

2016

Projektdoku

RUBIKS CUBE ROBOTER
STOOP ADRIAN

Inhalt

Requirements	2
Terminplan	2
Codeaufbau	3
Tracking	3
22.03.2016.....	3
08.04.2016.....	3
15.04.2016.....	4
22.04.2016.....	4
29.04.2016.....	4
06.05.2016.....	5
13.05.2016.....	5
20.05.2016.....	6
27.05.2016.....	7
03.06.2016.....	7
10.06.2016.....	7
17.06.2016.....	7
Überlegungen	7
Messungen/Tests	9
Vergleich mit Requirements.....	9
Messungen	10
Soll/ist Zeit vergleich	10
Reflexion/Erkenntnisse	10
Was ging gut	10
Was ging schlecht	11
Was kann ich daraus lernen	11
Summary	11

Requirements

In this project I will build a robot that which can solve a Rubik's cube.

The necessary software and hardware components have already been compiled by myself. I will use the Lego Mindstorm kit to drive the Lego motors and sensors.

I can initialize the colors of the cube with the sensor.

The program algorithms can be switched before the program starts.

I will program the x and o's algorithms.

My goal is to have the robot solve the cube in under 60 seconds.

The robot will be powered by an external power supply.

The robot must be small enough to fit into my school bag.

Terminplan

	Projektplan																	
Aufgaben	04.03.2016	11.03.2016	18.03.2016	25.03.2016	01.04.2016	08.04.2016	15.04.2016	22.04.2016	29.04.2016	06.05.2016	13.05.2016	20.05.2016	27.05.2016	03.06.2016	10.06.2016	17.06.2016		
Requirement schreiben	x	x																
Projektplan					x													
Einkauf der Bauteie			x	x														
Erster Roboterprototyp						x	x											
Code für ansteuerung der Motoren								x										
evt Prototyp verbessern								x										
Code für Farbensensor/tests									x									
Roboter komplett zusammenbauen									x	x								
Code für x and o's algorithmus						x	x				x	x	x	x				
Vortrag																		
Projektdoku					x			x		x		x		x		x		
Produktdoku					x			x		x		x		x		x		
Optimierung Code(vgeschwindigkeit)														x	x	x		
Gliederung Texte überarbeiten															x	x		
Soll Stunden(ohne Schulzeit):	0.25	0.25	2	0.5	5	20	20	20	20	25	20	30	20	20	25	20	248	Total
Ist Stunden(ohne Schulzeit):	0.5	0	3	1	4.5	25	20	23	21	30	25	24	25	30	35	40	307	Erledigt
Differenz:	0.25	-0.3	1	0.5	-0.5	5	0	3	1	5	5	-6	5	10	10	20	59	Zu tun

Codeaufbau

Da ich denke, dass die Beschreibung des Codes sehr viel wird habe ich mich dafür entschieden dies in einem extra File abzulegen.

Wie der Algorithmus funktioniert und den Würfel löst wird im File: RubiksCubezüge.docx

Finanzen

Ich habe mir ein Budget von 100.- vorgestellt. Mein Plan ist es, die Teile die ich einkaufe nach dem Projekt wieder zu verkaufen. Leider kann ich so nicht bestimmen ob ich das Budget übertraf oder einhielt vor dem Ende des Projekts. Ich habe ca. 800.- für alles ausgegeben und erhoffe mir auf Ricardo oder Ebay diesen Wert wieder zu bekommen.

Finanzen							
Datum	Wo	Artikel Nr.	Was	Wieviel	Stk Preis	Total Preiss	Link
22.03.2016	shop.lego.com	45502	Großer EV3 Servomotor	6	39.90	239.40	http://shop.lego.com/de-CH/Gro%C3%9Fer-EV3-Servomotor-45502
22.03.2016	shop.lego.com	45500	Intelligenter EV3-Stein	2	249.90	499.80	http://shop.lego.com/de-CH/Intelligenter-EV3-Stein-45500
22.03.2016	shop.lego.com	45514	EV3 Kabel-Paket	2	19.90	39.80	http://shop.lego.com/de-CH/EV3-Kabel-Paket-45514
22.03.2016	shop.lego.com	45506	EV3-farbensensor	1	44.90	44.90	http://shop.lego.com/de-CH/EV3-Farbsensor-45506
						823.90	

*Die falschen Adapter konnte ich zurückschicken und bekam mein Geld wieder.

