

Manual Cube_Solver_Robot

Inhalt

Hardware	2
Würfel tauschen	2
Reset	2
Zustands Automat	3
Allgemeine Funktionen und Modi	4
Funktionen im <i>Automatic</i> -Modus	4
Random moves	4
Change Cube	4
Scan Cube	4
Solve	4
Fast/Slow	4
Funktionen im <i>external_PC</i> -Modus	4
Random moves	4
Change Cube	5
Scan Cube	5
Receive Cube from Robot	5
Show Cube	5
Python-Skript	5
Solve	5
Darstellung des Würfels für den <i>external_PC</i> -Modus	6
Darstellung der Drehungen für den <i>external_PC</i> -Modus	7
Software	7

Hardware

- Sechs Schrittmotoren im Closed-Loop System: Nema 17HS15-1504-ME1K
- Sechs Closed-Loop Schrittmotortreiber: CL42T-V41
- Servo Motor: RS2
- FRDM-MCXA153 Development Board (MCUXpresso IDE)
- Farbsensor: TCS34725 RGB
- Würfel: QiYi M Pro 3x3 MagLev Ball-Core UV Coated (Achtung nicht jeder Würfel ist geeignet)
- OLED-Bildschirm: SSD1309
- 5 Taster
- Netzteil 24V: S-250-25
- Stepdown-Wandler: LM2596 (für Lüfter)
- Stepdown-Wandler: ICQUANZX DC DC 9A 300W (für 5V Netz)
- 4 Lüfter: SUN MF80252V3-1
- 48 Neodym 5x5x5mm
- Selbstentwickelte Platine
- USB-C Buchse und Konverter: CP2102
- Kaltgeräte Buchse und Kabel: 693-4302.2001

Würfel tauschen

Der Würfel ist über Magnete mit den Motoren verbunden. Dafür muss dieser jedoch leicht modifiziert werden, dafür wurden an jeder Seite des Würfels vier kleine Magnete verklebt. Dabei ist die Ausrichtung der Magnete zu beachten. Sie sind alle in einem Nord-Süd Wechsel angebracht. Dies ist ebenfalls beim Einbauen des Würfels zu beachten und es müssen **ggf. Motoren um 90° gedreht werden**, damit die Magnete des Würfels und der Motoren passend übereinstimmen.

ACHTUNG:

Beim Einsetzen MUSS die Gelbe Fläche nach oben und die Rote Fläche nach vorne zeigen!

