Karlsruher Institut für Technologie
Institut für Technik der Informationsverarbeitung

Dokumentation zum Projektpraktikum Informationstechnik

Gruppe:	Freitag 2, Gruppe 6
Gruppenmitglieder:	Nicholas-Philip Brandt, Marcel Vogel, Selina Eckel
Tutor:	Fabian Marc Lesniak
Abgabetermin:	23.01.2015
Semester:	WS2014/2015

Institutsleitung
Prof. Dr.-Ing. Dr. h. c. J. Becker
Prof. Dr.-Ing. E. Sax
Prof. Dr. rer. nat. W. Stork

KIT - Universität des Landes Baden Württemberg und nationales Forschungszentrum in der Helmholtz-Gemeinschaft

Inhaltsverzeichnis

Kapitel 1
Einleitung

Die praktisch orientierte Aufgabe im dritten Semester war das Projektpraktikum. Die Aufgabenstellen lautetet, die Hardwaresteuerung eines TivSeg, ähnlich einem Segway, zu programmieren. Hierbei wurde man in zufälligen Dreiergruppen eingeteilt.

1.1 Arbeitsumfeld

Als Arbeitsumfeld wählte man den webbasierten Hosting Dienst Github. Github ermöglicht eine unabhängige Arbeitsweise der Teammitglieder. Durch die Versionskontrolle wird sichergestellt, dass der Code stets konsistent bleibt. Durch die History ist der gesamte Verlauf des Projekts gut zu verfolgen und Backups sind jederzeit verfügbar.

1.2 Zielbestimmung

Laut Aufgabenstellung sollte bei dem Projektpraktikum gelernt werden, wie komplexe C++ Codeabschnitte geschrieben werden und wie eine hardwarenahe Programmierung realisiert. Des Weiteren sollte das Arbeiten mit einer integrierten Entwicklungsumgebung geübt werden. Das Herz des TivSegs, mit welchem die Sensorik und Aktorik gesteurt wird, besteht aus dem EI-Mikrocontroller Board, das bereits aus den Workshops der ersten beiden Semester bekannt ist. Darüber hinaus sollte auch das Testen der unabhängigen Module auf dem Board geübt werden.
Als Simulation eines Industrieprojekts gehörten die Komponenten Zeitplanung und Teamarbeit genauso zum Praktikum, wie das eigentliche Umsetzen des Quellcodes.

1.3 Einsatz

Das TivSeg soll in den Bereichen touristische Stadtführungen, Patrouillenfahrten bei Polizeibehörden sowie die Erleichterung für Lehrpersonals auf dem Weg zu Lehrveranstaltungen eingesetzt werden.

1.4 Zeitplanung

Beim ersten Treffen wurden die einzelnen Module auf die Gruppenmitglieder aufgeteilt. Marcel übernahm die Module ADC und ADC Sensor, Nicholas die Module PWM, Motor und GPIO und Selina den Timer sowie die Dokumentation. Das Testen der einzelnen Module auf dem EI-Board wurde von den jeweils Zuständigen übernommen.
Der Gesamttest wurde mit Hilfe der Simulation zusammen durchgeführt. Für die Programmierung der einzelnen Module wurden 4 Wochen eingeplant. Die Woche darauf wurden Probleme der einzelnen Gruppenmitglieder besprochen und mit Hilfe des Tutors besprochen. In dieser Woche wurden auch die Module den Einzeltests unterzogen. In Woche 6 wurde dann der erste Gesamttest gestartet und nach ein paar Anläufen funktionierte die Simulation. Abschließend wurde in Woche 7 die Dokumentation geschrieben und die einzelnen Module wurden von den jeweils Verantwortlichen nochmal den anderen Gruppenmitgliedern erläutert.

Kapitel 2
Konzeption

2.1 Regelalgorithmus

Der Regelalgorithmus wurde den Gruppen bereit gestellt.
Dieser Regelalgorithmus übernimmt die Aufgabe der Auswertung der Sensorwerte zur Berechnung der Motorgeschwindigkeit.

2.2 Benötigte Module und ihre Funktion

Die folgenden Module mussten programmiert werden:

- Timer
  Die Aufgabe des Timers besteht darin, periodisch nach einem festen Zeitintervall eine Interrruptroutine auszulösen.
  Intern zählt dieser bis zu einem bestimmten Grenzwert, bei dem der Interrupt ausgelöst wird. Das feste Zeitintervall ist nötig, da der Regelalgorithmus Messwerte zu bestimmten Zeiten erwartet.
  Als Zählmodus wurde eine Sägezahnkurve gewählt, der im Register "Wavesalemode"gesetzt wurde. Das Intervall des Timers wurde auf 10 ms gesetzt mittels variablen RC-Wert , der bei der Initialisierung berechnet wird.

- GPIO Sensor
  Der GPIO Sensor steuert einzelne Pins des Mikrocontroller Boards an oder liest deren binäre Werte aus.

- ADC
  Die ADC Klasse ist hauptsächlich dazu da, den ADC des Mikrokontrollers zu initialisieren, die Ausgangspins zu aktivieren, konkret Signale zu konvertieren und auszulesen.
  Durch setzten des ersten Bits im Status Register des ADC kann eine solche Konvertierung gestartet werden. Dies geschieht an einem Kanal, der zuvor über das Enable Register aktiviert wurde. Nach der Konvertierung kann der konvertierte Wert im Last Converted Data Register ausgelesen werden. Des weiteren kann in diesem Modul eingestellt werden, ob ein Pin durch den GPIO oder durch eine Peripheral Function kontrolliert wird.

