

FAQ zum RoboLab

Inhaltsverzeichnis

1	Allgemeines zum Praktikum	2
1.1	Was ist das RoboLab?	2
1.2	Wie lange muss ich arbeiten?	2
1.3	Was hat es mit den Meetings auf sich?	2
1.4	Wirklich jeden Tag einen Vortrag?	2
1.5	Was macht dieser Tutor?	2
1.6	Wie wird das Praktikum ablaufen?	2
2	C, die API und der Roboter	3
2.1	In Welcher Sprach wird programmiert?	3
2.2	Mit was programmieren wir?	3
2.3	Das Setzen der Path Variable unter Windows geht nicht richtig! .	3
2.4	Woher wissen wir, wie wir den Roboter ansteuern?	3
2.5	Wie können wir den Roboter piepen lassen?	3
2.6	Unsere Sensoren liefern keine oder falsche Werte zurück!	3
2.7	Das Anzeigen auf dem Display funktioniert nicht!	4
2.8	Wir haben mehr als ein .c File. Jetzt geht nichts mehr!	4

1 Allgemeines zum Praktikum

1.1 Was ist das RoboLab?

Im RoboLab steht Euch ein Lego Mindstorms NXT Set zur Verfügung, mit dessen Hilfe Ihr einen Roboter realisieren sollt, der selbstständig ein Labyrinth erkundet und Gegenstände in diesem findet.

1.2 Wie lange muss ich arbeiten?

Anwesenheitspflicht gibt es nur zu den Meetings am Vormittag. Die Zeit dafür liegt meist zwischen 9:00 und 10:00 Uhr, kann aber mit Eurem Tutor abgesprochen werden. Zur Prüfung sollte mindestens einer aus der Gruppe vor Ort sein. Zu anderen Zeiten müsst Ihr nicht in der Fakultät oder der Uni sein.

1.3 Was hat es mit den Meetings auf sich?

Jeden Morgen gibt es ein sogenanntes SCRUM. Euer Tutor wird dabei der SCRUM Master sein, dem Ihr einen kurzen Vortrag über den aktuellen Stand, (un-)gelöste Probleme und Euren heutigen Vorhaben haltet.

1.4 Wirklich jeden Tag einen Vortrag?

Der Vortrag sollte nicht länger als zehn Minuten dauern. Des Weiteren benötigt Ihr nur vier Folien: ein Deckblatt (mit Namen und Kennzeichnung, wer diesen Vortrag hält) und eine für jeden Bereich des SCRUMs.

Im Laufe der zwei Wochen muss jeder mindestens einmal präsentiert haben.

1.5 Was macht dieser Tutor?

Er wird Euch die zwei Wochen lang betreuen. Das heißt, er beantwortet allgemeine Fragen zum Programmieren, zum Ablauf des Praktikums, zu den Roboterfunktionen und zum Einrichten der Entwicklungsumgebung und was sonst alles mit dem RoboLab zu tun hat. Er ist nicht dafür gedacht, dass er Memory Leaks aufspürt oder den Fehler in Eurem großen Haufen Code findet (außer ihr wisst ganz genau, was wo nicht funktioniert).

1.6 Wie wird das Praktikum ablaufen?

Am Montag um 9:00 Uhr gibt es einen Einführungsvortrag, der beschreibt was Euch erwartet. Außerdem werdet Ihr in Eure Gruppen aufgeteilt.

Danach geht es ans Einrichten der Entwicklertools, das Roboter bauen und einer Linie sollte bis zum Nachmittag auch schon gefolgt werden. Die folgenden Tage habt Ihr dann Zeit, um weiter am Roboter und dem Code zu arbeiten.

Am letzten Donnerstag geht Euer aktiver Teil zu Ende und am folgenden Freitag findet die Prüfung und der Wettbewerb statt.

2 C, die API und der Roboter

2.1 In Welcher Sprach wird programmiert?

Es wird nur in reinem C programmiert: keine Vektoren oder Klassen, keine externen Bibliotheken.

2.2 Mit was programmieren wir?

Prinzipiell könnt ihr vom Notepad bis Visual Studio 2013 Ultimate alles benutzen. Vom Lehrstuhl wird aber ein fertiges Eclipse bereit gestellt, mit dem sich die Tutoren auch auskennen. Zum Einrichten unter den üblichen Betriebssystemen haben wir euch Anleitungen bereit gestellt. Der Anleitung ist Schritt für Schritt genau Folge zu leisten, sonst kann es sehr schnell zu unnötigen Problemen kommen. Falls Ihr bei einem dieser Schritte nicht wisst, was Ihr machen sollt oder es so nicht funktioniert, dann fragt einen Tutor.

2.3 Das Setzen der Path Variable unter Windows geht nicht richtig!

Ruft den Umgebungsvariablen Dialog am besten über nachstehende Abfolge auf: Systemsteuerung → System → Erweiterte Systemeinstellungen → Umgebungsvariablen. Die Path Variable findet Ihr dann in dem unteren Bereich. Falls Ihr MinGW benutzt, achtet darauf, dass der Cygwin Pfad vor dem von MinGW steht.

2.4 Woher wissen wir, wie wir den Roboter ansteuern?

Es existiert eine komplette Auflistung der Methoden zur Ansteuerung. Es ist jeweils beschrieben, was diese Methode macht, welche Parameter sie erwartet und was zurück gegeben wird. Zu Beginn solltet Ihr Euch diese Auflistung gut anschauen, um eine Vorstellung zu bekommen, was Ihr eigentlich alles machen könnt.

2.5 Wie können wir den Roboter piepen lassen?

Auch hier lohnt wieder ein Blick in die API: *ecrobot_sound_tone(freq, ms, vol)* gibt einen Ton der Frequenz *freq* mit der Lautstärke *vol* für *ms* Millisekunden aus. Bitte achtet darauf, dass es nicht zu laut und viel wird. Es nervt nach einiger Zeit nicht nur die anderen Teams. Die Lautstärke ist auch am Brick einstellbar.

2.6 Unsere Sensoren liefern keine oder falsche Werte zurück!

Die Sensoren müssen vorher initialisiert werden. Das geschieht am besten in der *ecrobot_device_initialize()* Methode. Des Weiteren überprüft, ob Ihr auch am richtigen Port auslesen lasst.

2.7 Das Anzeigen auf dem Display funktioniert nicht!

Zur richtigen Ansteuerung des Displays ist mehr als eine Methode notwendig. Zunächst muss per *display_clear(0)* der aktuelle Inhalt gelöscht werden. Danach wird der Cursor des Displays über *display_goto_xy(0,0)* auf eine sinnvolle Position (hier: 0,0) gesetzt. Im Anschluss kann mit verschiedenen Methoden (z.B.: *display_unsigned(23, 2)*) an diese Stelle geschrieben werden. Wenn mehr als nur eine Zahl oder ein Wort ausgegeben werden möchten, dann empfiehlt es sich, den Cursor manuell auf die nächste Position zu setzen. Nachdem alles geschrieben wurde, ist *display_update()* nötig, um den Inhalt sichtbar zu machen. Zusammengefasst das Schreiben von 1337 in die Mitte des Displays:

```
display_clear(0);
display_goto_xy(3,4);
display_unsigned(1337,4);
display_update();
```

2.8 Wir haben mehr als ein .c File. Jetzt geht nichts mehr!

Im Makefile (Projektordner→Makefile) müssen neben der *main.c* die anderen .c Dateien eingetragen werden. Einfach per Leerzeichen trennen.

Beispiel:

```
TARGET_SOURCES = $(addprefix src/, \
main.c methoden.c \
struktur.c )
```