Introduction to Embedded Microcontrollers V4

Greg Reynolds Aug. 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 bush, 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 an 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 producing 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 ATmega328 microcontroller and a serial to USB chip that allows us to communicate with the microcontroller (See the User Manual in the information package).

The Nano and the switch are probably already mounted on the prototyping board.

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 the wires. A drawing below shows what holes have common channels.

The Switch

The switch is soldered to a very small printed circuit board to allow properly spaced pins to be securely inserted in the prototyping board.

The switch is probably already inserted into the proto board.

The LED

LEDs only work when the current flows in the right direction so correct insertion into the proto board is critical. The negative terminal is identified by a small flat portion on the body of the LED. Look carefully. You can usually catch the flat in the light if you turn the LED slowly in your fingers. I find it easier to closely examine the LED for its flat edge before inserting it, and then marking that side of the LED with a black felt tip pen. But then again, your eyes are probably better than mine.

The Resistor

A resistor must be put into line with the LED. The Nano drives too much current into the LED, abusing it. The resistor limits the current.

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 to 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

NOTE: Most of the Grade 12 computers will have on their D drives a directory called "Keep for Kirby". This directory has the Arduino IDE, Arduino directory and Notebook.

2. Most of the grade 12 computers have the necessary USB driver that the Nano needs. To confirm this, plug the Nano into an USB port. Open the PC's Control Panel, and go to Drivers and Printers. At the bottom, under Unspecified, should be USB-SERIAL CH340 (COMx). Note the COM number. This number could change each time the Nano is plugged in again.

Open the IDE. Go to ArduinoIDE/Arduino-1.6.4-windows/Arduino-1.6.4/Arduino.exe and double click. Under the pulldown menu File and open Preferences. Using the Browse button, change the Sketchbook location to the Arduino directory you have on the D drive.

Pull down the Tools menu. Change the Board type to Arduino Nano. Confirm that the Processor type is ATmega328. Change the Port to the COM number you found.

3. Pull down File, Examples, Basics, Blink. Confirm the connection by uploading (arrow button at the top of the IDE widow) Blink program (or Sketch as the Arduino community calls them, so as not to frighten the artists in the group) that makes the LED on the Nano blink. If you like, you can alter the blink rate by changing the lines DELAY(1000) to DELAY (500).

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

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. See the program SimpleToggle_PB_LED.

6. Carefully examine the operation of this program. It will not work in a reliable manner. Your task will be to rewrite the program to make it reliable.

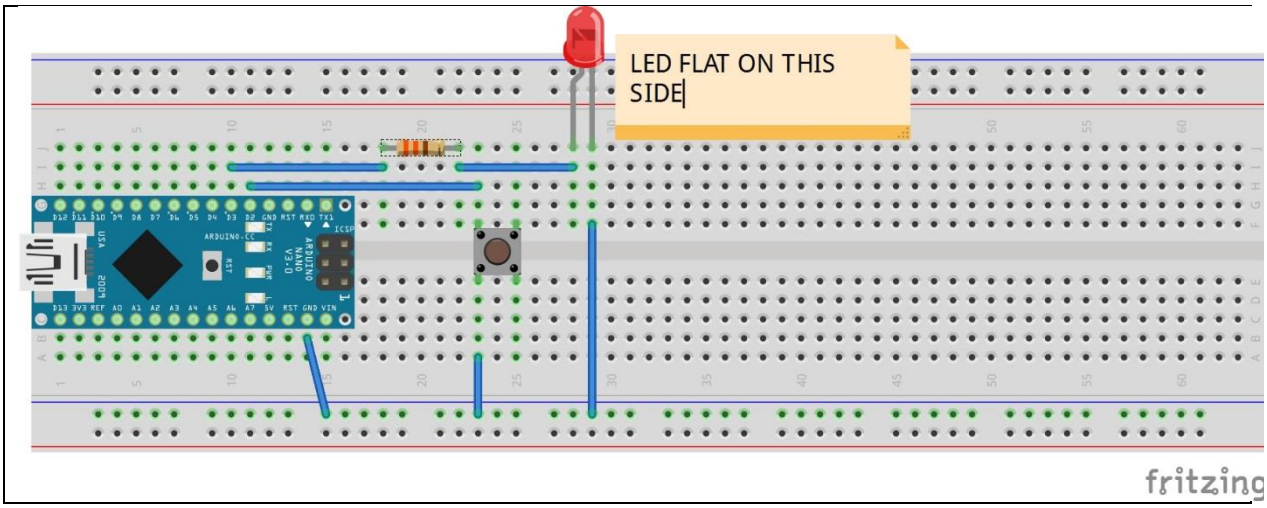
Terminology

Digital circuits represent the zeroes and ones of the digital world by voltage levels. Zeroes quite often are represented by 0 volts or the ground voltage. Ones by the voltage that powers the circuit. In this case, +5. This voltage is supplied to the Nano microprocessor through the USB connection from the PC. Quite often, engineers will interchange when talking between “+5” with “One” and “Ground” with “Zero”.

A Word about Programming

STEP AWAY FROM THE COMPUTER! Unless you are doing something fairly simple or you are gifted programmer you do not sit down and immediately start writing the code for a program. You give thought to what you want the processor to do and then express it in a flow diagram or pseudocode or maybe just random thoughts on paper. Then you start coding.

The following diagrams show the connections of protoboards, and the placement of components on the protoboards.



A Demonstration of the Serial Port on the Nano

The most useful technique used to debug programs is to put in a command that prints out the value of some significant variable in the program. You then use the Serial Monitor tool under Tools in the IDE to look at this variable. Page 29 of the Notebook describes the serial port commands.

Some of the students found this a useful way to better understand the millis() function (see page 27 in the Notebook) that you will need to complete this task. Write and load the following program into the Nano.

```
 void setup() {
  Serial.begin(9600); // Enable the serial port

}

void loop() {
  Serial.println(millis()); //Print out the value of the millis() function
  delay(1000); // Wait one second (1000 milliseconds)

}
```

As does all Arduino programs, the first function sets up the Nano configuration. In this case the serial port is turned on. Usually, the Nano starts running the program as soon as it is loaded. But when the serial port is turned on, the program does not run until the Serial Monitor is opened so that you can see all the output from the program from the very beginning.

The loop function loops continuously. It's the "motor" of the processor. It prints out the value of the millis() function once a second. Examining this output might help you understand what it does.