

## Megamark RobotC Setup Guide

ROBOTC is a cross-robotics-platform programming language for popular educational robotics systems. This tutorial will show you how to set up core RobotC for Arduino software to program the Choitek Megamark Robot Platform.



NOTE: Unlike the other software libraries, using RobotC to program the Choitek Megamark robot requires use of a special kind of Arduino, specifically the **Arduino Mega 1280**, shown below:



Keep in mind that the standard, newer **Arduino Mega 2560** is **not compatible with RobotC** due to the fact that it does not have an onboard **FTDI converter, which RobotC requires**. Although the Arduino Mega 1280 is deprecated and is no longer manufactured by the official Arduino organization, other sources produce clones for the Arduino Mega 1280 board which are still available.

## Downloading and Installing ROBOTC for Arduino

**Step 1:** First, download and install ROBOTC for Arduino from the [official RoboMatter website](#).



### ROBOTC for Arduino

**Platforms:** Duemilanove, Uno, Mega 1280 and Mega 2560

**Version:** 3.65

**Date Posted:** February 17, 2015



#### Free Download!

**Download version 3.65**  
(includes free trial)

#### Free License File

[Click here to download the FREE license file](#)

Place the attached license file in the ROBOTC 3.x directory (The default directory for a 64-bit computer is: C:\Program Files (x86)\Robomatter Inc\ROBOTC Development Environment 3.x). Run ROBOTC afterward once with elevated privileges. You can open ROBOTC normally after doing this once. Please contact support if you have any issues.

#### Already have ROBOTC for NXT or ROBOTC for CORTEX?

For more information on how to install additional platforms, [Click here for more instructions >>](#)



Note that RobotC for Arduino is now free software. Be sure to also download the corresponding Free License File.

**Step 2:** Now download the FTDI drivers for the Arduino Mega 1280 from Adafruit here:



### USBtinyISP

AVR programmer & SPI interface

[Overview](#)

[F.A.Q.](#)

[TOOLS](#) [PROGRAMMING](#) [♥](#)

### Drivers

by lady ada

- AVR programmer & SPI interface
- Windows 7, 8 & XP

Before you plug in your board, you'll need to possibly install a driver!

Click below to download our Driver Installer

[Download Adafruit Driver Installer \(v2.0.0.0\)](#)

(If you have not done so already, it is highly recommended that you also install the latest version of the Arduino IDE as well according to the steps in the [Megamark Arduino Setup Guide](#).)

## Running ROBOTC for Arduino to control the Megamark

**Step 3:** Go ahead and download the *Megamark Library for ROBOTC*, which can be found on Github or the main Choitek website. Extract it and place the examples in your desired location.

Choitek / Choitek-Megamark

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Branch: master Choitek-Megamark / Libraries / Megamark RobotC /

Create new file Upload files Find file History

johnchoi313 Added new libraries and updates! Latest commit 770a4d9 on Aug 24, 2017

File	Commit Message	Date
Elbows.c	Added new libraries and updates!	4 months ago
Grippers.c	Added new libraries and updates!	4 months ago
Lasers.c	Added new libraries and updates!	4 months ago
RobotExplorerTest.c	Added new libraries and updates!	4 months ago
RobotMotionTest.c	Added new libraries and updates!	4 months ago
Shoulders.c	Added new libraries and updates!	4 months ago
Wheels.c	Added new libraries and updates!	4 months ago

© 2017 GitHub, Inc. Terms Privacy Security Status Help

Contact GitHub API Training Shop Blog About

**Step 4.** Go to **File->Open and Compile** and open `elbows.c` from the Megamark RobotC examples.

```

1 #pragma config(CircuitBoardType, typeCktBoardMega)
2 #pragma config(Motor, servo_5, leftElbow, tmotorServoStandard, openLoop, reversed, IOPins, dgt140, None)
3 #pragma config(Motor, servo_6, rightElbow, tmotorServoStandard, openLoop, IOPins, dgt124, None)
4
5 /* This example rotates both left and right elbows up and down in a loop. */
6
7 task main()
8 {
9     while(true)
10    {
11        // Rotate left elbow fully down.
12        motor[leftElbow] = -127;
13        wait1Msec(2000);
14        // Rotate left elbow fully up.
15        motor[leftElbow] = 127;
16        wait1Msec(2000);
17        // Rotate right elbow fully down.
18        motor[rightElbow] = -127;
19        wait1Msec(2000);
20        // Rotate right elbow fully up.
21        motor[rightElbow] = 127;
22        wait1Msec(2000);
23    }
24 }
25
  
