Theodor-Andrei POPESCU, Teodora COSTACHE

DESIGNING A WINDOWS PROGRAM FOR CONTROLLING DC-MOTORS USING MICROSOFT VISUAL STUDIO AND ARDUINO IDE

Abstract: *This case study wishes to highlight the advantages of using a Windows based PC to control a DC Motor for applications in small projects, the DC Motor being driven by an Arduino Uno board. Arduino controllers represent the best method of interfacing electrical components with a computer after programming either a Graphical User Interface or a Command Line Interface. The study will also show the advantages of using Visual Studio to design a GUI Windows application to control the mentioned DC-Motor. The purpose of this application is the future usage of the DC-Motor in more complex mechatronic projects. The application can adapt to any type of motor as long as it is not a Stepper-Motor or Servo-Motor.*

Keywords: *Arduino, windows, design, programming, interface, software*

1. INTRODUCTION

Since the invention of the transistor, important advancements in hardware and software development brought us to this moment, where interested users with little experience in programming, or without any electrical knowledge can create a revolutionary device. People can control their smart home from their wrist, while being at work overseas. We can imagine something simpler, like wirelessly control a slowly-spinning one-legged table from your phone or laptop at home, via Bluetooth, so that all the guests in the house can reach a particular dish. The table is rotated by a DC-Motor, which is controlled by an Arduino Uno, which communicates with the PC using a cheap Bluetooth adapter. This information will be clarified in the following study.

2. MAIN HARDWARE AND SOFTWARE

2.1 Arduino

Arduino represents an open-source electronics platform based on easy-to-use hardware and software. Open source products permit the user to freely use the content of the product, the source code or the documents. We can program Arduino boards with the company's own integrated development environment (which will be referred in text as IDE), which is easy to use and it is based on C++ programming language.

While there are many different types of Arduino models, we chose the Uno R3 board for this study, for it is widespread and easy to find (fig. 1). The microcontroller ATmega328P is the perfect chip for this project as it operates with 5 Volts, has 14 Digital and 6 Analog Pins [1].

Fig. 1 Arduino UNO R3 Mainboard [2]

2.2 What is Visual Studio?

Visual Studio is an IDE made by Microsoft. The application has grown significantly over the years and it's uses include: Windows based applications, websites, web-apps, web services and smartphone apps. Visual Studio supports 36 different programming languages. Built-in languages include C, C++, C++/CLI, Visual Basic, .NET< C#, F#, JavaScript, TypeScript, XML, XSLT, HTML and CSS [3].

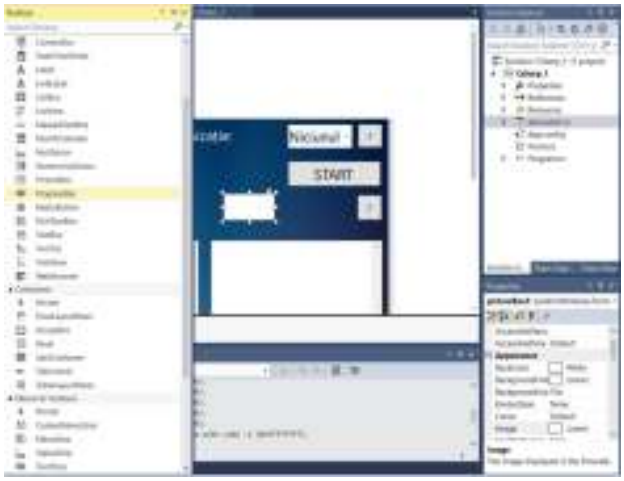

Fig. 2 Visual Studio's Design Editor

The main advantage of Visual Studio is that the programmer can see live updates in the IDE after compiling, without having the need to exit the IDE and run the application separately. As we can see in fig. 2, the user can drag the contents of the window just by dragging the mouse, while the program does all the arrangements automatically. This tool is very useful to create excellent software designs for products. After the design is completed, the user must program each element.

Another big point for Visual Studio is that the baseline software "Visual Studio Community Edition" is 100% free, making it the perfect beginner's IDE.