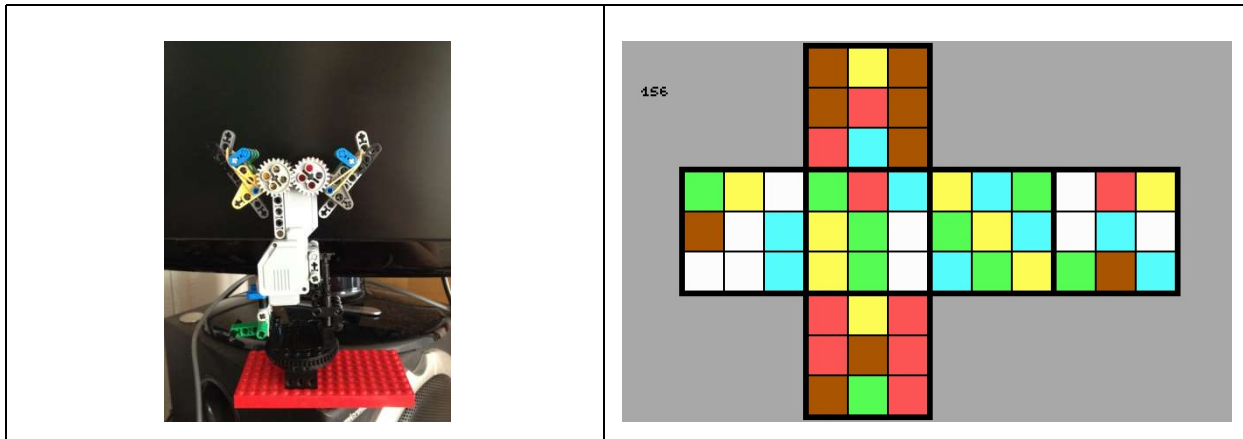
Tracking

22.03.2016

Heute habe ich die Bauteile bestellt und bekam 15min später eine E-Mail, dass dieser Betrag nicht möglich ist für Vorkasse/Rechnung. Ich solle mit VISA/Master Card bezahlen oder den Betrag vorher Überweisen. Nach ca. 1 Stunde hab ich dann die Überweisung machen können mit hilfe meines Vatters. Durch das ist die Ist-Zeit etwas grösser geworden. Zeitverbesserung ist hier nicht möglich >> keine Massnahmen treffen.

08.04.2016

Nach Stundenlangem zusammenbauen ist der erste Prototyp Arm des Roboters fertig. Zudem hab ich mein Operating System schon so umgerüstet, dass es meine Roboter Bewegungen grafisch darstellen kann. So kann ich mein Code später gut verbessern. Ich habe Variablen die mitzählen wie viele Züge der Roboter machen muss und wie viel speicheraustausche er gemacht hat. Zudem noch wie viele vergleiche(if abfragen) er machen wird.



15.04.2016

Die Grafische anzeige meines OS ist nun fertig und ich kann die genaue Züge nachschauen und Daten auslesen. Auslesen kann ich:

R_Züge = wie viele Züge das Programm machen muss um den Würfel zu lösen.

R_Züge_real = wie viele Züge der Roboter machen muss, da ein Zug im Programm z.B. 3 Züge mit dem Roboter sein kann.

R_berechnung = wie viele Arrays er vertauschen musste um zu der Lösung zu kommen.

R_Lweg = Lösungsweg für den Roboter in einem Array abgespeichert.

22.04.2016

Ich kann jetzt die Motoren des Prototypen ansteuern und den Würfel in einer Halterung bewegen. Ich benutzte Eobotc als Compiler, da er alles vorbereitet um den Kernel neu zu Programmieren mit einem Framework.

