauf aufzusetzen. Dies gibt insbesondere kleineren, nicht finanzstarken Projekten mehr Unabhängigkeit und Möglichkeiten, denn: free hardware enables everybody.

Da unsere Zielgruppe sind Wissenschaftler*innen, Ausbilder*innen, Maker und Entwickler*innen kleinerer Unternehmen sind, ergeben sich für diese, durch die Veröffentlichung unserer free-design Hardware und der dazugehörigen Software, die Möglichkeit zur Modifikation und Eigenproduktion und damit einhergehend die Förderung unabhängiger Entwicklungen. Durch ausführliche Dokumentation, Tutorials und Veröffentlichung aller Schaltpläne auf einer Platform wie GitHub, versuchen wir alle Interessierten einen möglichst einfachen Einstieg in eigene Projekte zu ermöglichen.

Mittlerweile gibt es eine ganze Reihe an Projekten, zum Teil von uns selbst angestoßen, aber wir haben auch komplett von uns unabhängigen Robotikprojekten mit der Verfügungstellung von Sensorimotor-Platinen und Software helfen können.

Die Arbeit am Sensorimotor hat uns nicht nur technisch weiter gebracht – mit den erzielten Ergebnissen ist der Grundstein für unser eigenes open-source Robotikprojekt gelegt - sondern uns auch als Team zusammen gebracht und gelehrt trotz aller Widrigkeiten zusammen zu arbeiten.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Das Projekt wurde mit dem Ziel begonnen eine vollständige neue Plattform für die Entwicklung von Mechatronikprojekten zu erschaffen und die bestehenden Plattformen zu ergänzen. Als Teil dieses Plans hatten wir uns vorgenommen mit evtl. vorhandener contingency, also Pufferzeit, nützliche Tools für Entwickler und Nutzer zu implementieren, z.B.:

* eine Mobile App für die Fernsteuerung von Servomotoren
* eine Grafische Anwendung zur Visualisierung von gemessenen Werten durch den Smartservo
* einen Bootloader, um das Updaten von Firmware zu vereinfachen

Jedes dieser Unterprojekte wurde begonnen, jedoch haben wir uns entschieden die Bibliotheken, Firmware und Dokumentation zu priorisieren. Daher waren die genannten optionalen Milestones zum Zeitpunkt der Abgabe noch nicht fertig.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Alle unsere Entwicklungen sind unter https://github.com/supremem achines zu finden.

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Wie bereits beschrieben ergaben sich aus dem Zusammenspiel der verschiedenen Komponenten zahlreiche Probleme - im Speziellen war das Verhalten der Hardware anders als erwartet und führte somit zu Schwierigkeiten.

Auch wenn diese Probleme zu erwarten waren und wir in der Planungsphase für ihre Lösung Zeit und Ressourcen eingeplant hatten, war unsere Schätzung etwas knapp bemessen. Um die Fertigstellung der wichtigsten Komponenten rechtzeitig zu schaffen, mussten wir die Arbeit an einigen optionalen Unterprojekten reduzieren, so dass diese zum Zeitpunkt der Abgabe noch nicht fertig sind.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Nein gab es nicht.