

Welcome back!

144 days to competition!

A few changes from the fall...

- Code::Blocks!
- Unit testing
- Stricter architecture
- ACTUAL CODE!

Code::Blocks



C::B v. Eclipse

- Eclipse
 - Powerful and popular
 IDE for many
 languages
 - Development
 experience can be
 buggy for C/C++
 - Multiple executables requires delving into makefiles

• C::B

- Specifically designed for C and C++
- Many templates for
 both console and GUI
 type applications
- Built in support for multiple executables

"Do one thing well or many things poorly."

Downloading Code::Blocks 10.05

😣 🖨 🗉 Ubuntu Software Center	
Isoftware Installed History	
Code::Blocks IDE Configurable and extensible IDE	
🕗 Installed	Remove
Code::Blocks is a cross-platform Integrated Development Environment (IDE). It is based on a self-developed plugin framework allowing unlimited extensibility. Most of its functionality is already provided by plugins. Plugins included in the base package are: Compiler frontend to many free compilers Debugger frontend for GDB (and CDB for windows platforms) Source formatter (based on AStyle) Wizard to create new C++ classes Code-completion / symbols-browser (work in progress) Default MIME handler Wizard to create new Code::Blocks plugins To-do list Extensible wizard based on scripts Autosave (saves your work in the unfortunate case of a crash) <u>Developer Web Site</u>	No screenshot available
Add-ons	
🞯 🕍 The GNU Debugger (gdb) 🛛 ★★★☆☆	
Sontrib plugins for Code::Blocks IDE (codeblocks-contrib)	
🞯 🕍 WxWidgets Cross-platform C++ GUI toolkit (GTK+ development) (libwxgtk2.8-dev)	
People Also Installed	

Setting up Code::Blocks



Project Files

- Code::Blocks projects are based on .cbp files (xml-style project description files)
- C::B uses "Global Variables" and relative paths for search directories
- We can take advantage of this to share a common project file in the git repo

Opening the IGVC Project File

- Open C::B
- Click File->Open
- Navigate to the IGVC git repo folder
- Select igvc[year].cbp

🖌 🔹 🖾 matt i	gvc 2013 software			
Places	Name	▼	Size	Modified
Q Search	📄 bin			01/03/201
Recently Used	📄 obj			01/03/201
🙈 matt	📕 src			01/13/201
Desktop	🖪 IGVC2013.cbp		4.2 KB	01/13/201
OS Documents				
OS				
Documents				
Pictures				
Videos				
Downloads				
÷ -		Code::Blocks proje	ect files	
		Ca	ancel	Open

- When you open the IGVC project, you will probably see the Global Variables window pop up asking you to define global variables for libraries used in our project, like Boost!
- To make the project work, you must install the needed libraries and complete the fields...

base

- This field must be filled
- If your library installs to a single directory with /lib/ and /include/ subfolders, put that path here
- Otherwise, as in this example, use the closest common parent folder

Current S	et:	default	-	Clone	New	Delete
Current Variable:		boost	-	Clone	New	Delete
Builtin f	ïelds				User-define	d fields
base	/usr/					
	the bas	e member is	mandato	bry		
include	/include	/boost				
lib	/lib					
obj						
cflags						
				=		

- include
 - This is the directory to find the library's header (.h/.hpp) files

• lib

 This is the directory to find the library's lib (.o/.so/etc) files

😣 Globa	l Variabl	e Editor				
Current S	et:	default	-	Clone	New	Delete
Current Variable:		boost	-	Clone	New	Delete
Builtin f	ïelds				User-defined	d fields
base	/usr/					
	the bas	e member is	mandat	огу		
include	/include	/boost				
lib	/lib					
obj						
cflags						
Iflags						
2						Close

• To reference Global Vars in C::B settings, use this format:

\$(#boost)

Refers to the base folder

\$(#boost).include

- Refers to the include subfolder
- Similar format can be used to access other subfolders

- Targets in C::B allow us to generate multiple executables from a single project file
- The following demo will demonstrate how to add a new target...

- Go to Project-> Properties->Build Targets
- Select Add and give the target a descriptive name

Project setting	Build targets	Build scripts Notes	Debugger	Libraries	C/C++ parser options		
Build targets		Selected build target o	options				
TestSerial	Add	Platforms:	tforms: All				
TestGPS	Rename	Туре:	Console application 💌				
	Duplicate		Pause when execution ends				
	Delete		Create import library				
			Create .DEF exports file				
		Output filename:	bin/Debug/TestSerial				
V	Virtual targets		🧭 Auto-generate filename prefix				
	Dependencies		🧭 Auto-generate filename extension				
	Re-order	Execution working dir:)[.		
	Ruild options	Objects output dir:	obj/Debug/				
	Build options	Build target files:					
		 src/IGVC.hpp src/actuators/motion src/actuators/motion 	ors/ArduinoM ors/ArduinoM	otorControl otorControl	/2013Arduino/Arduino /2013Arduino/arduino		
	Create project from target	Toggle checkmarks			Selected file propertie		

- Type: should be Console app
- Check "Pause when execution ends"
- Select all classes in the list that need to be compiled for your target

Project setting	Build targets	Build scripts Notes	Debugger Libraries C/C++ parser option		
Build targets		Selected build target o	ptions		
TestSerial	Add	Platforms:	All Console application		
TestGPS	Rename	Туре:			
	Duplicate		Pause when execution ends		
	Delete		Create import library		
			Create .DEF exports file		
		Output filename:	bin/Debug/TestSerial		
	Virtual targets		🧭 Auto-generate filename prefix		
	Dependencies		Auto-generate filename extension		
	Re-order	Execution working dir:	r: [][
	Build options	Objects output dir:	obj/Debug/		
	build options	Build target files:			
		 src/IGVC.hpp src/actuators/moto src/actuators/moto 	ors/ArduinoMotorControl/2013Arduino/Arduin		
	Create project from target	Toggle checkmarks	Selected file propert		