The main disadvantage of Visual Studio is the fact that the user interface can be pretty confusing at first and will need some "getting used to". The second main disadvantage is the fact that while the development team at Microsoft are pushing very frequent updates to the IDE, it still has its share of crashes, especially when a complex application it's being programmed. Users should always save after any significant changes to the code – this being a general advice for all computer users.

3. PURPOSE AND DESIGN

It's easy to imagine so many uses for this product. In this study, to make it even simpler, we will present an electrical assembly with a small, simple application. The electric motor we chose for this study is 36.5 mm long and with a body diameter of 21 mm, while its torque will be no greater than 0.2 Nm. This DC-Motor, that will be described from Chapter 3 and onwards, is already tested in lab. In this case, if the reader wants to get practical and build such an assembly, a more powerful motor will be required so that it will support the greater mass of the table. The rotation speed will be between 55 and 120 RPM. If lower speeds are required, a spur gear reducer can be easily designed. The design created will be theoretical with the DC-motor being a generic one.

The application for this contraption will be the slow-spinning one-legged table that the introduction specifies. mass of the table. The following 3D models are created entirely in Autodesk Inventor.

The following reference round one-legged dining table is designed for demonstrative purposes only (fig. 3). When designing anything, be it smart furniture or other projects, please have a complete set of calculations. The present study will focus more on designing the Visual Studio platform, Arduino software and assemble the electrical circuitry and not on creating a revolutionary dining table. We are aware that in this particular demonstration, the assembly might be too underpowered to spin the whole table, but a more powerful DC-Motor can be introduced for the assembly to work as stated.

Fig. 3 Referenced one-legged table modeled in Autodesk Inventor

At first glance, the table looks ordinary. The top is made of birch wood and the stand is made of a separate material. The foot has rubber pads to prevent easy falling, as seen in fig. 4.

Fig. 4 Rubber pads present on the table foot

If the reader pays attention, the holes present on the bottom of the table stand reveal the fact that the whole electric motor contraption is found inside of the table stand. If we change the transparency of the table stand the rotating shaft assembly is shown (fig. 5).

Fig. 5 Interior mobile shaft

If we zoom in, we will see the reducer assembly and the electric DC-Motor fixated with a metallic sleeve, to reduce vibrations.

Fig. 6 DC-Motor, Spur Gear Assembly

The red shafts hold the spur gears at the chosen distance, used to reduce the speed of the electric motor and multiply the torque applied to the mobile shaft. The DC-Motor is light-gray in fig. 6. To pair the motor with

the red shaft, a sleeve coupling is used; fig. 7 will show a sectioned view of it.

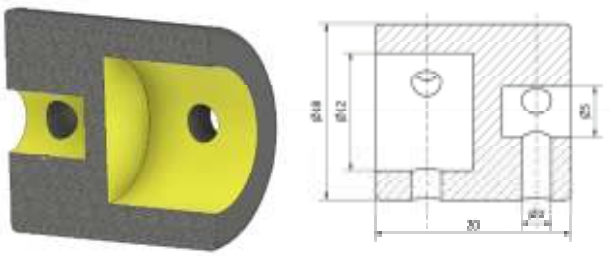

Fig. 7 DC-Motor-Red Shaft Coupling

The following calculations have been made, to fit the coupling on the shafts:

$$D \approx (1{,}4 \dots 1{,}8) * d = 1{,}8 * 5 = 18\ mm \quad (1)$$

$$l \approx (2 \dots 4) * d = 20\ mm \quad (2)$$

$$d_s \approx (0{,}25 \dots 0{,}3) * d = 0{,}25 * 12 = 3\ mm \quad (3)$$

The spur gear assembly was created with Autodesk Inventor's generator. The following sketch represents the reducer assembly (fig. 8):

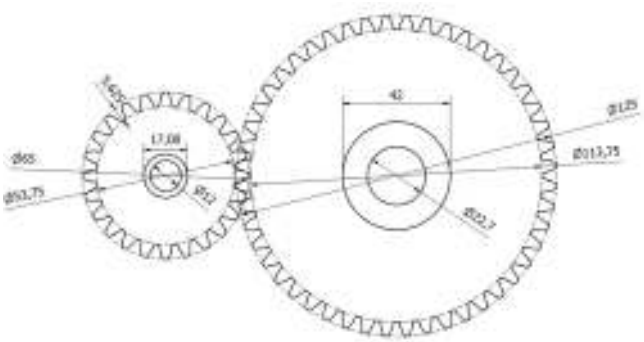

Fig. 8 Reducer Assembly

The big spur gear can roll easily with the help of a main shaft. The shaft is attached to the long rotating shaft of the table (fig. 9).