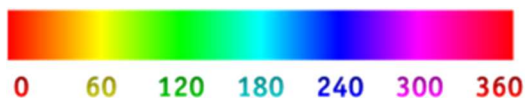
Nach langem probieren mit den Werten hab ich das Beste herausgefunden: der Arm muss sich 42 Grad drehen mit einer Geschwindigkeit von 45%. Danach hält der Arm den Würfel, weil er nur 41 Grad schafft. Danach kann ich den ganzen Arm drehen und wenn er an der Ziel Position ist, stoppe ich den ersten Motor und öffne ich wieder mit 41 Grad.

29.04.2016

Heute befasste ich mich mit dem Farbensensor. Der Sensor hat mehrere Möglichkeiten die Farben zu erkennen:

getColorAmbient = gib die Helligkeit des Objekts zurück (0-100), 0 eher hell und 100 eher dunkel. Dieser Modus ist unbrauchbar für meine Anwendung.

GetColorHue = gibt den Farbton zurück (0-360).



Diese Funktion ist sehr nützlich, da ich die Umgebungsbelichtung nicht beachten muss, wenn ich als Farbreferenz den mittleren stein nehme.

getColorName = gibt den Farben Name zurück. Folgende Farben sind möglich: schwarz, blau, grün, gelb, rot, weiss und braun.

Dieser Modus ist sehr einfach, doch leider wird die orange Fläche auf dem Würfel entweder als rot oder als gelb interpretiert. Daher ist dieser Modus auch nicht brauchbar.

Getcolorreflection = diese Funktion misst die zurückgeworfenen Wellen, fast wie getColorAmbient. Auch hier wird ein Wert von 0-100 als rückgabewert genutzt. Wie auch getColorAmbient ist diese Funktion nicht zu gebrauchen.

getColorRGW(r,g,b) = speichert in den 3 werten die rgb werte ab. Fast wie GetColorHue kann ich diese Funktion benutzen. Ich sehe aber keinen Grund 3 variablen zu benutzen wenn es mit einer geht.

getColorSaturation = gibt die Sättigung der Farbe zurück. Auch diese Funktion ist uninteressant für diese Arbeit.

06.05.2016

Ich habe über das Wochenende den Roboter neu zusammengebaut, weil ich mit dem Motor nach der Achse nicht zufrieden war. Der Prototyp war zwar fest angemacht, konnte sich aber mit der Zeit leicht lösen. Ich habe mir eine andere Ansteuerung ausgedacht wo der Motor zum festhalten des Würfel auch aussen sein kann:

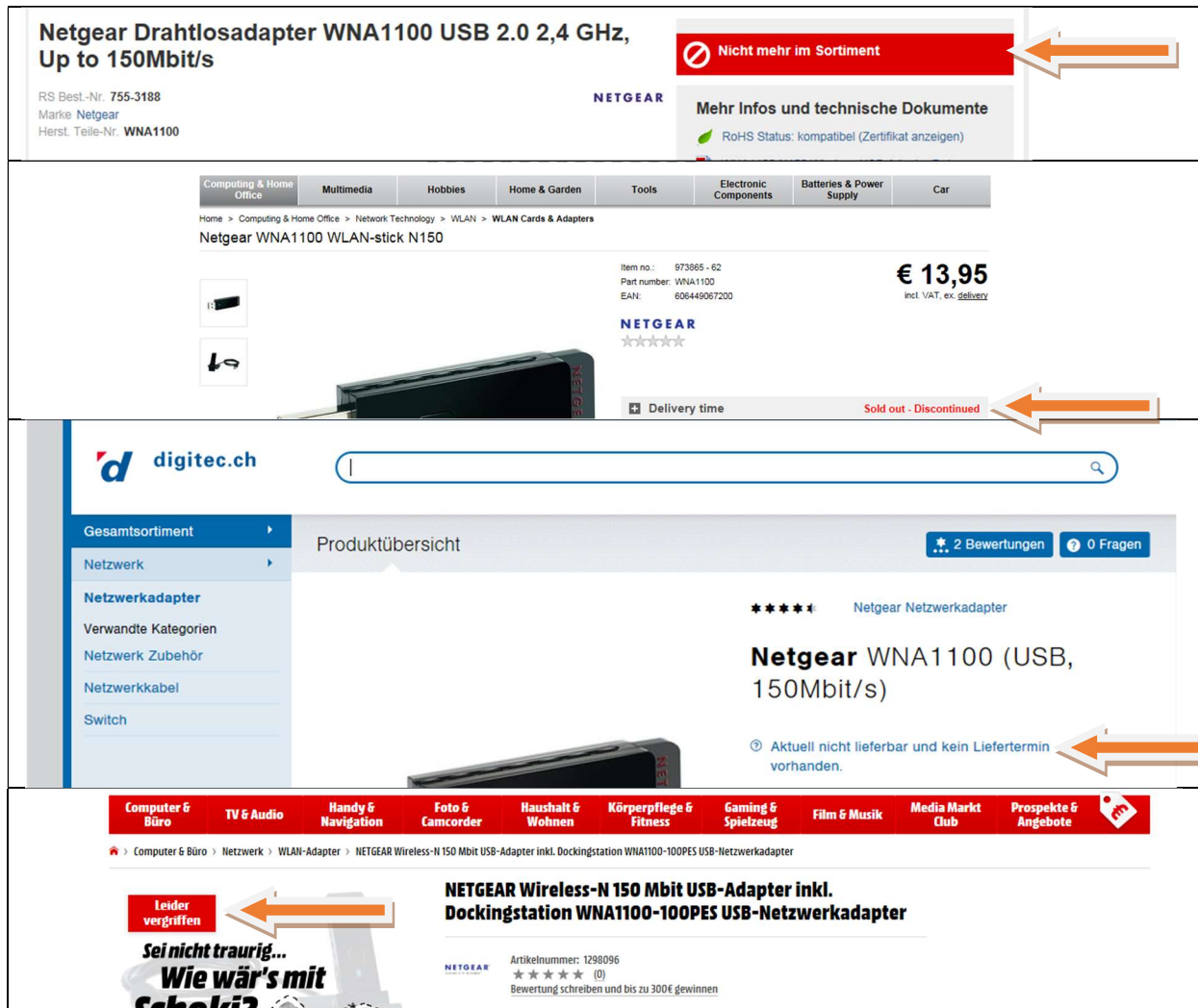
°#Bild fehlt@@@@@@@

13.05.2016

Ich arbeitete an meinem Code und gliederte ihn neu, damit er wieder übersichtlicher ist. Ich habe auch zwei Header File erstellt um gewisse Funktionen auszulagern wie z.B. die Grafikanzeige und die Bewegungszüge für das Array. Diese Funktionen sind sehr gross und erhöhen nur die unübersichtbarkeit im Code. Ich kaufte mir 2 Wlan Adapter, die ich brauche um meine Bricks miteinander kommunizieren zu lassen(laut Lego geht nur der WNA1100 von Netgear).

20.05.2016

Gestern kamen die Wlan Adapter an und ich muss leider feststellen, dass diese nicht gehen. Ich müsste den Netgear WNA1100 haben, doch Conrad schickte mir zwei WNA3100. Nach langer Internet suche musste ich feststellen, dass dieser Adapter nicht mehr zumindest in der Schweiz erhältlich ist:



Netgear Drahtlosadapter WNA1100 USB 2.0 2,4 GHz, Up to 150Mbit/s

RS Best.-Nr. 755-3188
Marke Netgear
Herst. Teile-Nr. WNA1100

NETGEAR

Nicht mehr im Sortiment

Mehr Infos und technische Dokumente
RoHS Status: kompatibel (Zertifikat anzeigen)

Computing & Home Office | Multimedia | Hobbies | Home & Garden | Tools | Electronic Components | Batteries & Power Supply | Car

Home > Computing & Home Office > Network Technology > WLAN > WLAN Cards & Adapters
Netgear WNA1100 WLAN-stick N150

Item no.: 973865 - 62
Part number: WNA1100
EAN: 606449067200

€ 13,95
incl. VAT, ex. delivery

NETGEAR
★★★★★

Delivery time **Sold out - Discontinued**

digitec.ch

Gesamtsortiment | Produktübersicht | 2 Bewertungen | 0 Fragen

Netzwerk
Netzwerkadapter
Verwandte Kategorien
Netzwerk Zubehör
Netzwerk kabel
Switch

Netgear WNA1100 (USB, 150Mbit/s)

Aktuell nicht lieferbar und kein Liefertermin vorhanden.