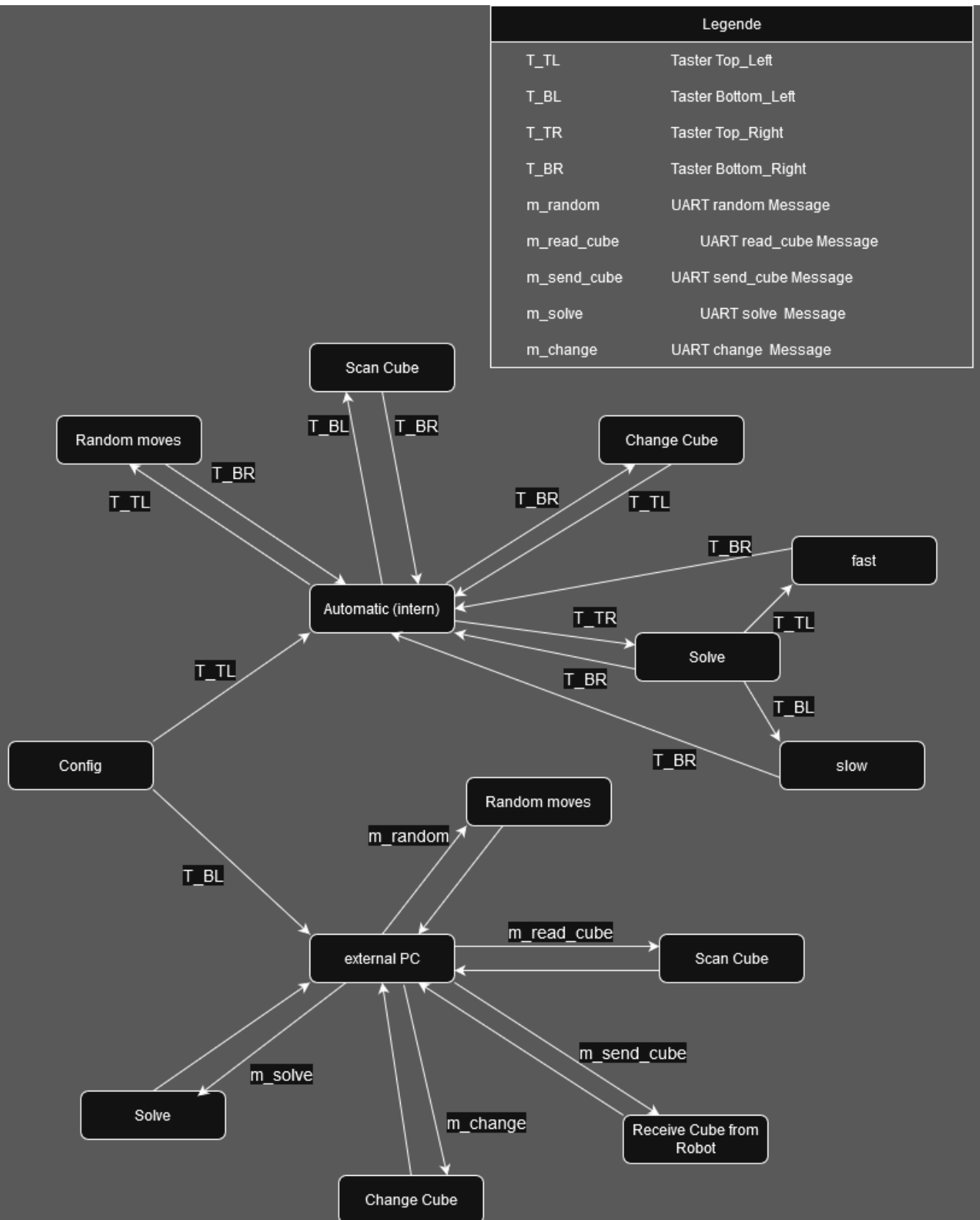
Der Würfel ist frei entnehmbar und sollte entweder **vor dem Anschalten** des Roboters eingesetzt werden oder in dem dafür vorgesehenen **Modus: „Change Cube“**.

Wichtig ist, dass sich der Roboter nach dem Einschalten und in dem oben genannten Modus kalibriert. Daher ist es wichtig, dass der Würfel annähernd passend und nicht verkanntet eingesetzt wird.

Reset

Falls sich der Roboter aufhängen sollte gibt es auch einen Reset Knopf, welcher mit einem passenden Stift betätigt werden kann. Auch hier wird sich der Roboter neu kalibrieren. Außerdem wird der Controller zurückgesetzt und verliert jegliche gespeicherte Informationen.

Zustands Automat



Allgemeine Funktionen und Modi

Der Roboter verfügt über zwei Hauptmodi: ***Automatic*** und ***external_PC***

Die Funktionen des Roboters sind dabei sehr ähnlich, jedoch ändert sich die Bedienung. In dem *Automatic*-Modus wird der Roboter über vier Taster und den Bildschirm gesteuert. Bei dem *external_PC*-Modus läuft diese Bedienung über die USB-Schnittstelle. Diese Auswahl muss nach dem Anschalten des Roboters über Taster ausgewählt werden. Für den *external_PC*-Modus existiert im ***/src/external_Computer*** Ordner eine Datei für die Kommunikation.

Funktionen im *Automatic*-Modus

Random moves

In dieser Funktion verdreht der Roboter zufällig 20 Seiten nacheinander. Der „Zufall“ wird hier über eine interne rand() Funktion gelöst, welchen den aktuellen Systemtick des Controllers als Seed nutzt.

Change Cube

Hier werden alle Motoren deaktiviert und der Würfel kann entnommen und/oder eingesetzt werden.

Nach dem Tauschen **muss ein Taster auf dem Roboter gedrückt werden**, um den Wechsel zu bestätigen. Im Folgenden kalibrieren sich alle Seiten wieder nacheinander und der Roboter ist erneut einsatzbereit.

