

PPM - Zeitdokumentation

Mattias Abfalterer

8. April 2015

08.10.2014

- Beschäftigen mit GitHub
- Pflichtenheft besprechen

15.10.2014

- Recherche LeapMotion
- Bestellung Bluetooth-Modul
- Fertigstellung Pflichtenheft
- Beschäftigen mit GitHub-Banches und mergen

22.10.2014

- LaTeX funktionsfähig machen
- Recherche Kommunikation zwischen Laptop und Arduino
- Organisatorisches im Team

29.10.2014

- Bestellung benötigter Hardware
- Programm Bluetooth Laptop-Arduino
- Hardware auseinanderbauen (zu Hause)
- Bestellung zusätzlicher Hardware (zu Hause)

29.10.2014

- Konzept Aussehen des Roboters mit Erweiterung

05.11.2014

- Teambesprechung
- Recherche serielle Kommunikation für C#

12.11.2014

- Bluetooth Kommunikation zwischen Laptop und Arduino testen
- C# Programm für serielle Kommunikation

19.11.2014

- Code für Bluetooth in C# - Programm, das die Daten von LeapMotion entgegennimmt, empfangen
- Ausführliche Testung der seriellen Kommunikation

Noch zu tun:

- Arduino-Programm, das die Daten interpretiert und den Motoren übergibt
- Hardware zusammenbauen und testen
- eventuell noch Zeit freihalten für Troubleshooting

25.11.2014 - Zuhause

- Provisorischer Zusammenbau der Hardware

26.11.2014

- Troubleshooting für Programm
- Besprechung der möglichen Fehler

03.12.2014

Tätigkeit

- Troubleshooting für Programm
- Fehler: Einlesen der Daten die vom PC kommen

Meilensteine

- Eingelesene Daten richtig spalten (10.12.2014)
- Bewegungsdaten verwerten (17.12.2014)

10.12.2014

Tätigkeit

- String richtig aufgespaltet und in Integer gewandelt
- Bewegungsdaten verwertet

Meilensteine

- Roboter löten (17.12.2014)
- Testen (17.12.2014)

17.12.2014

Tätigkeit

- Roboter gelötet
- Roboter testen
- Problem: keine LeapMotion

Meilensteine

- Test mit LeapMotion (07.01.2015)

07.01.2015

Tätigkeit

- Löten von Roboter wegen Beschädigung bei Transport
- Test mit LeapMotion
- Problem: Roboter interpretiert Daten falsch, obwohl richtig eingelesen

Meilensteine

- Roboter korrekt ansteuern (14.01.2015)

14.01.2015

Tätigkeit

- Löten von Roboter wegen Beschädigung bei Transport
- Arduino-Programm so verändert, dass Roboter Daten richtig verwertet
- Testung - Funktioniert!

Meilensteine

- Proportional zur Neigung der Hand die Geschwindigkeit des Roboters ändern (28.01.2015)

28.01.2015

Tätigkeit

- Geschwindigkeit des Roboters ist proportional zur Neigung der Hand, beim Lenken ist der Kurvenradius proportional zur Neigung
- Testung - Funktioniert!

Meilensteine

- Infrarot-Distanzmesser einbauen (04.02.2015)

04.02.2015

Tätigkeit

- Infrarot-Sensor eingebaut
- Recherche Funktionsweise

Meilensteine

- IR-Sensor richtig ansprechen - Sicherheitsabstand von Wand (18.02.2015)

18.02.2015

Tätigkeit

- Infrarot-Sensor programmiert - bleibt vor einer Wand stehen

Meilensteine

- Umdrehroutine mit ins Programm einbauen (25.02.2015)

25.02.2015

Tätigkeit

- Infrarot-Sensor programmiert - dreht vor einer Wand automatisch um

Meilensteine

- Dokumentieren (04.03.2015)

04.03.2015

Tätigkeit

- Dokumentation

Meilensteine

- Präsentation vorbereiten (bis 08.04.2015)