```

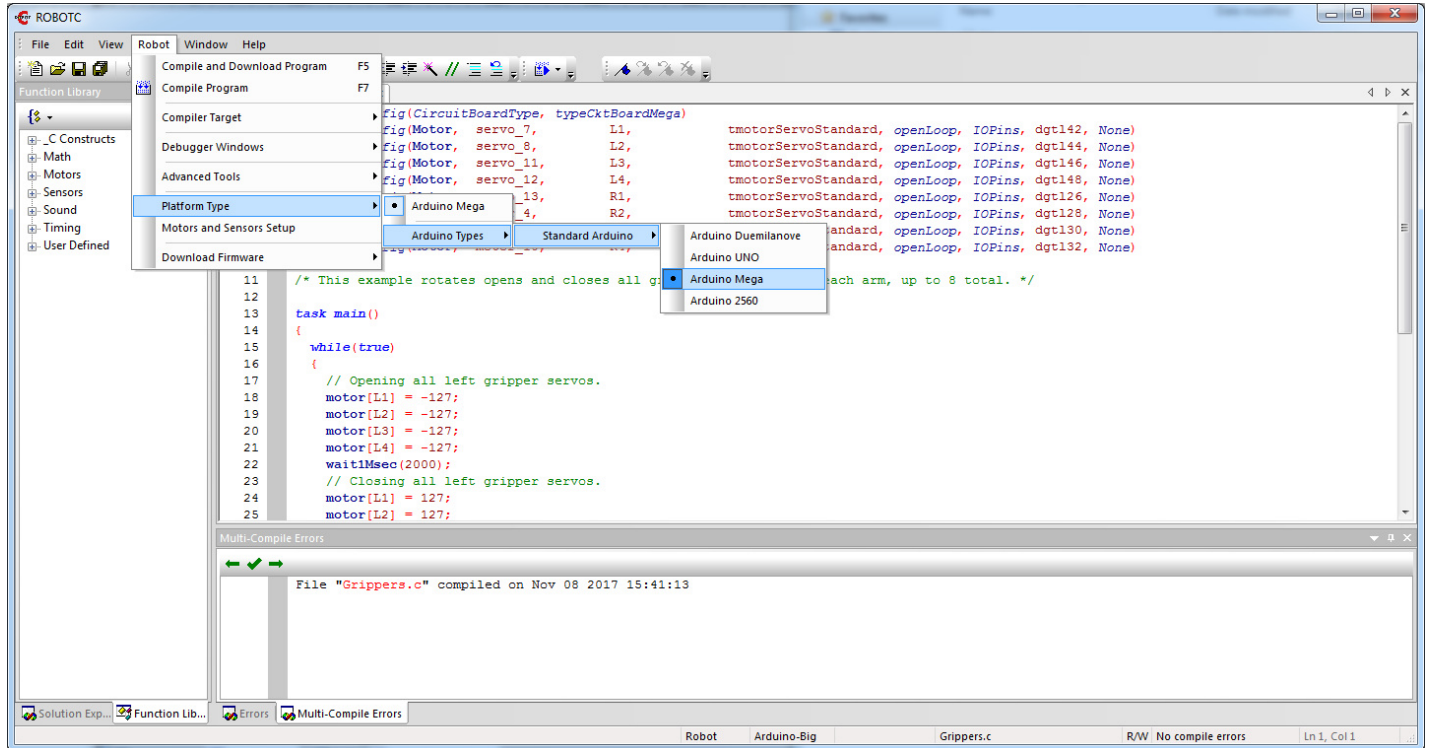
Errors

File "Elbows.c" compiled on Dec 31 2017 23:12:45

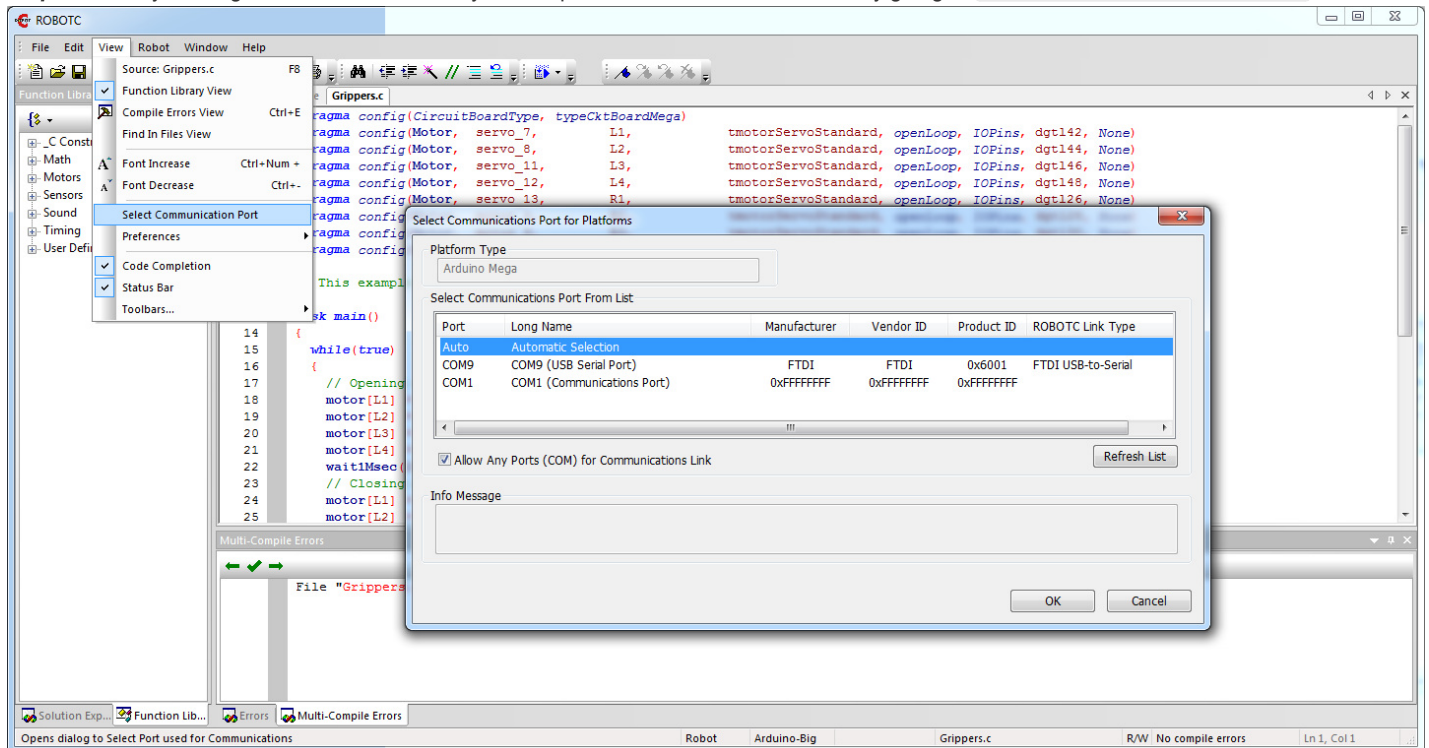
Solution Exp... Function Lib... Errors Multi-Compile Errors

Open an existing document Robot Arduino-Big Elbows.c R/W No compile errors Ln1, Col1