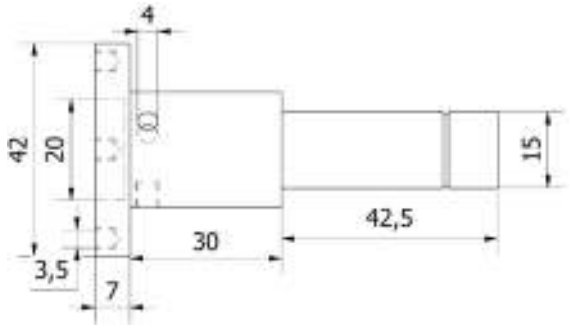

Fig. 9 Big spur gear shaft

The shaft can rotate freely with the help of two ball bearings.

For this example, the electronics box is placed separately (fig. 10-11), but a future project may include all components necessary to drive a DC-Motor by Bluetooth in a single contraption, including an AC/DC converter.

Fig. 10 Electronics Box

The electronics box will contain all the elements – except the DC-Motor – that will be described in Chapter 3. As stated, we need a connection to 12VDC. The red LED switch must be on before connecting to the table.

Fig. 11 Side-view of the electronics box

The design of the electronics box can easily exchange heat thanks to the CNC cut cooling grills, and can provide cable output for the motor without effort.

When the user turns on the switch, the serial port of the Arduino becomes open, explained in detail at Chapter 4. Now he has to connect the USB Cable (that comes out of the electronics box, through the motor output hole) to his PC, or wirelessly connect via Bluetooth.

The rest is easy to imagine, get your laptop out, click on START and get your favorite dishes closer, no effort. As stated earlier, the spinning table is a brute example, as the purpose of the study is to design the Visual Studio application and program the Arduino code. A very large number of practical approaches can be done with the wirelessly controlled DC-Motor: smart garage door opener (if no WiFi is present – otherwise this would be controlled with already existing methods), changing modular desk reach or height, medical care, mirror control, raise an OLED Television from a hidden position, electric curtains, etc.

4. ELECTRONIC ASSEMBLY

In the following figures we will show and explain each component used for the electrical assembly. To design and program the interface, we first need a working prototype. For this study, we chose a $0.2\ N*m$ HN-GH12-1632T-R 12V brushed DC-Motor with included reducer, shown in fig. 12.

Fig. 12 DC-Motor used in present study [4]

To drive this motor, we need a strong electronic driver. Initially we believed the L298N integrated circuit will do the trick, but unfortunately the driver heats up so much that even retailers sell it with a heatsink installed. Also, the seller recommends using it for stepper motor projects. The driver can be used for smaller motors with currents under 2A, shown in fig. 13, with the referenced link sending to a buy now page. Another advantage is that the driver is really cheap.

Fig. 13 L298N DC-Motor driver [5]

After researching online, we settled on an IRF540N Power MOSFET circuit (fig. 14). The metal–oxide–semiconductor field-effect transistor led to a revolution in electronics technology, being the first small transistor that can be used for many digital applications.

Fig. 14 IRF540N Power MOSFET [6]

This power MOSFET supports a continuous drain current of 33A at standard room temperature, dropping to 23A at 100 degrees Celsius. The datasheet convinced us to use this component for the project. The proper circuit schematic can be found in fig. 15. It includes a 1N4004 diode and a $10k\omega$ Resistor. D05 represents the fifth digital port on the Arduino board.