18.03.2015

Tätigkeit

- Dokumentation
- Testung Gesamtprogramm

Meilensteine

- Präsentation vorbereiten (bis 08.04.2015)

25.03.2015

Tätigkeit

- Dokumentation
- Vorbereitung Präsentation

Meilensteine

- Präsentation vorbereiten (bis 08.04.2015)

08.04.2015

Tätigkeit

- Dokumentation
- Vorbereitung Präsentation

Tätigkeitsbericht Simon Bodner

LeapMotionRobotTeam

8.10.2014

- Überarbeitung des Pflichtenheftes mit Alexander Hoppichler
- Festlegung der Arbeitsaufteilung (mein Teil: Programmierung der LeapMotion)
- Entwurf des Codes für die Datenübermittlung zwischen LeapMotion (bzw. PC) und Roboter (bzw. Arduino) – Code siehe Pflichtenheft
- Recherche zur Verwendung und Programmierung der LeapMotion

15.10.2014

- Überarbeitung Pflichtenheft (hauptsächlich Beschreibung des verwendeten Codes für die Übermittlung von Daten)
- Einblick in GitHub
- Recherche zur Verwendung und Programmierung der LeapMotion

22.10.2014

- Einlesen in Programmierung der Leap-Motion
- Verfassen von Kurzanleitung zur Benützung von GitHub

29.10.2014

- Beispielprogramm von der LeapMotion-SDK erzeugt, Libraries hinzugefügt
- Programmierung von grundlegenden Funktionen der LeapMotion

03.11.2014

- Erstellung eines Beispielprogramms zum Einlesen von Handposition (ohne Test)

05.11.2014

- Definition von maximalen und minimalen Neigungen der Hand
- Festlegen eines sinnvollen Toleranzbereiches um den Nullpunkt (Handposition gerade)
- Programmierung: Einlesen von benötigten Winkeldaten von LeapMotion, Beginn Programmierung der Methode für die Erzeugung des zu sendenden Codes.

06.11.2014

- Programmierung der Methode für das Erzeugen des zu sendenden Codes. Die von der LeapMotion eingelesenen Werte müssen mit einem Toleranzbereich um den Nullpunkt versehen werden und auf einen Prozentwert (ganzzahlig) in ihrem Gültigkeitsbereich (z.B. Toleranzbereich 10, maximale Neigung der Hand 50 → Gültigkeitsbereich/Prozentbereich zwischen 10 und 50) berechnet werden.
Noch Fehler in Berechnung.

12.11.2014

- Erzeugen des zu sendenden Codes über C#-Programm funktioniert
- Fehlersuche (Fehler war in Berechnung des Codes. Werte von LeapMotion eingelesen: Basis ist für Vor und Zurück, sowie Links und Rechts 100 (0: minimal bzw. Links, 200: maximal bzw. Rechts))

19.11.2014

- Simulation von Testdaten (da uns LeapMotion dieses Mal nicht zur Verfügung stand)
- Hilfe und Test beim Erstellen des Arduinoprogramms mit der gesamten Gruppe für die Entgegennahme und Interpretation der Werte
- Test von Zusammenspiel von C#- und Arduino-Programm – erfolgreich. Ausgabe der Motoren (noch nicht mit wirklichen Motoren) in TeraTerm.

26.11.2014

- Troubleshooting bzgl. Senden der Daten von LeapMotion zu Arduino
- Fehler bei der Übertragung, allgemeine Fehler

3.12.2014

- Troubleshooting bzgl. Senden der Daten von LeapMotion zu Arduino. Arduino empfängt gesendete Daten nicht korrekt. Anscheinend liegt dies am Delay im Arduino- bzw. Thread.Sleep im C#-Programm.
Meilensteine für nächste Einheit: Korrektes Aufspalten der Daten am Arduino

10.12.2014

- Erfolg beim Aufspalten des gesendeten Codes am Arduino:
Code wird als String gesendet und über die C-Methode strtok aufgespalten. Anschließend werden die beiden Array-Einträge mithilfe von atof von String in int umgewandelt. Nun

liegen am Arduino die beiden notwendigen Werte wie gewünscht vor. Es folgt die Interpretation.

