

Introduction to Embedded Microcontrollers V2

Greg Reynolds Jan. 2017

It is the rare electrical item that is any more complex than a hair dryer that doesn't have a small microcontroller in it. Even an electric tooth brush, with its timer, has a micro in it. So do stoves and refrigerators. Automobiles have probably over 20 micros embedded in components like the fuel injector, speed controller, entertainment centre, tire pressure monitor, and so on. None of these items appear to be a computer since there is no mouse, screen, or keyboard. The computer is embedded in the item.

All these micros must have software written for them. This is one of the reasons that it is projected that there will be a real shortage of skilled programmers.

The purpose of this course is to introduce you to these microcontrollers. We will mount a Arduino Nano controller on a prototyping board along with a pushbutton switch and an LED. Then we will write a program to transfer the status of the switch to the LED. Basically, we will produce a very complicated light switch.

Sounds boringly simple, but even here there are hidden hardware problems which will have to be overcome with software,

One of the many attractions of solving hardware problems with software is the hardware must be put into every product made. Not good for keeping the price down. Software is paid for once. After that it is "free" to put in a product.

The Arduino Nano

This is little printed circuit board with 25 components. Most important to the user is the Atmel ATmega168 microcontroller and a serial to USB chip that allows us to communicate with the microcontroller (See the User Manual in the information package)

The Prototyping Board

In engineering, the first time a thing is made, it is usually referred to as the prototype. In electronics, the prototype is the components in the design connected together to try out the design and see if it works. These boards provide a relatively convenient way to connect components together. There is an array of square holes on 0.1" spacing. Under the holes are U shaped channels that run under groups of holes. Pushing a wire into a hole drives it into the channel. Another wire driven into another hole that has the same channel under it connects to wires. A drawing in the info packet shows what holes have common channels.

Connected holes run the whole length of the board along the sides. These are usually connected to the binding posts at the end of the board. This allows power and ground to be distributed to other holes on the board. By convention the red post is power and the black post is ground.

The Switch

The legs on the PB (push button) switch are on a metric spacing so they do not line up exactly with the holes in the protoboards. Some careful bending and fiddling with long nosed pliers might be required.

The LED

LED only work provided the current flows in the right direction. The negative terminal is identified by a small flat portion on the body of the LED. Look carefully.

The Resistor

A resistor must be put into line with the LED. The reason for this will be demonstrated by examining the data sheets for both the LED and the micro, and the value of the resistor will be calculated using Ohm's Law.

The Arduino IDE (Integrated Development Environment)

One of the reasons for the popularity of the Arduino series of microcontroller boards is a very friendly (Compared with other IDEs) programming interface. This IDE must be installed on the PC that you will use. As well, the USB driver for the USB port on the Nano must also be installed on the PC. The Arduino Programmer's Notebook should also be installed for your use. You are encouraged to study this notebook in detail at home. It will give you a very good overview of what you can make the Nano do.

The Steps We Will Follow

1. Go to Mrs. Kirby's directory on the school library and download the following onto the D drive of the PC you are using.

 Arduino IDE, Arduino directory, CH341SER driver, Arduino Notebook

2. Install the CH341 driver by opening the directory and double clicking SETUP. Now plug the Nano into an USB port. Open the PC's Control Panel, go to Drivers and Printers. At the bottom, under Unspecified, should be USB-SERIAL CH340 (COMx). Note the COM number. This number can change each time the Nano is plugged in again.

Open the IDE. Under the pulldown menu File open Preferences. Using the Browse button, change the Sketchbook location to the Arduino directory you put on the D drive.

Pull down the Tools menu. Change the Board to Arduino Nano. Change the Port to the COM number you found.

3. Pull down File, Examples, Basics, Blink. Confirm the connection by uploading Blink that makes the LED on the Nano blink. Alter Blink to change to blink rate.

