# Propeller C Tutorials

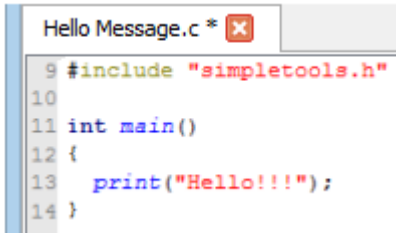
## Propeller Brains For Your Inventions (/tutorials/language/propeller-c/propeller-brains-your-inventions)

So what is a microcontroller? And then what is a *multicore* microcontroller? In this short read, meet the Propeller chip, a multicore microcontroller that makes inventing easier with teamwork.

## Propeller C - Set Up SimpleIDE (/tutorials/language/propeller-c/propeller-c-set-simpleide)

Download and set up SimpleIDE, the programming environment for Propeller C. Installers available now for Windows and Mac, and i386 binaries are available for Linux.

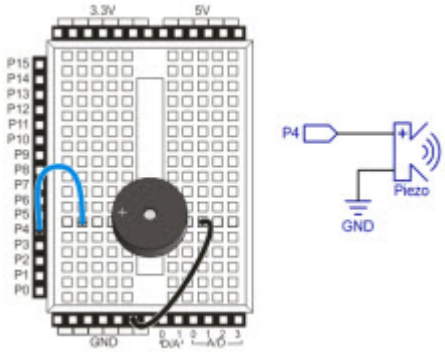
## Propeller C - Start Simple (/tutorials/language/propeller-c/propeller-c-start-simple)



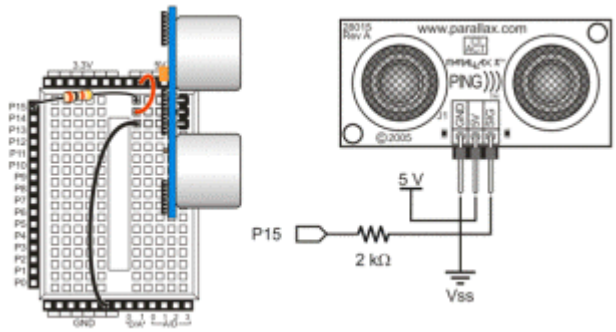

Get started with C-language programming for the multicore Propeller microcontroller. The simple example programs use variables, do math, count loops, make decisions, and store information.

## Propeller C - Simple Circuits (/tutorials/language/propeller-c/propeller-c-simple-circuits)

Build simple circuits on the Propeller Activity Board, and write Propeller C programs to interact with them. Blink lights, beep a speaker, measure voltage, and more.




## Propeller C - Simple Devices (/tutorials/language/propeller-c/propeller-c-simple-devices)




Our C device libraries and example code make it easier than ever before to take advantage of the Propeller chip's multiple cores. This is part of the **Propeller C Tutorial series** (/propeller-c-tutorials) for the **Propeller Activity Board** (/propellerAB)

## Propeller C - Functions (/tutorials/language/propeller-c/propeller-c-functions)

```
void hello(void);

int main()
{
  hello();
}

void hello(void)
{
  printf("Hello!");
  pause(500);
}
```

A function is a little piece of reusable code designed to do a specific task. This Propeller C Tutorial series will show you how to use functions, pass information to them, and get information back from them. It will show you how C functions work with the Propeller's memory and multiple cores.

## Propeller C Reference (/support/C/propeller-c-reference)


Find references for Simple Libraries, the SimpleIDE software, Propeller C, and the PropellerGCC compiler, as well as an archive of previous Propeller C tutorials.

## Propeller C (/tutorials/propeller-c-0)


**Do More with 8 Cores**

8 processors in one chip bring your electronics project from idea to reality *fast.* And, our Propeller C Tutorials, Activity Board, and Simple Libraries make it *easy.*

## (Legacy Version) Propeller C Programming with the ActivityBot

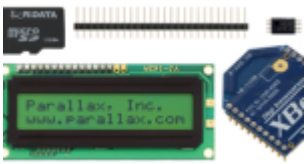
(/tutorials/robot/activitybot/legacy-version-propeller-c-programming-activitybot)

```
#include "simpletools.h"
#include "abdrive.h"

int main()
{
  drive_speed(-64, 64);
  pause(350);
  drive_speed(0, 0);
}
```

Get ready to build your robotics skills with this **Propeller C** Tutorial for the zippy ActivityBot. Learn how to make your robot navigate using a variety of sensors, using touch, ultrasound, visible light, and more!