• Output Filename

- This is the executable file that will be generated when the target is built
- This file should be somewhere in the /bin/ subdirectory and should probably match the name of your target so it's easy to know what's what when looking through /bin/

Project set	tings	Build targets	Build scripts	Notes	Debugger	Libraries	C/C++ parser options	
Build targets			Selected build	target o	ptions			
TestSerial		Add	Platforms:	Platforms: All				
TestGPS		Rename	Туре:		Console app	lication	•	
	D	Duplicate			Pause when execution ends			
		Delete			Create import library			
	Virtual ta				Create .DEF exports file			
			Output filename:		bin/Debug/TestSerial .			
		irtual targets			👿 Auto-generate filename prefix			
		ependencies			Auto-gen	erate filenan	ne extension	
		Re-order	Execution wor	king dir:)	
		uild options	Objects outpu	bjects output dir:	obj/Debug/			
		ound options	Build target fil	es:				
		src/IGVC.h	ors/mot	ors/ArduinoM	lotorContro	l/2013Arduino/Arduinol		
		Create project from target	Toggle checkmarks			iotor contro	Selected file propertie	

Lastly, let's look at one of the many wizards
 C::B provides to make our C++ lives magicaly delicious

- Go to File->New-> Class
- Fill out the Class name
- Most of the rest will be filled out for you at that point

🛛 😵 Create nev	v class				
Class definition Class name:	lyClass			Member	variables Add new:
Arguments:					unsigned int m_Counter
🗹 Has destruc	tor	Has copy ctor			🗹 Add "Getter" method
🗹 Virtual dest	ructor	uctor 📃 Has assigment op.			🧭 Add "Setter" method
Inheritance	nother class				Remove prefix:
Ancestor:					Add
Ancestor's inc	clude filename:	<>		Remov	/e
Scope:	Scope: public			Documen	tation ocumentation where appropriate
File policy Header and Folder: //bome	implementatio	on file shall be in same fo	older		
	implomostati	n filo shall always he los	105 6360		(<u></u>)
	Implementatio	on the shall always be lot	wer case		1
Header rile				nentation r	lementation file
rilesses		igvc/2013/soltwaj		incruce imp	
Filename:	MyClass.h		Folder	r:	/home/matt/igvc/2013/soft
	Add guard	block in header file	Filena	me:	MyClass.cpp
Guard block:	MYCLASS_H		Header include:		"MyClass.h"
		Create	Cance	el	

- If you already have the list of properties your class is going to have, you can use the *Member Variables* group to add properties with optional getters and setters auto-generated
- These can, of course, always be added/edited by hand later

🛛 😣 Create nev	v class					
Class definition						
Class name: 🛛	IyClass			Member	variables	
Arguments:					unsigned int m Counter	
🗹 Has destruc	tor	Has copy ctor			Add "Getter" method	
🗹 Virtual dest	ructor	Has assigment op.			👿 Add "Setter" method	
Inheritance					Remove prefix:	
🗌 Inherits ar	nother class				m	
Ancestor:						
Ancestor's inc	lude filename:			Remov	Add	
Connection				Documen	tation	
scope.	Scope: public			Add d	locumentation where appropriate	
File policy	implementati	an filo chall ha in camo fi	aldar			
Meader and	Implementatio	on file shall be in same fo	older			
Folder: /home	/matt/igvc/20	13/software/				
Header and	implementatio	on file shall always be lo	wer case			
Header file			Impler	nentation f	ïle	
Folder:		igvc/2013/softwa	🗹 Ge	👿 Generate implementation file		
Filename:	MyClass.h		Folde	r:	/home/matt/igvc/2013/soft	
	🗹 Add guard	block in header file	Filena	ime:	MyClass.cpp	
Guard block:	MYCLASS_H		Head	er include:	"MyClass.h"	
		Create	Canc	el		

IMPORTANT

For the purpose of consistency in the organization of our code, you MUST check "Header and implementation.... same folder" !

😣 Create nev	v class					
Class definition	w Class			Member	variables	
Arguments:	Jecos,				Add new: unsigned int m_Counter	
🗹 Has destruc	tor:	Has copy ctor			Add "Getter" method	
Virtual dest	ructor	Has assigment op.			Add "Setter" method	
🗌 Inherits ar	other class				m_	
Ancestor:	ludo filopomo			Remov	Add	
Scope:	Ancestor's include ritename: <-> Scope: public			Documen	itation	
File policy Header and	implementatio	on file shall be in same fo	older			
Folder: /home	/matt/igvc/20	13/software/				
Header and	implementatio	on file shall always be lov	wer case			
Header file			Implen	nentation f	ile	
Folder:		igvc/2013/softwa	🗹 Ge	🧭 Generate implementation file		
Filename:	MyClass.h		Folder	r:	/home/matt/igvc/2013/soft	
	🛃 Add guard	block in header file	Filena	me:	MyClass.cpp	
Guard block:	MYCLASS_H		Heade	er include:	"MyClass.h"	
		Create	Cance	el		

Important things to remember

- Always use relative paths or environment variables to keep the project file sharable
- Use 'targets' to add new executables
- Don't forget the "Put headers and source files in same folder" check-box when creating new classes

The End

There are many more features and wizards that you will discover and take advantage of as you work in Code::Blocks. These are only the basics that you'll use early and often.