4. Install the Nano, LED, resistor, push button switch, and connecting wires as shown
 In the information package.

5. Examine the program Simple_PB_LED under File, Sketchbook, AY Proto Board.

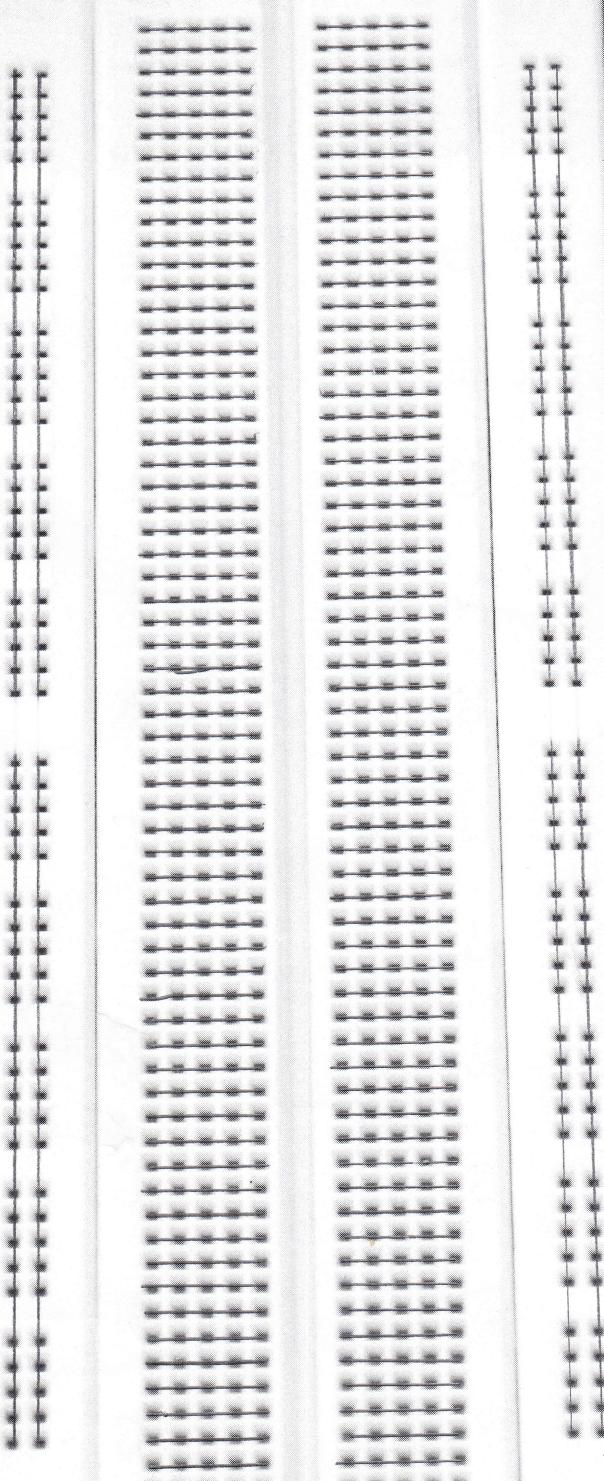
 Load into the Nano. The switch should control the state of the LED.

5. Write a program to toggle the LED with the switch. In other words, press once to change the state of the LED.

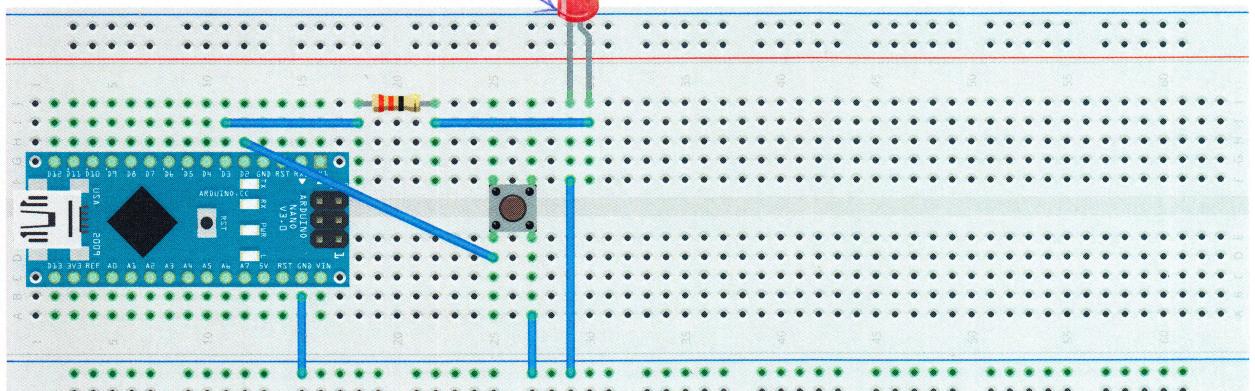
6. Carefully examine the operation of this program.

