Source: C# Corner (www.c-sharpcorner.com) PRINT

Article

Controlling Fan Speed Using Arduino Through Microsoft Visual Studio

By **Kishore Chowdary** on **Feb 15 2017**

In my previous articles, I have explained about the concept of enabling the LED, creating home automation concepts, and so on. Now, in the following article, we are going to control the speed of the fan, using the PWM pins, using Arduino. Here, we are not going to use any external devices like relay or some other power reducing elements. We will just make use of the PWM pins of Arduino. These pins have the ability to take control over the power supply of the pins. Before we get into it, I need you to look at the following articles to get an idea about the IOT.

- Home Automation In Windows Forms Using Visual Studio
- Turning LED OFF And ON Through Voice Recognition using Microsoft Visual Studio
- Home Automation Using Arduino with Microsoft Visual Studio

PWM

Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip. External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in the value. See the attached Interrupt() function for the details. PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.

- *SPI*
  10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, using SPI library.

- *LED*
  13. There is a built-in LED driven by digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

- *TWI*
  A4 or SDA pin and A5 or SCL pin. Support TWI communication, using the wire library.

Items required

- A DC motor
- Arduino
- Connecting wires

Programming Arduino

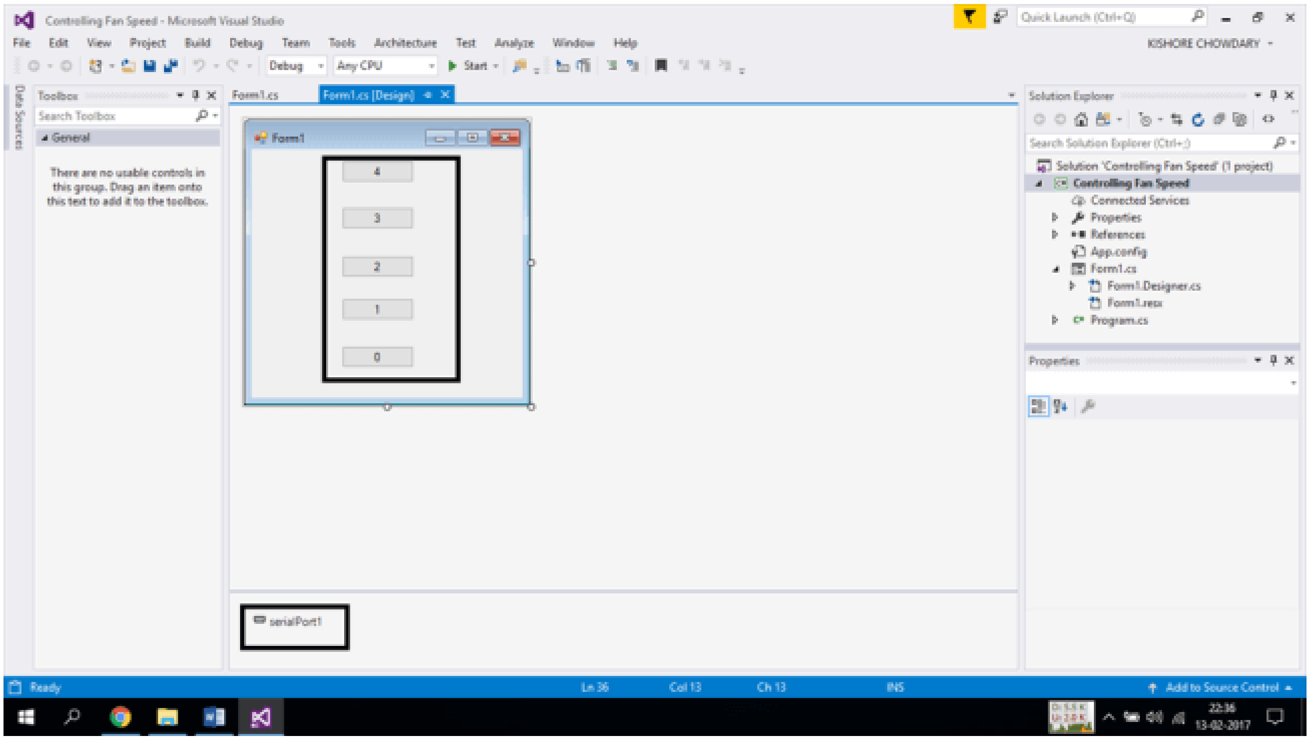
If you have come across the articles in the above links, you will be clear about the concept of uploading the program into Arduino. If not, please go through the articles first. If you know the concept of uploading the code, then upload the code given below.

```
char incomingdata;
```

```
void setup() {
    pinMode(9, OUTPUT);
    Serial.begin(9600);
}
void loop() {
    incomingdata = Serial.read(); {
        if (incomingdata == 'a') {
            analogWrite(9, 255);
        } else if (incomingdata == 'b') {
            analogWrite(9, 200);
        } else if (incomingdata == 'c') {
            analogWrite(9, 155);
        } else if (incomingdata == 'd') {
            analogWrite(9, 100);
        } else if (incomingdata == 'e') {
            analogWrite(9, -255);
        }
    }
}
```