## Propeller C - Simple Protocols (/tutorials/language/propeller-c/propeller-c-simple-protocols)




Learn about communication protocols with this Propeller C tutorial series from Andy Lindsay. This series is currently a work-in-progress; activities on Half-Duplex and Full-Duplex Serial Communication are available now, with more to come in the near future. Keep checking back!

## Propeller C Library Studies (/tutorials/language/propeller-c/propeller-c-library-studies)

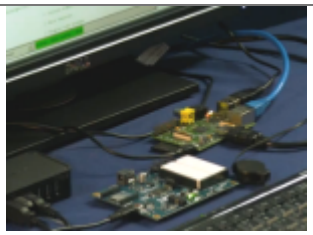
Test Awesome Library.c

```
/*
  Test Awesome Library.c
*/

#include "simpletools.h"
#include "awesome.h"
```

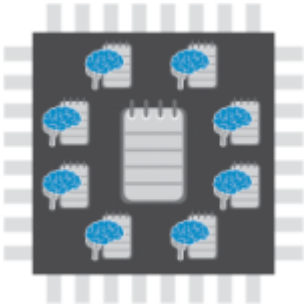
Learn how to make your own **custom Simple Libraries** for your Propeller Activity Board. This Propeller C Tutorial walks you through creating a library, tricks for reducing code size, and making your library to work well with the ones already in your SimpleIDE Learn folder.

## Raspberry Pi (/tutorials/language/propeller-c/propeller-c-set-simpleide/raspberry-pi)



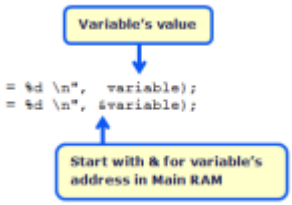
Learn how to program your Propeller microcontroller using your Raspberry Pi device. These step-by-step instructions will help you show you how to install and operate SimpleIDE on a Raspberry Pi operating system.

## Multicore Approaches (/tutorials/language/propeller-c/multicore-approaches)



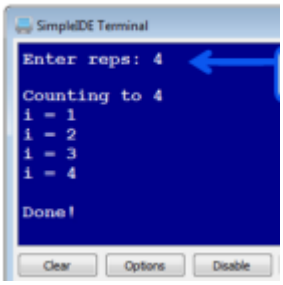
This Propeller C Tutorial shows you how make use of the Propeller chip's multiple cores. Launch functions into other cores, monitor processes in other cores, share data between cores, and print from other cores.

## Variable Values and Addresses (/tutorials/language/propeller-c/propeller-c-start-simple/variable-values-and-addresses)



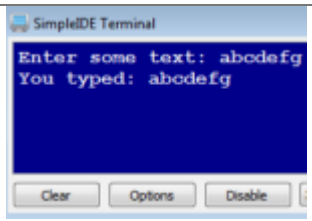

When working with variables, knowing where they are being stored is often just as important as knowing what value is being stored. This tutorial will introduce you to the topic of variable addresses.

## Get Values From Terminal (/tutorials/language/propeller-c/propeller-c-start-simple/get-values-terminal)



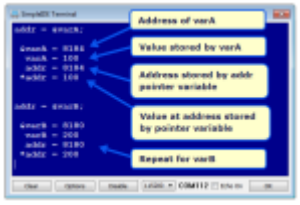

This Start Simple tutorial will teach you how to pass variable addresses to a function, and then use those addresses later on in your code.

## Get Text from Terminal (/tutorials/language/propeller-c/propeller-c-start-simple/get-text-terminal)




Learn how to store text in character arrays and create your own password checker with this Propeller C - Start Simple tutorial.

## Propeller C - Pointers on Data (/tutorials/language/propeller-c/propeller-c-pointers-data)



This guide introduces some common data management techniques you'll find useful with Propeller C programming. This is an advanced topic, beginners may want to work through Start Simple and Functions before trying this section!

## About SimpleIDE Windows, Projects, Tabs, and Files (/support/C/propeller-c-reference/about-simpleide-windows-projects-tabs-and-files)


This reference page explains how to open multiple Propeller C programs with SimpleIDE. It also explains what multiple tabs within SimpleIDE are used for, and the difference between Simple View and Project View.