Bsp.:

Input:	A100.0B100.0	A100.0 bedeutet nach vorne fahren (100 ist Grenze – 99 wäre bereits rückwärts)
Motor vor		B100.0 bedeutet geradeaus fahren (keine Lenkung)
Input:	A177.0B18.0	A177.0 → nach vorne fahren
Motor vor		B18.0 → nach links lenken
Motor left		(linker Motor aus, rechter Motor ein)
Input:	A57.0B144.0	
Motor zuruck		A57.0 → rückwärts fahren
Motor rechts		B144.0 → nach rechts lenken
Input:	A42.0B107.0	
Motor zuruck		A42.0 → rückwärts fahren
Motor rechts		B107.0 → nach rechts lenken

- Löten von manchen Stellen des Roboters

17.12.2014

- Erstellen eines Testlaufs über den C#-Code (vor, lenken, zurück, lenken).

Nächster Meilenstein: Fertigstellung der Hardware (kalte Lötstellen, brüchige Verbindungen etc), Testläufe

7.1.2015

- Löten einer Bruchstelle
- Test mit LeapMotion, Problem: Daten kommen am Arduino richtig an, Roboter interpretiert sie aber falsch (statt zurück z.B. stopp)

Nächster Meilenstein: Roboter soll korrekt nach vorne und hinten fahren

14.1.2015

- Ansteuerung des Roboters – es funktioniert: nach vorne, hinten Fahren sowie Lenken. Sollkriterium ist erreicht (fahren mit konstanter Geschwindigkeit).

Weitere Schritte: Geschwindigkeit des Roboters soll von Neigung der Hand vorgegeben werden können. – bis 28.1.2015

21.01.2015

Fehlersuche bzgl. heute nicht funktionierender Bluetooth-Verbindung – am Ende funktionierte wieder alles.

28.1.2015

Es funktioniert mittlerweile, den Roboter mit verschiedenen Geschwindigkeiten zu steuern – je nach Lage der Hand. Auch Lenken funktioniert abhängig von der Handschräge unterschiedlich schnell. Damit ist das erste Kannkriterium bereits erreicht.

Weitere Vorgehensweise:

Umsetzung der zwei weiteren Kannkriterien (Abstandsmessung über Infrarotsensor und Stehenbleiben bevor Auffahren auf ein Hindernis bzw. Leuchten einer Warn-Led bei geringem Abstand zu einem Hindernis.

Nächster Meilenstein nächste Woche:

Aufrüsten des Roboters um einen Infrarotsensor

4.2.2015

Roboter aufrüsten um Infrarotsensor und Led (Blinken wenn Abstand zu Hindernis zu klein)

In C#-Programm werden dem Benutzer nun alle verfügbaren COM-Ports aufgelistet, er kann einen davon auswählen.

Meilenstein nächste Woche: Test, ob Funktionalität mit Infrarotsensor (Stop und Drehung bei zu wenig Abstand) funktioniert

25.02.2015

Verfassen der Dokumentation über das C#-Programm und den erstellten Code.
Nächste Woche: Weiterführen der Dokumentation

4.3.2015

Beginn mit Erstellen der Präsentation

18.3.2015

Dokumentation und Erstellen der Präsentation

25.3.2015

Erstellen der Präsentation

8.4.2015

- Vorführen des Roboters
- Erstellen Präsentation
- Erstellen Dokumentation

Tätigkeitsberichte Hoppichler

Tätigkeitsbericht 8.10.2014

Anfangs wurde eine kleine Besprechung durchgeführt, in welcher die Arbeitseinteilung festgelegt wurde.