Project Boards

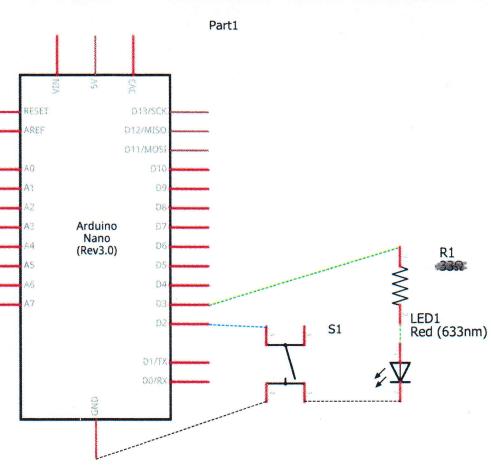
Model
KDH
GL-12F



FLAT



fritzing



fritzing

Resistor Color Codes

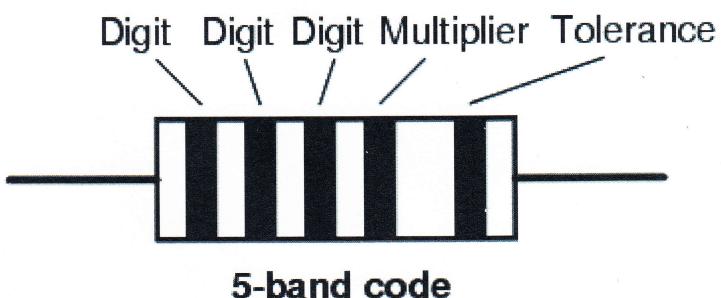
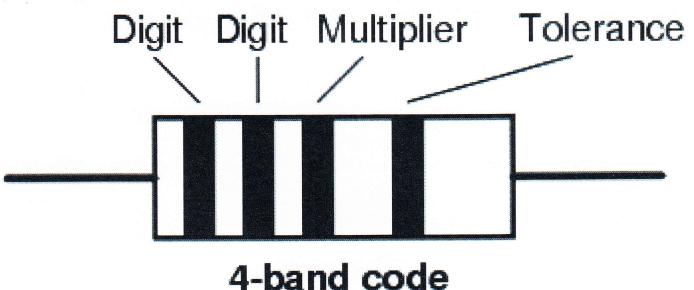
Chapter 2 - Color Codes

Try out our Resistor Color Code Calculator in our Tools section.

Components and wires are coded are with colors to identify their value and function.

Color	Digit	Multiplier	Tolerance (%)
Black	0	10^0 (1)	
Brown	1	10^1	1
Red	2	10^2	2
Orange	3	10^3	
Yellow	4	10^4	
Green	5	10^5	0.5
Blue	6	10^6	0.25
Violet	7	10^7	0.1
Grey	8	10^8	
White	9	10^9	
Gold		10^{-1}	5
Silver		10^{-2}	10
(none)			20

The colors brown, red, green, blue, and violet are used as tolerance codes on 5-band resistors only. All 5-band resistors use a colored tolerance band. The blank (20%) "band" is only used with the "4-band" code (3 colored bands + a blank "band").



Example #1



A resistor colored *Yellow-Violet-Orange-Gold* would be $47\text{ k}\Omega$ with a tolerance of $\pm 5\%$.

Example #2



A resistor colored *Green-Red-Gold-Silver* would be $5.2\ \Omega$ with a tolerance of $\pm 10\%$.

Example #3



A resistor colored *White-Violet-Black* would be $97\ \Omega$ with a tolerance of $\pm 20\%$. When you see only three color bands on a resistor, you know that it is actually a 4-band code with a blank (20%) tolerance band.

Example #4



A resistor colored *Orange-Orange-Black-Brown-Violet* would be $3.3\text{ k}\Omega$ with a tolerance of $\pm 0.1\%$.