Computer & Büro | TV & Audio | Handy & Navigation | Foto & Camcorder | Haushalt & Wohnen | Körperpflege & Fitness | Gaming & Spielzeug | Film & Musik | Media Markt Club | Prospekte & Angebote

Computer & Büro > Netzwerk > WLAN-Adapter > NETGEAR Wireless-N 150 Mbit USB-Adapter inkl. Dockingstation WNA1100-100PES USB-Netzwerkadapter

Leider vergriffen

NETGEAR Wireless-N 150 Mbit USB-Adapter inkl. Dockingstation WNA1100-100PES USB-Netzwerkadapter

Artikelnummer: 1298096
★★★★★ (0)
Bewertung schreiben und bis zu 300€ gewinnen

Sei nicht traurig... Wie wär's mit Schoki?

Mit dieser Information muss ich leider feststellen, dass ich mein geplanter Roboter nicht ansteuern kann. Darum werde ich mich auf eine Andere Bauform anpassen bei der man nur einen Ev3 Stein braucht.

27.05.2016

Ich habe mir eine Anleitung für den neuen Roboter aus dem Internet heruntergeladen, damit ich sicher sein kann, dass er funktioniert. Dieser Roboter ist bekannt unter dem Namen Mindcube3r.

<http://mindcuber.com/mindcub3r/mindcub3r.html>

03.06.2016

Der Roboter ist fast fertig, einige Bauteile besitze ich leider nicht. Ich musste einige Änderungen vornehmen aber ich denke dies wird kein Problem sein. Ich überlegte mir ob ich irgendwie den 4ten Anschluss für die Motoren noch irgendwie benutzen könnte für eine weitere Ansteuerung. Ich muss aber feststellen, dass dies nicht so Effizient ist wie ich das wollte.

10.06.2016

Der Roboter ist fertig und ich probiere schon das Scannen des Würfels. Es ist schwierig dies zu testen, da ich alle Farben irgendwie auf dem LCD-Display anzeigen muss oder mit dem Debugger nachschauen.

17.06.2016

Ich bin mit dem Roboter fertig, muss aber feststellen, dass im Code noch paar Bugs vorhanden sind. Ich hoffe ich kann diese noch finden und verbessern. Ich arbeite nun nur noch an der Projektdokumentation und am Code. Der Roboter ist fertig.

Überlegungen

Am Anfang des Projektes wollte ich meinen Roboter Komplet selber Zusammenbauen. Mit eigenem uC und Motoren von Maxonmotor. Per Zufall sah ich auf YouTube wie andere solche Roboter mit Lego Mindstorm und Lego Technik Bausteine bauten. Nach kurzem recherchieren fand ich heraus, dass man diese Mindstorm Motoren mit C,C# oder java ansteuern kann. Daraufhin hab ich mich entschieden das Projekt mit Lego Mindstorm Motoren und Bausteinen zu bauen. Von der Kindheit hab ich zuhause noch sehr viel Lego Bausteine.

Nachdem ich viele Videos gesehen habe konnte ich mir selbst ein Bild machen wie ich den Roboter am besten baue. Ich werde Links und rechts und unten einen Arm bauen der sich drehen kann und nach dem Würfel greifen kann. Der Arm am Boden wird noch mit einem Farbensensor ausgestattet.

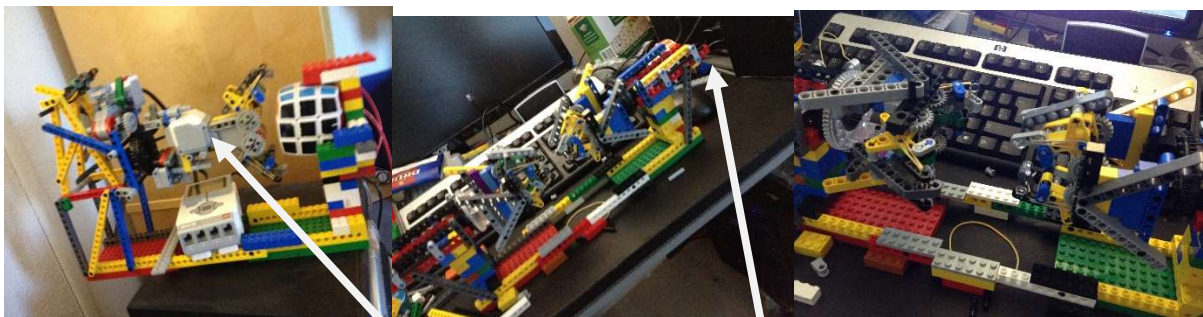
Da ich später, bei dem verbessern des Codes, wissen möchte ob er besser wurde erstelle ich mir 10 Muster Stellungen des Rubiks Cube als Arrayvorlage (oben rechts beim 08.04.2016 sieht man ein solches Muster grafisch Dargestellt). Somit kann ich softwaremässig immer die gleiche Ausgangssituation testen und trotzdem immer ein anderen Code benutzen.

