

Schritt für Schritt Einführung VEXcode IQ







Schritt 1 – Herunterladen & Installieren

Kostenfreier Download auf der Webseite von VEX Robotics:

https://www.vexrobotics.com/vexcode-download

Die Programmierumgebung ist für Microsoft Windows, Mac und als App verfügbar.







Schritt 2 – Aktuelle Firmware

Für eine einwandfreie Funktion sollte immer die aktuelle Firmware verwendet werden. Alle Details zur aktuellen Version und die Download-Funktion, für das zum Aufspielen der Firmware benötigte Hilfsprogramm VEXos, werden auf der Seite von VEX Robotics angezeigt:

https://www.vexrobotics.com/iq/products/vexos

Update your VEX IQ f	irmware in 3 easy steps!
VEXos is a robotics operating system that harnesses the flexibility and power of VEX hardware for the VEX Robotics, uses real-time processing for repeatable operation at the fastest possible speeds. The Mac OS X 10.10, 10.11 & 10.12.	e rigors of competition and the diverse needs of education. This operating system, written completely by e "VEXos Utility" program simplifies updating VEX IQ hardware, and is compatible with Windows 7-10, and
All of the VEX IQ Smart Devices (Robot Brain, Controller, Smart Motor, and sensors) contain their own features. The best way to ensure that your VEX IQ system is functioning properly is to keep the Robo	n internal processors and run special software. This software is what allows for advanced programming tt's software up to date.
For information on changes between software versions, visit the VEX IQ Firmware Changelog.	
Current Public Release: VEXos Version 2.1.5 Updated February 2020 See what's new in VEXos (click to expand)	
Download VEXos for Windows	Download VEXos for Mac OS X
Note: If you have already installed the VEXos Utility Program, plug all sensors and motors into the Robot Brain, and proceed to step one below.	Note: If you have already installed the VEXos Utility Program, plug all sensors and motors into the Robot Brain, and proceed to step one below.
A. Click on the download button above to download the installer.	A. Click on the download button below to download the installer.
B. If given the option, choose "Run". Otherwise, save the file and open it.	B. Mount the DMG by opening the downloaded file and following the on-screen instructions.
C. Follow the on-screen instructions to install the VEXos Utility.	

C. Open the DMG (found on your desktop) and drag the VEXos Utility to the Applications folder

insite >>> education



Schritt 2 – Aktuelle Firmware

Alle VEX IQ Smart Devices (Robot Brain, Fernsteuerung, Motoren und Sensoren) haben eigene Prozessoren und damit auch eigene Firmware. Diese Firmware kann nachträglich aktualisiert werden, um neue Funktionen bereitzustellen oder Probleme zu beheben.

WICHTIG: Wenn die Firmware des Robot Brains aktualisiert wurde, muss auch die Firmware aller Devices erneuert werden.

Firmware aktualisieren in 3 Schritten:



Alle Smart Devices mit dem Robot Brain verbinden, dann das Brain via USB an den Rechner anschließen.



Brain einschalten.



VEXos öffnen und den Install-Button drücken.





Schritt 3 – VEXcode IQ starten

Durch Anklicken des Symbols die Programmieroberfläche starten. Es erscheint folgender Startbildschirm:







Schritt 4 – Sprache wählen

Über Symbolleiste Sprache auswählen:







Schritt 5 – Funktionen & Hilfen

Über Symbolleiste Beispiele öffnen und auswählen:







Schritt 5 – Funktionen & Hilfen

Über Symbolleiste Tutorials öffnen und auswählen:









Schritt 5 – Funktionen & Hilfen

Über Symbolleiste Hilfe öffnen und gewünschten Befehlsblock einfügen:









Schritt 6 – Programm neu starten oder öffnen

Über Symbolleiste neues Block- bzw. Textprogramm starten oder öffnen:







Schritt 7 – Geräteeinstellungen

Über Symbolleiste Gerätemanager öffnen. Je nach Aufbau des Roboters können nun die verwendeten Geräte (Fernsteuerung, Motoren, Sensoren) per Klick hinzugefügt und der entsprechende Port am Robot Brain ausgewählt werden. Für jedes Gerät kann der Name, bei manchen auch die Einstellungen angepasst werden.









