



(Formerly known as, "Getting Started in Electric Radio Controlled Aircraft"). A site to share my RC passion and enthusiasm with others, filtered through the lens of an aeronautical engineer and hobbyist, and to share with others some of the interesting technical things I've learned about electric radio controlled aircraft, helicopters, gyroscopic precession, lipo batteries, motors, ESCs, sensors, Arduino microcontroller programming, etc. Thanks for reading! ~Gabriel

[Blog Home](#)
[eRCaGuyStore\\$\\$](#)
[Hot Link: Parallel Charging LiPos](#)
[Hot Link: Beginner Airplane Setup](#)
[Hot Link: Beginner RC Helicopter & Quadcopter](#)
[Feedback? Corrections? \(Contact me\)](#)
[Hot Link: Scratch Building 20+ Planes](#)
[Getting Started In Electric RC Aircraft \(Old Website\)](#)
[My Instructables.com Page](#)
[My GitHub Repositories](#)

**SUNDAY, JANUARY 5, 2014**

## The Power of Arduino

**...And how you can learn to use computer programming to control and customize the physical world around you, including your RC vehicles...**

Subscribe to and share this blog by clicking the links to the right -->

By: Gabriel Staples

Written: 5 Jan. 2014

Last Updated: 14 June 2014

-added a bunch of new links at bottom of article (mostly Intermediate) - 14 June 2014

-added details and corrections pertaining to microcontrollers being used in servos - 17 March 2014

***Be sure to check out the links and resources at the end of this article, as they are EXTREMELY useful to anyone who uses Arduino microcontroller development boards!***

### Other Related or Interesting Articles:

- [eRCaGuy Store Just Opened](#) - please support my work
- [Parallel Charging Your LiPo Batteries](#)
- [Propeller Static & Dynamic Thrust Calculation](#)
- [Bungee-Launched "Stingray" Free Flight Glider](#)
- [Beginner RC Airplane Setup](#)
- [TED Talk - Massimo Banzi: How Arduino is open-sourcing imagination](#)

**One Good Use of an Arduino: Use it as a Touch Lamp controller:**

### SUBSCRIBE & SHARE!



Like Share 19  
Tweet 0



MY FAVORITE QUOTE/MY MOTTO:

**"Be a *thinker*, and be a *doer*."**

~Destin, from his "Smarter Every Day" series on YouTube

TOTAL PAGEVIEWS

**40,354**

SEARCH THIS BLOG

Search

BLOG ARTICLES BY MONTH  
(CLICK ARROWS TO EXPAND LISTS)

▼ 2014 (12)

► May (2)

► April (3)

► February (5)

▼ January (2)

## Background:

So, you may be wondering why I haven't written an article in the last few months. To be honest, writing a well-informed article, with plots, pictures, and sources takes vast amounts of time, and life gets busy. However, I should also tell you that I have been heavily involved in **Arduino microcontroller programming** for the last several months, and I have spent countless hours learning about and programming Arduino. I have now built up a variety of Arduino tools and resources that I use, including dozens of bits of test code, many projects and programs, and hundreds of files, links, and resources. **I have even written some code and functions which I have seen nowhere else, which I think are very valuable, and which I intend to periodically share here on my blog. In other words, I think I have some useful things to contribute to the vast world of Arduino, some of which will help bridge the world between Radio Control, robotics, and Arduino microcontroller programming.** Though I am constantly learning, and constantly seeking help from resources and people who know much more than I, I would consider myself a very informed Arduino user with a sound, yet constantly growing, understanding of electricity, electrical engineering, programming, and computer science.

## Why have I recently been doing Arduino programming instead of doing Radio Control stuff?

Learning Arduino *is* doing Radio Control stuff. It turns out that the world of Radio Control is completely inundated with microcontrollers and programming--though perhaps very few Radio Control guys/gals are aware of this. A microcontroller is, in short, a miniature computer, which operates as intimately with hardware as is possible for code to do. It is designed to control the physical world around us. RC devices are controlled by them. The Arduino development board utilizes Atmel-brand microcontrollers. The Arduino therefore is an incredibly capable little computer which makes customizing your RC experience, adding additional sensors or functionality, or even turning your RC device into an Unmanned Aerial Vehicle (UAV), Unmanned Aerial System (UAS), or Unmanned Ground Vehicle, possible! With Arduino, you can take a standard RC device and make it an autonomous or semi-autonomous robot. The limit of what you can do is only constrained by your ingenuity and imagination!