Um den Roboter zu bewegen in der richtigen Art (Winkel, Geschwindigkeit) musste ich mich sehr langsam an die besten Werte ran tasten.

Um in jeder Situation die richtige Farbe zu bestimmen benutzte ich die Funktion GetColorHue. Wenn ich in einem dunklen Zimmer die rote Farbe messe, dann kommt z.B. der Wert 5, aber wenn ich es vor einem Fenster messe erscheint da eine 12. Um dieses Problem zu lösen speichere ich rot als wert nicht ab sondern messe die Mittleren steine um die 6 Farben werte abzuspeichern. Danach kann ich die anderen 8 steine den referenzsteinen zuordnen und kontrollieren ob von derer Farbe 9 steine da sind.

Irgendwie gefällt mir der Roboter noch nicht ganz und ich möchte ihn umbauen, damit der Motor des Arms für das Festhalten des Würfels auch aussen angebracht werden kann. Ich habe natürlich zuerst im internet nach ähnlichen Robotern gesucht, um mehrere Lösungen bereit zu haben. Ich habe mich für ein Zahnrad entschieden das um eine Achse gedreht werden kann aber durch diese Umdrehung das Festhalten kaum beeinflusst wird:

Alt(1) im Vergleich zu neu(2-3):



Man sieht auf dem Bild wie der Motor an einer neuen Stelle angebracht werde kann.

Ich habe festgestellt, dass Robotc keine Kommunikation zwischen zwei Ev3 Steinen zulässt. Ich muss mir ein anderer Compiler suchen, der dies unterstützt. Auf der Seite Monobrick.dk hab ich eine Anleitung gefunden die mir weiter helfen kann. Dort steht wie ich mit Wireshark und einem Wlan Adapter mit den zwei Ev3 Steinen kommunizieren kann über TCP/IP mit C++. Ob ich den Code in C oder C++ schreibe habe ich nicht definiert. Dies ist mir egal, denn ich kann beide dieser Sprachen.

Ich musste leider Feststellen, dass die Wlan Adapter nicht gehen, da es das falsche Model ist. Nach langer Suche stellte ich fest, dass der Adapter den ich brauche nicht erhältlich ist und wenn, dann nur einen auf Amazone(schon gebraucht). Ich werde jetzt einfach den Roboter klein halten und nur mit einem Ev3 Stein ansteuern. Da ich aber den anderen Roboter auch irgendwann testen möchte verbaue ich nur neue Teile, damit der alte noch ganz bleibt.

Im Internet suchte ich nach Bauarten des Roboters mit nur einem Ev3 Stein um mir ein paar Ideen zu sammeln. Auf einer Seite habe ich dann eine Anleitung gefunden, die ich zum Teil benutzen kann. Gewisse Teile besitze ich nämlich nicht, die dort verbaut werden.

Während dem Programmieren und Compiler ist mir aufgefallen, dass mein OS die 32kbyte Grenze Erreicht hat. D.h. ich kann mit dem Aktuellen Bootloader nicht den ganzen Code in den Arbeitsspeicher laden. Um dies zu ändern musste ich den Bootloader komplett umschreiben um mehr Code vom Laufwerk zu laden. Eine Floppy besteht aus C(ylinder), H(ead) und S(ector). Mein alter Code startete bei CHS = 0,0,2 und gab 63 Sektoren als zu lesende "Blocks" (= Sektoren) ein, dann berechnet das BIOS selbst die korrekten C- und H-Werte. Das werde ich nun selbst übernehmen, damit ich hier weiter komme, denn die 63 Sektoren kommen von einer Beschränkung der

Eingabedurch nur 6 Bit. Um dies zu ändern muss ich die Register ch, cl und dh ändern bevor ein weiterer Sektor eingelesen werden kann. Der Linkscript wurde natürlich auch angepasst.