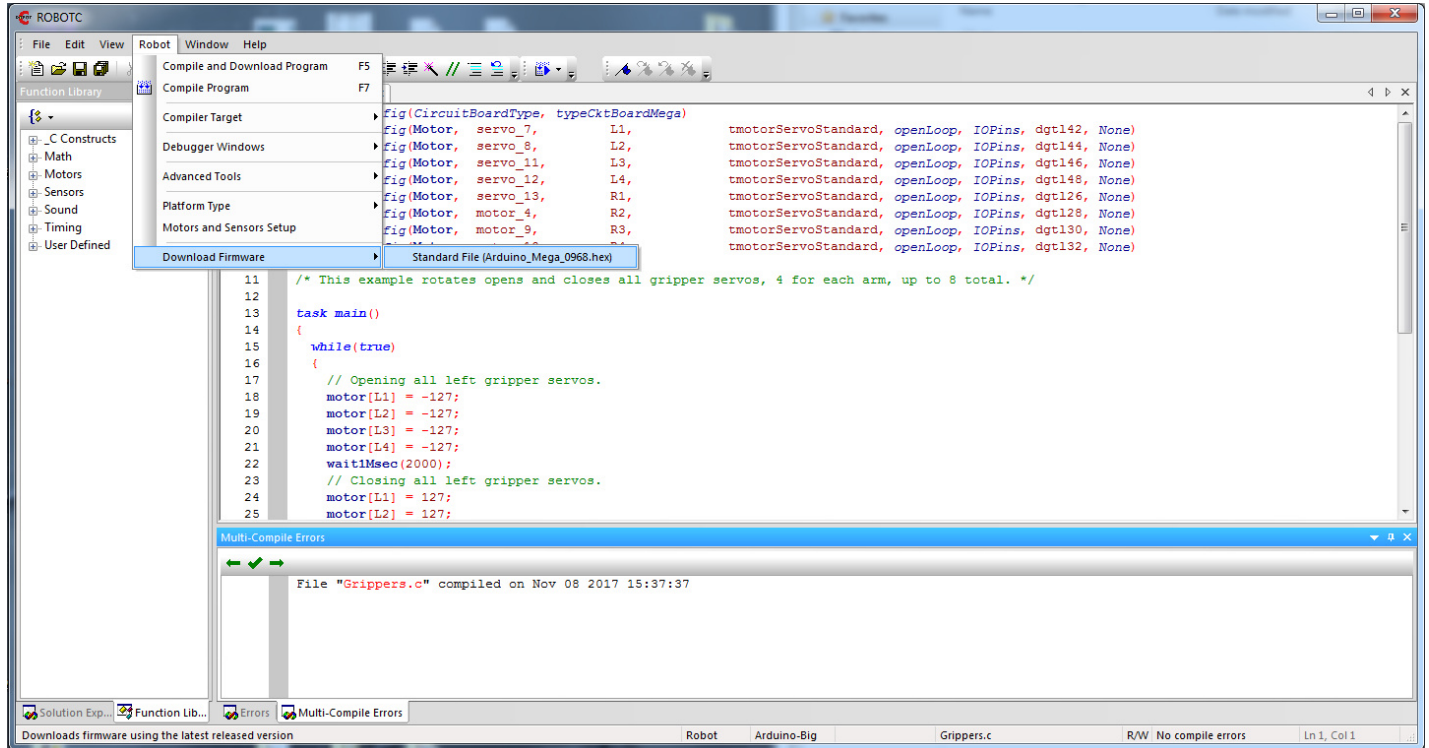
**Step 5.** Select the Board type by going to Robot->Platform Type->Arduino Types->Standard Arduino->Arduino Mega:



**Step 6.** Attach your Megamark robot via USB to your computer and select the COM Port by going to View->Select Communication Port:

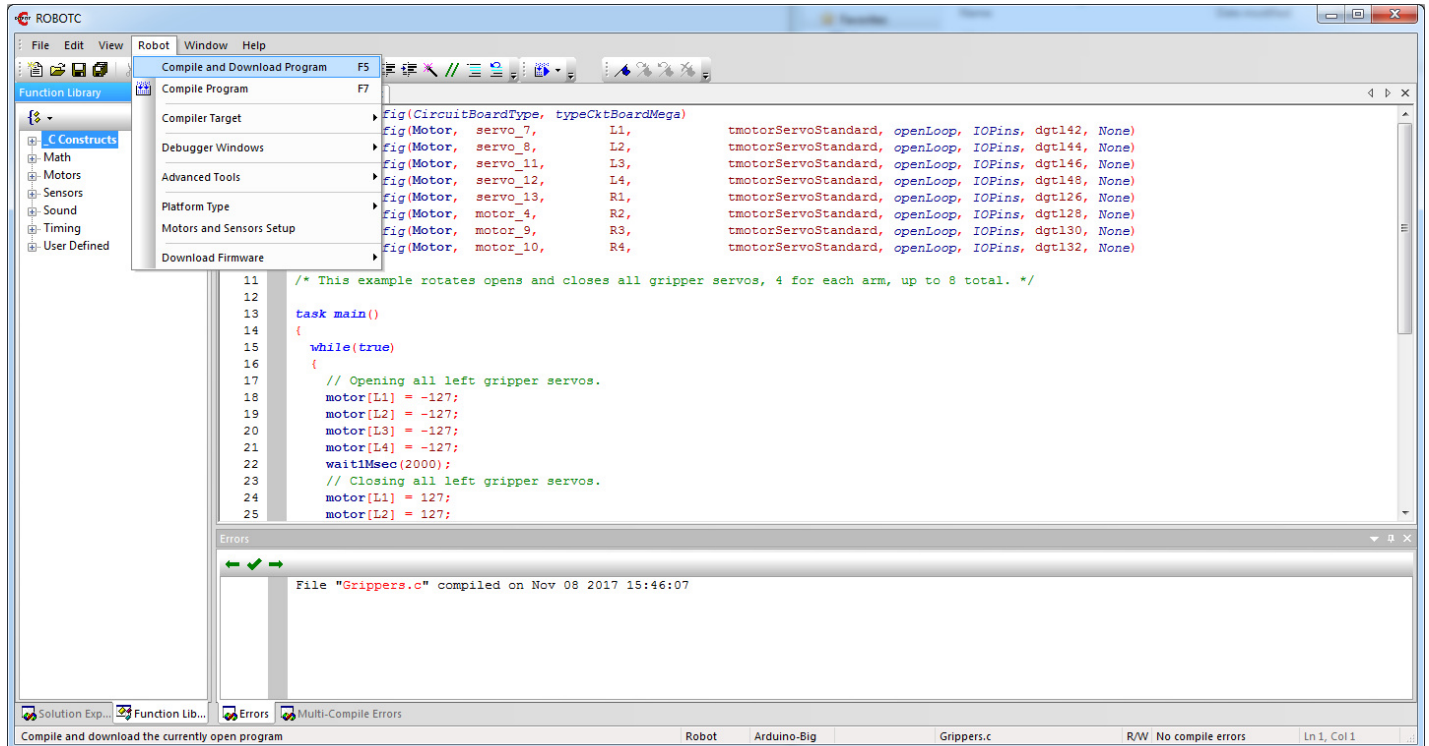


**Step 7.** Flash the RobotC firmware by going to Robot->Download Firmware->Standard File (Arduino\_Mega\_0968.hex):



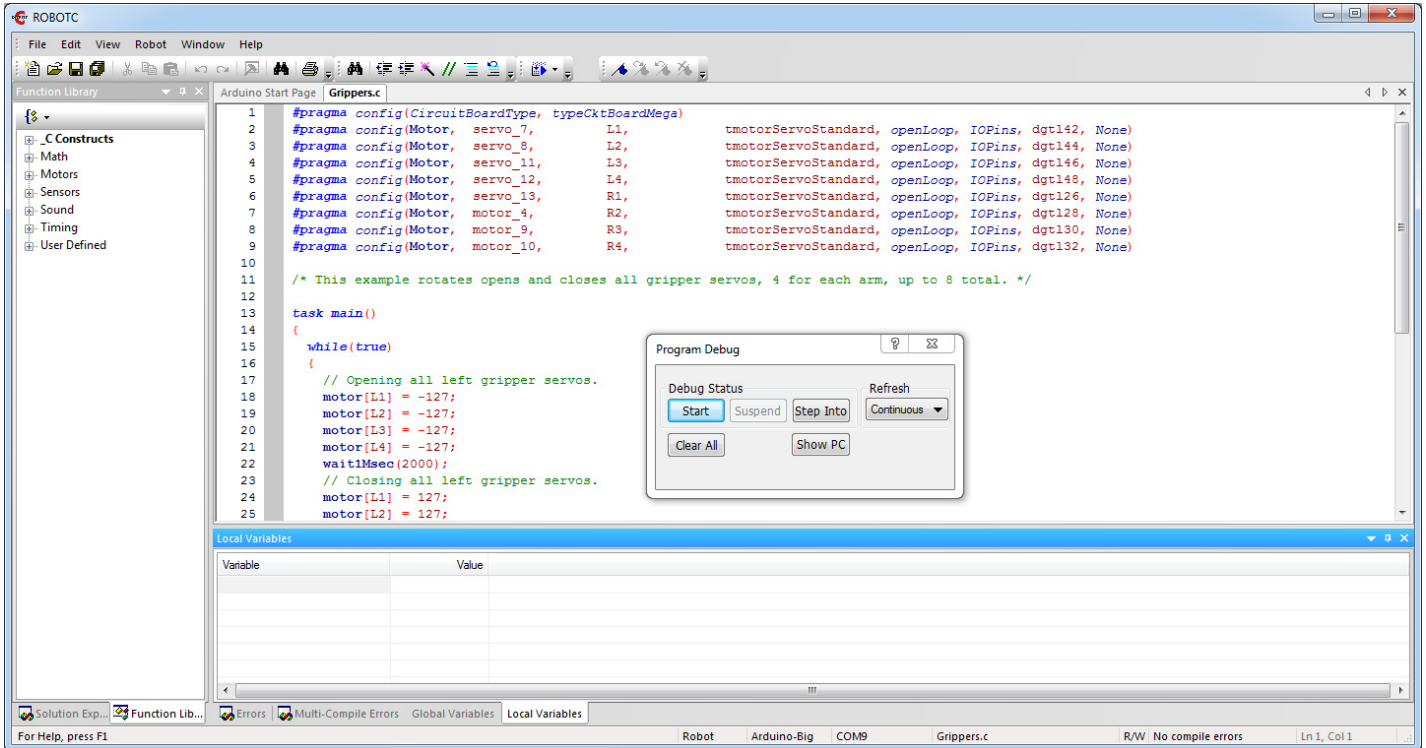
(Note: You only need to do this once.)