Ebenfalls wurde in dieser Einheit von mir gemeinsam mit Simon das Pflichtenheft überarbeitet. Im Anschluss recherchierte ich über die Programmierung des Arduinos und der LeapMotion, gemeinsam mit Simon.

Tätigkeitsbericht 15.10.2014

In der heutigen Einheit wurde von mir mit dem Zeichenprogramm Dia das Schema des Datenaustausches gezeichnet. Anschließend wurden bestimmt, wie die Daten zwischen den Devices ausgetauscht und weiterverarbeitet werden.

Als nächstes recherchierte ich im Internet, wie die LeapMotion ihre Daten als Signal ausgibt/weitergibt. Das Internet bietet in dieser Hinsicht nicht viele Informationen.

Erneute Überarbeitung des Pflichtenheftes gemeinsam mit dem Team.

Verwendung von Github mit dem restlichen Team wurde besprochen und wir versuchten, aufgetretene Probleme gemeinsam zu lösen.

Tätigkeitsbericht 22.10.2014

Kleine Recherche, wie der Arduino für unser Vorhaben programmiert werden könnte.

Den Code, der über das Bluetooth Signal an den Arduino übermittelt wird, wurde nochmal besprochen und verschiedenste Beispiele wurden durchbesprochen.

Anschließend habe ich ein Programm für einen Arduino erstellt, welches einen zufälligen Code erzeugt. Dieser generierte Code wird dann am SerialMonitor ausgegeben.

Tätigkeitsbericht 29.10.2014

Anfangs habe ich das Programm, welches am 22.10. erstellt wurde (ZufallsCodeGenerator) kommentiert sodass es wartungsfreundlicher wird.

Dokumentation, wie unser Code ausschauen soll, was in der Vorwoche nur verbal besprochen wurde habe ich schriftlich festgehalten.

Beispiel vom 22.10. wurde erweitert, sodass ausgegeben wird, wohin gelenkt wird und mit wie viel % der Geschwindigkeit gefahren werden soll. Anschließend wurde das Programm zur Erleichterung der Wartung kommentiert (noch nicht fertig).

Tätigkeitsbericht 05.11.2014

Fertig kommentieren des Arduinoprogrammes vom 29.10.

Erstellen eines Testprogrammes das am Roboter ausprobiert werden soll.

Tätigkeitsbericht 12.11.2014

Programmierung des Arduinos: Fertigstellung des Testprogrammes vom 05.11.2014.

Troubleshooting, da ein Fehler aufgetreten ist.

Fehler: Arduino gibt nicht das Signal aus, was verlangt wird.

Wird z.B. eine Zahl zwischen 0 und 100 eingelesen, funktioniert alles und der Linke Motor bekommt ein Signal (HIGH), und der Rechte Motor bekommt ebenfalls ein Signal (LOW)

Wird anschließend eine Zahl zwischen 101 und 200 eingelesen, bekommt der Rechte Motor zwar das richtige Signal (HIGH), doch der Linke Motor bleibt auf HIGH, dies sollte nicht sein!

Fehler wurde noch nicht entdeckt.

Tätigkeitsbericht 19.11.2014

Fehlerbehebung: Fehler vom 12.11.2014 wurde behoben

Fertigstellung des Programmes, welches die Motoren des Roboters ansteuern

Zusammenfügen der Programme mit den Teammitgliedern

„Trockentestlauf“ wurde anschließend erfolgreich durchgeführt.

Tätigkeitsbericht 26.11.2014

Es wurden die erstellten Programme getestet.

Es wurde herausgefunden, dass das C# - Programm einwandfrei funktioniert, doch beim Arduinoprogramm zum Einlesen der Werte via Bluetooth gibt es Fehler.

Werte werden nicht korrekt eingelesen.

Fehler wurde gesucht aber noch nicht gefunden.

Tätigkeitsbericht 03.12.2014

Erneutes Troubleshooting, da das Programm immer noch fehlerhaft ist.