## Microcontrollers in Radio Control Today

Without microcontrollers, like those used by the Arduino development platform, RC as we know it today would not exist! Though I have become knowledgeable on the subject very quickly, I have been doing Arduino microcontroller programming for less than a year. It wasn't until approximately April 2013 that I first picked up an Arduino and made it blink, and it wasn't until approximately July 2013 that I actually caught the vision. Now it's time for you to catch the vision too!

When I fly an RC airplane, here's where microcontrollers are used:

1. The transmitter (Tx): I know for a fact my computer-based Tx uses at least one microcontroller. That's the "computer" in the radio that stores and changes settings, shows your settings on a display, and creates the PPM (Pulse Position Modulation) signal which comes out the trainer port in the back of the radio. I don't know for sure, but I believe there is also an additional microcontroller in the RF (Radio Frequency) module which actually sends the signal out over the airwaves. I presume that if this one exists in the RF module, it would be what controls the pseudo-random 2.4Ghz frequency hopping algorithm.
2. The receiver (Rx): Every one of my 2.4Ghz Rx's uses a microcontroller to read in the signal, process it, separate the channels, and send out individual channel signals as PWM (Pulse Width Modulation) signals. These PWM signals are what go to the servos and ESC (Electronic Speed Controller).
3. The Electronic Speed Controller (ESC): Every ESC **must have** a microcontroller. Brushed speed controllers use one to chop the battery voltage via PWM, in order to throttle the motor. Brushless speed controllers use one to chop the battery voltage via PWM, in order to throttle the motor, **as well as** to do a very complicated commutation algorithm, based on feedback, which times the switching and "rotating" of electrical signals around the motor in order to allow the electromagnets to push and pull on the permanent magnets, in just the right places and at just the right times, in order to cause the motor to rotate.
4. The servos (some). Though many servos use integrated circuits (ICs), some servos have an embedded microcontroller chip to decode the incoming PWM signal from the Rx, read a potentiometer to decipher their position, and move themselves to a commanded position, using potentiometer feedback, to ensure they get there. This is what servos must do, and though the servos I have looked at use an Integrated Circuit (IC) like the AA51880 Servo Motor Controller, some more advanced servos are programmable and have customizable gains, slew rates, and other control parameters, and use embedded microcontrollers rather than ICs. One such servo is the open-source "OpenServo."
5. The charger. Every smart charger uses a microcontroller. It is safe to assume that if it has a display, it uses a microcontroller.
6. Your voltage checker and/or low voltage alarm. I use a voltage checker on the ground to verify battery voltage and cell balance before and after each flight. It uses a microcontroller. I opened it up and checked.

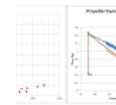
eRCaGuy Store Just  
Opened

The Power of Arduino

► 2013 (23)



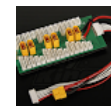
### POPULAR POSTS



Propeller Static &  
Dynamic Thrust  
Calculation - Part 1  
of 2



Getting into  
Scratch Building -  
20+ Planes with  
ONE Motor & ONE  
Power Pod!



Parallel Charging  
Your LiPo Batteries



Beginner RC  
Airplane Setup



Building the  
FliteTest NutBall  
Swappable



Arduino micros()  
function with 0.5us  
precision - using  
my  
Timer2\_Counter

Library



Thunder  
AC680/AC6  
Charger &  
Computer Data-  
Logging Software

### BLOG ARTICLES BY CATEGORY, ARRANGED ALPHABETICALLY (ARTICLES CAN BE UNDER MULTIPLE CATEGORIES)

ADC (1)  
Administrative (1)  
Analog to Digital Converter (1)  
Arduino (11)  
Arduino code (4)  
Arduino Libraries (3)  
ArduPilot (1)  
Aside (1)  
Beginner RC Airplane (11)  
Beginner RC Helicopter (2)  
Bomb Drop (1)  
Boy Scouts (1)  
Capacitors (1)  
Chargers/Charging (3)  
Contact Me (1)  
Copper Etchant Solution (1)  
Dynam Hawk Sky (1)  
Dynamic Thrust (2)

I also use a low voltage alarm on my plane to beep when my battery is low. It uses one too.

So, each and every time I fly an airplane, there are at least 5 places microcontrollers are used. Each device that uses a microcontroller must have specially-designed code on it to perform very specific tasks, that a computer programmer (or someone like you or me) put there! In a single flight, I may be using up to 10 or more (for sure I am using at least 5) microcontrollers just to make it happen. Now, with the power of Arduino, I can learn to customize and enrich my RC experience by turning my RC devices into robots, by creating custom code (firmware) to load onto an Arduino and put on my airplane or car in order to make it do...whatever I want!