Scan Cube

In dieser Funktion wird der Würfel eingelesen. Dafür wird jede Fläche nacheinander von einem Farbsensor gelesen dann interpretiert und abgespeichert. Um alle Flächen auf die untere Seite zu bekommen wird eine bestimmte Zugabfolge genutzt, welche den Würfel nicht verändert. Somit ist am Ende nach ca. 35 Sekunden der gesamte Würfel eingelesen und wurde dennoch nicht verändert.

Die einzelnen Steine werden nach einem bestimmten System in einem Array abgespeichert, welches später genauer erläutert wird.

Solve

Hier löst ein von uns selbst geschriebener Algorithmus den eingelesenen Würfel und speichert die benötigten Züge in ein Array auf dem Controller ab. Dieser Algorithmus ist auf der GitHub Seite des Repo verlinkt.

Fast/Slow

In diesen Funktionen wird das in *Solve* berechnete Lösungsarray mit der entsprechenden Geschwindigkeit ausgeführt. Nach dem Erfolgreichen Lösen des Würfels werden die benötigten Züge und Zeit auf dem Bildschirm angezeigt.

Funktionen im *external_PC*-Modus

Random moves

In dieser Funktion verdreht der Roboter zufällig 20 Seiten nacheinander. Der „Zufall“ wird hier über eine interne rand() Funktion gelöst, welchen den aktuellen Systemtick des Controllers als Seed nutzt.

Change Cube

Hier werden alle Motoren deaktiviert und der Würfel kann entnommen und/oder eingesetzt werden.

Nach dem Tauschen **muss ein Taster auf dem Roboter gedrückt werden**, um den Wechsel zu bestätigen. Im Folgenden kalibrieren sich alle Seiten wieder nacheinander und der Roboter ist wieder einsatzbereit.

Scan Cube

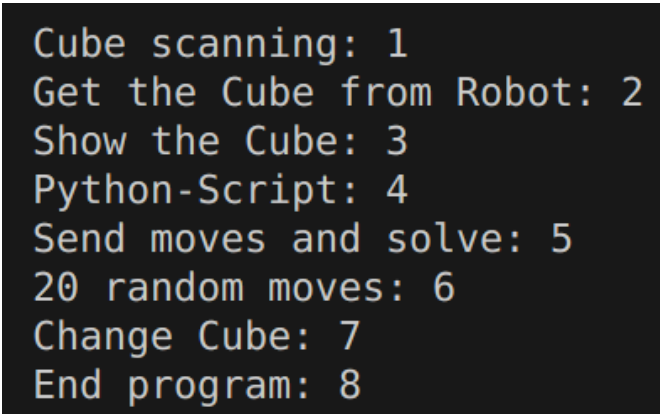
In dieser Funktion wird der Würfel eingelesen. Dafür wird jede Fläche nacheinander von einem Farbsensor gelesen dann interpretiert und abgespeichert. Um alle Flächen auf die untere Seite zu bekommen wird eine bestimmte Zugabfolge genutzt, welche den Würfel nicht verändert. Somit ist am Ende nach ca. 35 Sekunden der gesamte Würfel eingelesen und wurde dennoch nicht verändert.

Die einzelnen Steine werden nach einem bestimmten System in einem Array abgespeichert, welches später genauer erläutert wird.

Receive Cube from Robot

Hier wird der eingelesene Würfel vom Roboter über die USB-Schnittstelle an den externen PC gesendet und dort abgespeichert.

Wichtig: In den externen Bedienoptionen gibt es noch zwei weitere Funktionen, welche nicht auf dem Bild des Zustands Automaten oben aufgeführt sind.



```
Cube scanning: 1
Get the Cube from Robot: 2
Show the Cube: 3
Python-Skript: 4
Send moves and solve: 5
20 random moves: 6
Change Cube: 7
End program: 8
```

Show Cube

In dieser Funktion wird der zuletzt Empfangene Würfel im Terminal ausgegeben.

Python-Skript

Hier wird vom C Programm über die Funktion `fork()` ein Python-Skript ausgeführt. Dieses berechnet dann die benötigten Züge, um den Würfel zu lösen. Die Kommunikation zwischen dem C Programm und dem Python Skript wird über *Inter-Prozess-Communication* (IPC) realisiert.