Designing in Visual Studio

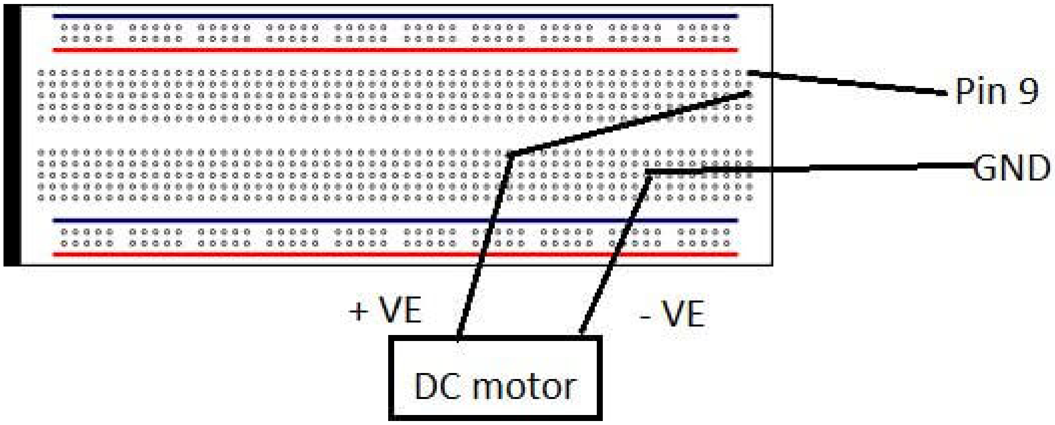
The next step is to design an Application in Visual Studio to send the commands to Arduino to execute the program. Open up your VS and create a new Windows Forms Application. Subsequently, add 4 buttons to the main page. Name them accordingly and then double click on them to add button click event handlers for them. Also, add a serial port tool from the tools box to communicate with Arduino. If you are very new to Arduino programming, using VS, you won’t understand all these. Thus, I suggest you read the articles from the links given above, if you are new. For each button, you will be using a separate acknowledgment, which will be sent to Arduino. Your design and code will be, as shown below.



```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
namespace Controlling_Fan_Speed {
    public partial class Form1: Form {
        public Form1() {
            InitializeComponent();
        }
        private void speed5_Click(object sender, EventArgs e) {
            serialPort1.Open();
            serialPort1.Write("a");
            serialPort1.Close();
        }
        private void speed4_Click(object sender, EventArgs e) {
            serialPort1.Open();
            serialPort1.Write("b");
            serialPort1.Close();
        }
        private void speed2_Click(object sender, EventArgs e) {
            serialPort1.Open();
            serialPort1.Write("c");
            serialPort1.Close();
        }
        private void speed1_Click(object sender, EventArgs e) {
            serialPort1.Open();
            serialPort1.Write("d");
            serialPort1.Close();
        }
        private void speed0_Click(object sender, EventArgs e) {
            serialPort1.Open();
            serialPort1.Write("e");
            serialPort1.Close();
        }
    }
}
```

Pin Connection

Connect the +ve end of DC motor to the pin number 9 and the negative end to the ground. Have a look over the image to get an idea.



Execution

The code which we wrote will make use of the PWM pins of Arduino. These pins will control the power supply, which is based on the cycle speed. Once you create the Application in your VS and upload your program into Arduino, run your Application code. By clicking on each button, you can control the speed of your fan.

This is how you can control the speed of your DC motor without using any external device. Comment out, if you have any queries. Hope, this will help you to learn something new about Arduino. Thank you. In comig days, I will present more interesting articles.