Mein neues Bios kann nun über 1MB einlesen ohne das der Stack oder Speicher beschädigt wird oder überladen.

Ich habe einige Teile des neuen Roboters nicht und muss so mit einer Zange und etwas Heissleim nachhelfen. Als ich heute meine Code für den Roboter ausprobieren wollte viel mir auf, dass mein Trial Benutzer abgelaufen ist und ich nichts mehr Compilen oder Downloaden kann. Nach schnellem Aufsetzen einer Virtuellen Maschine mit Win7/32bit und Internen USB „Verlegung“ konnte ich nach 2 Stunden wieder meine 10 Tage gratis Version „testen“.

Um exakte ungelöste Würfel zu bekommen benutzte ich die Seite <https://rubiks-cube-solver.com/>. Dorf gibt es eine Ansicht so wie mein Array und es ist sehr einfach dies zu kopieren.

Der Würfel hab ich heute noch mit einem Speziellen Silikonspray eingesprüht. Mit dem gleiten die Kanten viel besser und ein wenig WD40 schadet ja auch nie ☺

Messungen/Tests

Vergleich mit Requirements

Da ich nicht meinen Roboter benutzten kann, sondern einen fremden jetzt habe, kann ich nicht alle Requirements erfüllen.

1. Der Roboter kann den Würfel Lösen.
2. Der Roboter besteht aus Lego und die SW wurde von mir geschrieben.
3. Die Farben können eingelesen werden mit dem Farben Sensor.
4. Es kann leider nur ein Algorithmus benutzt werden.
5. Der Algorithmus heisst x and o's.
6. Der Roboter kann den Würfel nicht in einer Minute lösen, da der Code für zwei arme geschrieben wurde. Der jetzige Code benutzt sehr viel seiner Schwachstellen >> über die Mitten drehen(M,E,S);
7. Der Roboter wird mit Batterien betrieben.
8. Ich kann beide Roboter in eine Tasche Transportieren.

Ich bin zufrieden mit dem was ich geleistet habe und sehe hier auch nichts, dass ich verbessern könnte. Ich habe zwar nicht alle requirements erfüllt aber zumindest einen Ansatz davon.

Messungen

Hier eine Tabelle von den 10 Mustern die gelöst wurden.

Muster:	R_Züge	R_berechnung	R_Wechsel	R_Lösungsweg	R_RealLw
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					

Soll/ist Zeit vergleich

Am Anfang des Projektes und in der Mitte klappte es sehr gut mit dem Terminplan, manchmal paar Stunden zu viel und mal paar zu Wenig. Am Ende, wo ich merkte, dass ich einen Zweitet Roboter bauen muss, ignorierte ich den Terminplan, da ich zu diesem Zeitpunkt nicht daran ändern hätte konnte. In den Letzten 2-3 Wochen Arbeitete ich mehr als erwartet und geplant. Für das Nächste Projekt werde ich mich mehr über die Bauteile und uC informieren und nicht einfach auf Videos vertrauen.

Ich denke ich kann sehr gut einen Terminplan planen und Einhalten aber wenn etwas nicht so läuft wie ich es geplant habe, habe ich sehr wenig Freiraum im Rest des Projektes. Nächstes mal Plane ich noch Reservezeit ein damit so was nicht mehr vorkommt.