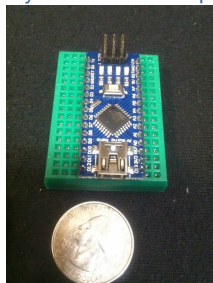
**Arduino video: Reading the throttle signal from an RC car receiver, and using it to change the blink rate of an LED (could be used as a head-lamp or tail light on the car, for instance):**

## My Arduino History:

- Sometime during the **beginning of 2013**: My good friend [Michael Cardoza](#), a high school Spanish Teacher (and great person to inspire one to try new Arduino and electronics projects), showed me his Arduino. I had no idea what that was. I wasn't very interested. Didn't look very impressive, and to me, it wasn't related to RC (boy was I wrong!).
- **13 April 2013**: I made a used Arduino Duemilanove blink. It took me ALL day on Saturday to accomplish this (prob. 8~10 hrs.). This was a very rough transition into microcontroller programming, as the stinking thing was used and had a bad bootloader. I eventually figured out how to reload the bootloader onto it using a [USBasp AVR programmer](#) I had picked up earlier from HobbyKing, but never used until that day. (By the way, the USBasp programmer uses....guess what....a microcontroller! :)).
- **July 2013**: I picked up the Arduino again. This time, I caught the vision.
- **By October 2013**: I had become completely absorbed in Arduino microcontroller programming, as a new primary hobby for the time being (it will always be part of my RC endeavors now). I began doing various projects and things, some being quite complicated and difficult for me, to continue learning Arduino.
- **Today (5 Jan. 2014)**: I continue making leaps and bounds, as able, in my understanding of and ability to use Arduino microcontroller programming.

To date (13 April 2013 to 5 Jan 2014), I estimate (via some calculations I just did) that I have spent approximately **472 hrs.** learning about and/or doing Arduino microcontroller programming or learning about and doing related circuit analysis and electrical engineering. Arduino and microcontrollers are such an amazing tool.

Please support my work by purchasing your very own Arduino-compatible Nano V3.0 device [here](#).



**A Few Helpful Arduino Resources - BE SURE TO CHECK THESE OUT - these are really fantastic sources to learn from!**

[Equations \(2\)](#)  
[ESC's \(1\)](#)  
[Feedback & Corrections \(1\)](#)  
[Flight School \(2\)](#)  
[FliteTest NutBall \(6\)](#)  
[Free Flight \(1\)](#)  
[Helicopter Control \(1\)](#)  
[HobbyKing Bixler \(1\)](#)  
[Inductance \(1\)](#)  
[Inspirational \(5\)](#)  
[Large Gauge Wire \(1\)](#)  
[LiPo Batteries \(1\)](#)  
[Microcontrollers \(6\)](#)  
[Motivation \(3\)](#)  
[Multirotors \(2\)](#)  
[Parallel Charging \(2\)](#)  
[PC Board \(1\)](#)  
[PPM \(Pulse Position Modulation\) \(2\)](#)  
[programming \(4\)](#)  
[Propellers \(2\)](#)  
[Scratch Building \(6\)](#)  
[Soldering \(1\)](#)  
[Static Thrust \(2\)](#)  
[Stingray \(1\)](#)  
[Technical Information \(9\)](#)  
[Timer2\\_Counter \(2\)](#)  
[Tips and Tricks \(8\)](#)  
[Tricopters \(1\)](#)  
[Videos \(1\)](#)  
[Website Links \(3\)](#)  
[XT60 Connectors \(1\)](#)  
[XT90 Connectors \(1\)](#)

## I'M A MORMON

[YouTube Mormon Channel](#)  
[Mormon.org](#)  
[Worship With Us](#)

## For the Beginner:

- <http://arduino.cc/en/Main/Products> - it is very important that you thoroughly understand your Arduino -- **learn all about your particular Arduino board here!**
- <http://learn.adafruit.com/category/learn-arduino> - **absolute beginner tutorials**--very fantastic website and information!
- <http://arduino.cc/en/Tutorial/HomePage> - **hundreds of beginner tutorials and examples**
- <http://arduino.cc/en/Reference/HomePage> - **Arduino reference page** -- you will come here often! **If you aren't regularly referencing these pages, you should be!** *You can also find the entire Arduino Reference right on your computer by going to Help --> Reference right inside the Arduino IDE (Integrated Development Environment). This is SUPER useful when working offline.*
- <http://forum.arduino.cc/> - the **Arduino forum** - ask your questions and get help here
- Programming Arduino Getting Started with Sketches, By Simon Monk, \$8.48 - GREAT BOOK.