- ADC Sensor
  Die ADCSensor Klasse bildet einen Wrapper für einen bestimmten ADC.
  Sie initialisiert einen ADC und leitet get -Aufrufe anschließend an den ADC weiter. Bei der Initialisierung werden Slope Factor und Offset des ADC gesetzt.

- PWM
  Das PWM Modul steuert die Geschwindigkeit der Motoren.
  PWM ist eine wichtige Steuerungsweise, bei der Leistung über das Ein/Aus-Verhältnis einer binären Spannung auf einem Zeitintervall geregelt wird. Je großer der Ein-Anteil, desto großer die Leistung und damit die Geschwindigkeit.
  Die PWM wird mit Hilfe eines Zählers umgesetzt. Liegt der Zähler über dem Duty Cycle so liegt hohe Spannung an, sonst null. Erreicht der Zähler die Period wird er wieder zurück gesetzt.

- Motor
  Die Motor Klasse funktioniert analog zur ADCSensor Klasse als Wrapper für eine PWM. Dabei werden Duty Cycle und Period gesetzt. Das Verhältnis Duty Cycle zu Period entspricht dem Verhältnis Ein- zu Intervallzeit.

2.3 Beschreibung der Schnittstellen

- Fußschalter: Gibt die Steuerung des TivSegs frei und schützt den Anwender vor eventuellen Schäden.
- Lenkstange: Dient zur Richtungssteuerung. Das TivSeg fährt in die Richtung, in die die Stange geneigt wird.

Kapitel 3
Realisierung

3.1 Hardware Abstraction Layer

Zu Deutsch Hardwareabstraktionsschicht oder kurz HAL ist eine Schicht des Betriebssystems, die als Einzige auf die Hardware zugreifen kann. Sie bietet abstraktere Befehlsstrukturen und vereinfacht dadurch die Programmierung.
Die restliche Software nutzt die HAL, um Befehle umzusetzen.

3.1.1 Implementierung

Die Beschreibung der einzelnen Funktionen in den jeweiligen Modulen sind im Anhang "My Project", erzeugt durch Doxygen, in dieser Dokumentation zu finden.

3.1.2 Zusammenfügen der Module

Die Stärke von Github liegt in der Versionskontrolle, die wir nutzten um die Module zusammenzufügen. Dabei wurden die unabhängig voneinander bearbeiteten Äste des Projekts zusammengefügt (gemergt) und auftretende Konflikte manuell gelöst.

Kapitel 4
Test

4.1 Test der HAL

Die beiden Module PWM und Timer konnten auf dem EI-Board getestet werden. Beim PWM wurde die Funktion getestet, indem die LEDs in unterschiedlichen Helligkeitsstufen leuchten sollten. Beim Timer wurde die erfolgreiche Initialisierung durch eine LED angezeigt und das Hochzählen durch eine blinkende LED realisiert. Bei den Tests traten hauptsächlich Schreibfehler und Unaufmerksamkeiten auf, die meist bei der Kompilierung festgestellt wurden.

4.2 Abschlusstest

Da die Simulation nur funktioniert, wenn alle Module funktionsfähig sind, bildet die Simulation gleichzeitig auch den Logiktest.
Nach einigen Versuchen im Simulator funktionierte die abschließende Version des Quellcodes in Woche sechs konform mit der Musterausgabe.

Kapitel 5
Zusammenfassung und Ausblick

5.1 Zusammenfassung

Zusammenfassend kann man sagen, dass das TivSeg nun für den täglichen Gebrauch verwendet werden kann.
Nach einer Einarbeitungszeit und guter Zusammenarbeit zwischen den Teammitglieder konnte die Aufgabenstellung gelöst werden. Der Umgang mit dem Datenblatt wurde immer vertrauter und man lernte wie sich die Hexadezimal-Adressen der einzelnen Komponenten zusammensetzen.
Das Konzept der hardwarenahen Programmierung mit Register und Bits wurde erarbeitet und vertieft.
Erfreulich ist die Praxisnähe des Projekts, da man das eigene Resultat auch selbst am entwickelten Produkt ausprobieren darf, was bestimmt mit jeder Menge Spaß verbunden ist :P.

5.2 Ausblick

Die Fortbewegung als Grundfunktion des TivSegs funktioniert nun. Für die unterschiedlichen Einsatzgebiete könnten dies weitere Erweiterungsmöglichkeiten sein:

- Touristenbranchen
  Bei Stadtführungen sind Fotos schöne Erinnerungen. Damit man nicht immer absteigen muss, wenn ein schönes Motiv festgehalten werden soll, könnte man beispielsweise eine GoPro(TM) (Actionkamera) anbringen und diese mit einem Fernauslöser, angebracht an den Haltegriffen, bestücken. So könnte schnell und ohne absteigen Schnappschüsse geschossen werden und kein Motiv geht mehr verloren.

- Polizei
  Für Patrouillenfahrten sollte das TivSeg auch geländetauglich sein. Hierfür müssten Reifen mit tieferen Profilen bereitgestellt werden und eine Assistent zur Unterstützung der Gewichtsausgleichung entwickelt werden.

- Unigelände
  Eines der wichtigsten Komponenten im Uni-Alltag ist der Kaffee. Da wäre es doch schade, wenn der Kaffee noch schnell getrunken werden muss, damit man noch pünktlich mit dem TivSeg in der nächsten Vorlesung erscheint. Da wäre eine Kaffeehalterung doch ideal. Mit einem Sensor könnte die Füllhöhe bestimmt werden und somit berechnet werden, in welchem Winkel maximal das TivSeg gebeugt werden kann, damit der Kaffee nicht ausgeschüttet wird.

Kapitel 6
Anhang