Reflexion/Erkenntnisse

Was ging gut

Ich wusste von Anfang an wie ich den Code ungefähr schreiben möchte, den man sollte vor dem coden wissen wie man es machen will. Ich bestellte alle Teile direkt am Anfang des Projektes und musste so nicht irgendwo eine Pause einlegen. Der erste Prototyp des Roboters fand ich schon so gut, dass ich ihn direkt übernommen habe für den richtigen Roboter. Ich hatte nie wirklich einen Zeit Stress, da ich immer den Überblick hatte über mein Projekt. Ich wusste wo ich nachfragen musste wenn mir was nicht klar war mit der LIB von Robotc. Ich habe gemerkt, dass mein Räumliches Vorstellungsvermögen sehr wichtig war und ich es bestimmt stärkte in dieser Zeit. Ich schrieb mir eine Hilfe mit der ich den Code schreibe (RubiksCubezüge.docx). Als ich das Summery geschrieben habe, korrigierte ein Kollege von mir dies Direkt. Über das ganze Projekt habe ich viel mehr gelernt als ich erwartet habe.

Was ging schlecht

Wenn ich code was nicht stimmte musste ich immer mit Bildschirmausgaben den Fehler suchen, den das Programm war in einem Grafikmodus. Durch das verlor ich viel Zeit beim suchen aber verstand den Code somit immer besser und konnte ihn wenn nötig noch verbessern. Das Compiler Programm hatte nur eine 10 Tages Lizenz, die ich total vergessen habe, dies kostete mich Zeit. Ich habe mich zu wenig über die Verbindungen des Steins informiert und hätte früher feststellen müssen, dass man zwei Steine nicht miteinander reden lassen konnte. Ich benutzte zu Hause einen 64-bit Rechner und ich musste den Code immer auf einer Virtuellen Maschine Compilieren, da es 16-bit Code darunter hat. Ein 64-bit Rechner kann kein 16-bit Code generieren, da die Register min. 32 bit gross sind. Über Dropbox wurden dann immer die Files getauscht.

Was kann ich daraus lernen

Ich werde nächstes Mal mich am Anfang des Projektes über alles besser Informieren. Wenn ich Meilensteine habe, sollte ich mich an dehnen richten und jede Woche nachschauen ob ich noch im Terminplan bin. Ich werde Reserve zeit einplanen, damit ich nicht den ganzen Tag daran arbeiten muss wenn ich hinter dem Zeitplan bin. Nächstes Mal werde ich in einer Gruppe Arbeiten um auch mal diese Erfahrungen zu Sammeln.

Summary

The motivation of the developer was to implement his graduation project from 9th grade. He built a Rubik's Cube solving robot and coded it.

Product requirements / specification of the product

The functional requirement is 1 Lego Ev3 brick, 3 large Ev3 motors, 1 Ev3 color sensor, Lego building blocks, Ev3 wire, sw. robot, source code. These items cost around 400 Fr.- The developer needed five to seven hours to build the final robot and about 200 hours to finalize the code.

Result / product

The robot can figure out the colors of the cube and solve it. The algorithms name is "x and o's". The robot can be taken along in a school bag and is powered by batteries.

Not all requirements are fulfilled. Another robot with 2 bricks was planned but the developer can't connect this two bricks. Bluetooth, wlan, usb are not supported by any compiler at this time. The final robot cannot solve the cube under a minute. He has only one arm instead of two which was planned. Many cube stones are moved over the middle plane, and that's his weak point.

Project planning: resources, time schedule, risk management

The resources were good distributed. The budget is 100.- and the expected costs are 750.-
The project has two phases:

#1: build robot

#2 Program the code.

Many solution paths can be simulated in an external program coded by the developer. So thousands cube settings will be solved in minutes and error messages are outputted directly in the console. For the construction the developer has no instructions.

The developer sees no future problems in this project.

Performance

In contrast to the scheduled process the measured progress exceeded the deadline. The amount of code increased and the overview of the code got better than in the last project. The developer worked every week on the project and in the finalization nearly every day. The developer built a second robot. The second robot was built according to instructions from the internet. The source code needed to be converted to fit the new definitive situation where just one robot was used. The total costs are 823,90.- and the budget was 100.-. The developer resells the parts as the project is done. He hopes the budget will not exceed after getting the resale paid back.

The project plan has been followed until to the point where the second robot was built.

Retrospective / conclusion, evaluating the project

The project was completed successfully. The developer has learned more than he expected to do but many more hours were needed. Next time he will be better informed about the components because the older brick just supported data transfer between two NTX bricks. The Lego products 31313 are needed to build the robot by the internet instruction. Robotc, the compiler which was used, is free software.