Schritt 8 – Programm erstellen

Ein Programm startet immer mit dem Befehlsblock "wenn gestartet". Dieser befindet sich schon auf der Programmieroberfläche, die anderen Befehlsblöcke werden aus der Toolbar per Drag & Drop auf die Fläche gezogen. Nachdem ein Programm gestartet wird, werden die Befehle der Reihe nach auf dem Robot Brain abgearbeitet. Die Toolbar zeigt nur Befehle an, welche durch die Geräteeinstellungen des Roboters möglich sind.

Beispiel: Der Robot soll 200mm vorwärts fahren und sich anschließend um 90° nach rechts drehen:



12



Schritt 9 – Programm speichern und starten

Zum Speichern auf der Symbolleiste in das Feld *"VEXcode Project"* klicken. Programm benennen und an ausgewähltem Ort speichern. Das Robot Brain via USB mit dem Rechner verbinden. Sind Rechner und Brain verbunden, leuchtet das Symbol Brain grün. Einen der 4 Speicherplätze (Slots) wählen und das Programm auf das Brain herunterladen. Das Programm kann über das Feld *"Ausführen"* oder direkt auf dem Brain gestartet werden.







Schritt 10 – Programm in Text-Code konvertieren

Über Symbolleiste Programm als Text-Code anzeigen und konvertieren:



_			1	// Include the IQ Library
	Bewegung	-	2	<pre>#include "iq_cpp.h"</pre>
Bewegung	Motor.spin(forward);	?	3	
	Manage anis Tan (forward 00 degrees) -		4	// Allows for easier use of the VEX Library
Aussehen	Notor.spinror(loiward, so, degrees);	\odot	6	using namespace vex,
	Motor.spinToPosition(90, degrees);	?	7	int Brain_precision = 0;
Klang		-	8	<pre>float myVariable, SigA, SigB, SigC, SigA1, SigB1, SigC1;</pre>
•	Motor.stop();	(?)	9	int charGtantedt() (
Ereignisse	Motor setBosition(0 degrees):	2	10	while (true) {
	nooriocorobroron(o/ degreeb//	\bigcirc	12	// Cotnrollerfahrt
Steuerung	Motor.setVelocity(50, percent);	?	13	<pre>SigA = Controller.AxisA.position();</pre>
			14	<pre>SigC = Controller.AxisC.position();</pre>
Fühlen	Motor.setStopping(brake);	(?)	15	SigA1 = SigA + SigC;
	Manual (50		16	Sigg1 = SigA - SigC;
Variablen	motor.setMaxiorque(su, percent);	$\mathbf{\cdot}$	18	Motor6 setVelocity(SigA1 percent)
	Motor.setTimeout(1, seconds):	(?)	19	Motor7.setVelocity(SigB1, percent);
Funktionen	,-	0	20	Motor11.setVelocity(SigC1, percent);
- united to the	Aussehen		21	Motor6.spin(forward);
	Denie Genere anies (Webbles)		22	Motor7.spin(forward);
	brain.Screen.print("hello");		23	Motor11.spin(forward);
	Brain.Screen.setCursor(1, 1);	(?)	24	// Links
		0	25	Mator6 setVelocity(-30 0 percent):
	Brain.Screen.newLine();	?	27	Motor7.setVelocity(30.0, percent);
			28	Motor11.setVelocity(-100.0, percent);
	Brain.Screen.clearScreen();	?	29	Motor6.spin(forward);
	Brain Screen clearline(1);	2	30	Motor7.spin(forward);
	brain.Screen.clearLine(1);	\odot	31	<pre>Motor11.spin(forward);</pre>
	Kland		32	Walt(20, msec);
	Riang		34	/ Rechts
	Brain.playSound(siren);	\odot	35	while (Controller.ButtonEUp.pressing()) {
	Brain.playNote(3, 0, 1000);	?	<> En	rors
	Ereignisse		00	
	Controller.ButtonEUp.pressed(callback):	(?)	0	
	2.2	0		

In ein Text-Projekt konvertieren





FAQs

info@insite-education.de