Fehler beim Einlesen wurden behoben, dies funktioniert nun.

Bei der Weiterverarbeitung der eingelesenen Daten treten immer noch Probleme auf, welche in den nächsten Einheiten schnellstmöglich behoben werden müssen.

Meilensteine (gleich wie bei Abfalterer/Bodner)

Bis 10.12.2014: String (Zeichenkette) richtig aufteilen

Bis 17.12.2014: Bewegungsdaten richtig verwerten.

Tätigkeitsbericht 10.12.2014

Gefehlt krankheitsbedingt

Tätigkeitsbericht 17.12.2014

Zusammenbau der Hardware (löten, verkabeln, etc.).

Durchführung weiterer Tests, diesmal schon mit Hardware.

Leider wurde die LeapMotion vergessen.

Meilensteine

Bis 14.01.2015: Durchführung diverser Tests und Testläufe mit der LeapMotion

Tätigkeitsbericht 07.01.2015

Lötarbeiten am Roboter wegen Beschädigung.

Testlauf mit LeapMotion.

Problem: Daten werden falsch interpretiert, obwohl diese richtig eingelesen werden.

Meilensteine

Bis 14.01.2015: Roboter soll richtig vorwärts bzw. rückwärts fahren

Bis 21.01.2015: Restliche Daten richtig verwerten

Tätigkeitsbericht 14.01.2015

Weitere Lötarbeiten am Roboter wegen erneuter Beschädigung.

Roboterprogramm für die Steuerung wurde überarbeitet und vor- und rückwärtsfahren wurde getestet - funktioniert einwandfrei.

Das Programm zum Lenken wurde auch überarbeitet und getestet – funktioniert auch.

Roboter fährt durch Steuerung der LeapMotion!

Überlegung: Wir wollen die Daten, die wir über die LeapMotion bekommen, als Geschwindigkeitswert verwenden – erste Überlegungen, wie wir das machen sollen.

Meilensteine

Bis 28.01.2015: Bewegungsdaten der LeapMotion zum Vorwärtsfahren als Geschwindigkeitswert verwenden (proportional zur Neigung der Hand).

Tätigkeitsbericht 21.01.2015

Leider konnte ich den Unterricht nicht besuchen.

Tätigkeitsbericht 28.01.2015

Leider konnte ich den Unterricht nicht besuchen.

Tätigkeitsbericht 04.02.2015

Erstellung der Struktur unserer Dokumentation.

Ausarbeitung der Dokumentation + Hilfeleistung für Mattias.

Tätigkeitsbericht 18.02.2015

Weiterschreiben der Dokumentation

Tätigkeitsbericht 25.02.2015

Weiterschreiben der Dokumentation

Finaler Testlauf mit erweitertem Programm (Infrarotdistanzmesser)

Meilensteine

Bis 04.03.2015: Dokumentation erweitern

Tätigkeitsbericht 04.03.2015

Projektdokumentation:

Weiterschreiben an der Projektdokumentation

Präsentation mit Bodner begonnen zu entwerfen

Meilensteine

Bis 18.03.2015: Doku fertigstellen, Präsentation erweitern

Tätigkeitsbericht 11.03.2015

Schitag

Tätigkeitsbericht 18.03.2015

Projektdokumentation:

Weiterschreiben der Projektdokumentation

Projektpräsentation weiter entworfen

Meilensteine

Bis 08.04.2015: Fertigstellung der Projektpräsentation

Tätigkeitsbericht 25.03.2015

Projektpräsentation weiter entworfen

Projektdokumentation:

Weiterführung der Projektdokumentation (Probleme während der
Projektentwicklung/Projektentstehung)

Meilensteine:

Bis 08.04.2015: Fertigstellung der Projektpräsentation

Tätigkeitsbericht 08.04.2015

Fertigstellung der Projektdokumentation / Projektbericht.

Fertigstellung der Projektpräsentation.

Vorführung des Projektes und „Notendiskussion“ mit Lehrer