Passend dafür existiert im `/src/external_Computer` Ordner eine Vorlage in welcher nur noch der eigene Lösungsalgorithmus implementiert werden muss. In dieser Vorlage wird der Würfel aus dem C Programm eingelesen und am Ende das Lösungsarray mit maximal 1024 Züge zurück an das C Programm gesendet.

Solve

In dieser Funktion wird das aus dem Python-Skript berechnete Lösungsarray an den Controller geschickt und dort auch sofort ausgeführt. Am Ende werden die benötigten Züge und die Zeit auf dem Bildschirm angezeigt.

Darstellung des Würfels für den *external_PC*-Modus

Jede Farbe ist einer Zahl zugeordnet:

Farbe	Nummer
Weiß	1
Gelb	2
Rot	3
Grün	4
Orange	5
Blau	6

Das eingelesene Array besteht aus 54 Zahlen und jede repräsentiert eine farbige Seite eines Steins auf dem Würfel. Das Array besteht so theoretisch aus 6 Blöcken, welche jeweils eine Fläche darstellen. Jeder dieser Blöcke beinhaltet 9 Zahlen. Die Reihenfolge der eingelesenen Blöcke ist:

Weiß -> Rot -> Blau-> Orange -> Grün -> Gelb

Jeder Stein auf dieser Fläche wird dabei nach dem folgenden Muster eingelesen:

									<table><tr><td>1</td><td>2</td><td>3</td></tr><tr><td>4</td><td>5</td><td>6</td></tr><tr><td>7</td><td>8</td><td>9</td></tr></table>			1	2	3	4	5	6	7	8	9
1	2	3																		
4	5	6																		
7	8	9																		
1	2	3	1	2	3	1	2	3	1	2	3									
4	5	6	4	5	6	4	5	6	4	5	6									
7	8	9	7	8	9	7	8	9	7	8	9									
									<table><tr><td>1</td><td>2</td><td>3</td></tr><tr><td>4</td><td>5</td><td>6</td></tr><tr><td>7</td><td>8</td><td>9</td></tr></table>			1	2	3	4	5	6	7	8	9
1	2	3																		
4	5	6																		
7	8	9																		

Würfel = [

W1, W2, W3, W4, W5, W6, W7, W8, W9,

R1, R2, R3, R4, R5, R6, R7, R8, R9,

B1, B2, B3, B4, B5, B6, B7, B8, B9,

O1, O2, O3, O4, O5, O6, O7, O8, O9,

G1, G2, G3, G4, G5, G6, G7, G8, G9,

Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y8, Y9

]

Tipp: Die Zahlen an der Position 5 des Blockes repräsentieren die Farbe der Fläche, da sich die Mittelsteine nicht verschieben lassen!

Darstellung der Drehungen für den *external_PC*-Modus

Jede Drehung ist mit folgenden Zahlen codiert (Genormte Notation als Vergleich).

Rechts beschreibt die Drehung im Uhrzeigersinn und *Links* die gegen den Uhrzeigersinn.

Seite	Rechts		Links		Doppel		Notation
Weiß	1	D	7	D'	21	D2	Down
Gelb	2	U	8	U'	22	U2	Up
Rot	3	F	9	F'	23	F2	Front
Grün	4	R	10	R'	24	R2	Right
Orange	5	B	11	B'	25	B2	Back
Blau	6	L	12	L'	26	L2	Left

Das Ergebnisarray, welches an den Roboter geschickt wird, darf somit **nur Zahlen zwischen 1 und 12 oder 21 und 26** beinhalten. Außerdem ist die **Länge auf 1024 Züge begrenzt**.

Software

Für Änderungen in der Software ist der Code für den Mikrocontroller ebenfalls mit im Repo. Zu beachten sind hier die `#defines` in der `Cube_Project.h`, welche für den jeweils verwendeten Roboter einzustellen sind, da sich die beiden Modelle doch etwas unterscheiden.

Diese Unterschiede sind in der Kalibrierung der Achsen und des Farbsensors zu beachten und gegebenenfalls anzupassen.