**Step 8.** Compile and download the program by going to Robot->Compile and Download Program:



(Note: Make sure all of your code syntax is correct! If not, the program will fail to load onto the Arduino Mega.)

Step 9. Finally, run the program by clicking on the Start button in the Program Debug dialog that pops up:



The screenshot displays the ROBOTC IDE interface. The main window shows the code for `Grippers.c`. The code includes motor configurations for eight servos and a `task main()` function that opens and closes the servos. A `Program Debug` dialog box is overlaid on the code, showing the `Start` button highlighted. Below the code editor, the `Local Variables` window is visible, which is currently empty.

```
1 #pragma config(CircuitBoardType, typeCktBoardMega)
2 #pragma config(Motor, servo_7, L1, tmotorServoStandard, openLoop, IOPins, dgt142, None)
3 #pragma config(Motor, servo_8, L2, tmotorServoStandard, openLoop, IOPins, dgt144, None)
4 #pragma config(Motor, servo_11, L3, tmotorServoStandard, openLoop, IOPins, dgt146, None)
5 #pragma config(Motor, servo_12, L4, tmotorServoStandard, openLoop, IOPins, dgt148, None)
6 #pragma config(Motor, servo_13, R1, tmotorServoStandard, openLoop, IOPins, dgt126, None)
7 #pragma config(Motor, motor_4, R2, tmotorServoStandard, openLoop, IOPins, dgt128, None)
8 #pragma config(Motor, motor_9, R3, tmotorServoStandard, openLoop, IOPins, dgt130, None)
9 #pragma config(Motor, motor_10, R4, tmotorServoStandard, openLoop, IOPins, dgt132, None)
10
11 /* This example rotates opens and closes all gripper servos, 4 for each arm, up to 8 total. */
12
13 task main()
14 {
15     while(true)
16     {
17         // Opening all left gripper servos.
18         motor[L1] = -127;
19         motor[L2] = -127;
20         motor[L3] = -127;
21         motor[L4] = -127;
22         wait1Msec(2000);
23         // Closing all left gripper servos.
24         motor[L1] = 127;
25         motor[L2] = 127;
```

Variable	Value

**Step 10:** Your robot will now play the RobotC script continuously until the script closes. If you were running the `elbows.c` script, the robot should now be happily moving its elbows in a continuous up and down motion! Be sure to try out the other examples to get a more comprehensive sense of how to program the Megamark robot using RobotC.



*That was pretty easy wasn't it? Now go out there and make some code of your own like the awesome robotics engineer you know you are!*