## For the Intermediate:

- Programming Arduino Next Steps: Going Further with Sketches, By Simon Monk, \$12.89 - GREAT BOOK.
- <http://playground.arduino.cc/Main/LibraryList> - **hundreds of user-contributed Arduino Libraries** to expand what you can do with your Arduino!
  - Libraries I have contributed have names which begin with "eRCaGuy," so do a search for "eRCaGuy" on this page and you'll see my libraries.
- <http://playground.arduino.cc/Main/InterfacingWithHardware> - tons of user-contributed code and examples to **interface with hardware**
- <http://playground.arduino.cc/Main/InterfacingWithSoftware> - tons of user-contributed code and examples to **interface with software** (including data-logging and plotting, live data display, controlling your Arduino via a GUI on your computer, etc)
- <http://www.cplusplus.com/> - **Arduino is based on C/C++**. As you become an intermediate user, you need to know that you are NOT limited just to the functions on the main [Arduino Reference page](#). Rather, you have access to nearly \*any\* standard C/C++ function.
- <http://www.nongnu.org/avr-libc/user-manual/modules.html> - **AVR Libc reference pages** - This link contains **a list of all of the C functions (organized by header files, .h files) that are available for use on ATmega microcontrollers/Arduino!**
  - For a real description, & examples, of how to use one of these functions, however, I recommend simply Googling the function name you want, as follows: ex: I want to know how to use "scanf," so I Google "scanf c", and the first link that pops up is this: <http://www.cplusplus.com/reference/cstdio/scanf/>.
    - Or, let's say I don't know that "scanf" exists, but I want to know how to read parts of a string, out of a string. So, I Google "c read parts of a string", and the first link I find is this: <http://bytes.com/topic/c/answers/221802-how-read-part-string>. Now, I read their code and see that "scanf" is used in their code, so I can either look for the reference page for this function on Cplusplus.com directly, \*or\* I can Google "scanf c", as described above, to find a good reference page describing it in detail.
  - Alternatively, you can simply go directly to <http://www.cplusplus.com/>, or other similar C/C++ websites, and search for the function you want to use. Again, an Arduino can use virtually any C/C++ function, since that's what its language is based on!

## For the Advanced:

- <http://www.righto.com/2009/07/secrets-of-arduino-pwm.html> - an absolutely FANTASTIC article on the intricacies of controlling the **low-level PWM capabilities of an AVR microcontroller** on an Arduino
- <http://playground.arduino.cc/Main/TimerPWMCheatsheet> - excellent info. to help you modify your **Arduino PWM** output frequencies quickly
- [http://www.atmel.com/Images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P\\_datasheet.pdf](http://www.atmel.com/Images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet.pdf) - this is the **660 pg. datasheet of the Atmel ATmega 328 microcontroller**, used by most Arduinos -- this is an absolutely ESSENTIAL document if you plan on doing low-level Arduino microcontroller programming, including direct manipulation of its timers, ports, registers and things
  - Find the main ATmega 328 webpage here:  
<http://www.atmel.com/devices/atmega328p.aspx>
- <http://gammon.com.au/interrupts> - an IMMENSELY USEFUL and well-written article, by Nick Gammon,

on **AVR microcontroller interrupts**

- <http://www.gammon.com.au/forum/?id=11504> - another EXTREMELY USEFUL article by Nick Gammon, on **AVR microcontroller timers and counters**
- <http://yourduino.com/sunshop2/> and <http://arduino-info.wikispaces.com/> - I have found a lot of good, useful info here.

### Other Inspiring Links:

- TED Talk - Massimo Banzi (the primary founder of Arduino) - [How Arduino is Open-Sourcing Imagination](#)

The End.

***If you like this article, please be sure to subscribe to my blog, tweet and like it on Facebook! Links to like and subscribe can be found just below, and at the upper-right of my blog. Look for the "Subscribe & Share" section to the top-right! Also, feel free to leave comments below.***

Posted by [Gabriel](#) at 11:04 PM



+2 Recommend this on Google

Labels: [Arduino](#), [Microcontrollers](#), [programming](#)

No comments:

### Post a Comment

Thanks for your comment or question! If it is a question, I will try to get back to you quickly. Sincerely, Gabriel Staples.

Enter your comment...

Comment as: [Gabriel \(Googl](#) ▼

[Sign out](#)

[Publish](#)

[Preview](#)

☐ Notify me

### Links to this post

[Create a Link](#)

[Newer Post](#)

[Home](#)

[Older Post](#)

Subscribe to: [Post Comments \(Atom\)](#)

#### COMMENTS

Please feel free to add questions and/or comments below the articles in this blog, and I will be happy to try and answer your questions!



Picture Window template. Powered by [Blogger](#).