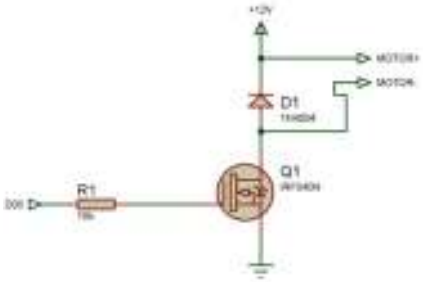

Fig. 15 IRF540N Circuit Diagram

To interface the motor with the Windows platform, we will need some type of serial connectivity, as this type is fully supported by the Arduino board. We settled on two types of connectivity to let users in a future project to choose: Bluetooth wireless communication and USB wired communication. The Arduino board has an integrated USB Type-B port, so all that remained was to find a Bluetooth module. HC-05 Bluetooth modules are widespread and very cheap [7]. They are also really easy to program and need only 3 Volts (seller recommends 5V) to function. The integrated circuit can be found in fig. 16.

Fig. 16 HC-05 Bluetooth IC

The Integrated Circuit has 5 pins: State, RXD, TXD, GND, VCC, Key. For this study we will only need four of them: Receive/Send (RXD, TXD) which will be connected on the same ports on Arduino, but inversed (RX Arduino port will connect to TXD and TX will connect to RXD). This is how we managed to make it work, but many other guides can be found online. The full schematic can be drawn with Fritzing, an open-source software used to make guides for electrical assembly. It is really easy to use and design circuits and it's free. For ease-of-use, we have also connected a reset switch on the circuit breadboard so that the reset button present on the Arduino Uno microcontroller will not have to be reached every time a new firmware is uploaded. Figure 17 shows the whole diagram.

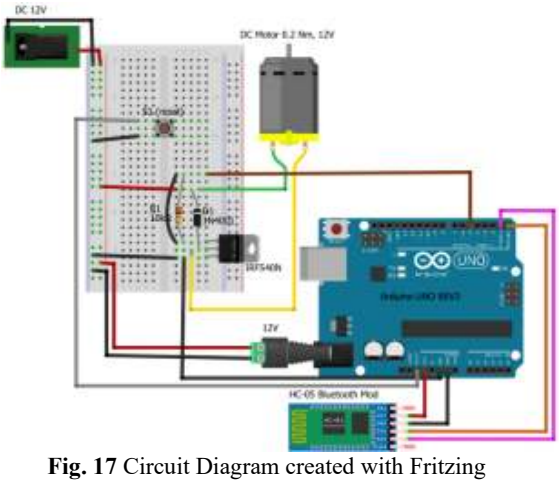

Fig. 17 Circuit Diagram created with Fritzing

5. DESIGNING THE SOFTWARE

As mentioned earlier, the software for driving the DC-Motor will be compiled for Windows based PC's, but can be easily adapted to work on an Android/iOS-based device with little modifications. The software that will be run on the Arduino is written and compiled in Arduino's IDE.

5.1 Arduino Software

We will start by telling the microcontroller what we have connected to its ports, in this case Digital no.5. From this pin a PWM-modulated signal will travel to the MOSFET gate for adjusting the motor speed.

```
#define Motor 5
```

The next step is to define the RX, TX ports which will start/stop the communication between the PC and the microcontroller.

```
#define FORWARD 1
 #define STOP 0
```

In the brackets of "void setup" every line of code will be run only once after the Arduino is restarted.

```
void setup() {…}
```

Serial communication is measured in pulses/sec or more commonly known as "baud rate", in this case we will set it to 9600.

```
Serial.begin(9600);
```

The last line in void setup should make the Arduino understand that the fifth digital pin is used for output, not input, so that the microcontroller can set the voltage required to activate the MOSFET, when the user wants to spin the motor.

```
pinMode(Motor, OUTPUT);
```

This is the last line in void setup(). We can now close the brackets.

Now, the Arduino has to wait for a start command from serial communication for an infinite time, (or until the user presses "stop" or disconnects DC power). Also, once the motor has started, we have to state that it has to run indefinitely. We also have to map values so that a slider on the software interface can set the motor spinning speed. Void loop () is the function that will run indefinitely after the setup.

```
void loop() {…}
```

Inside this function we have to tell the Arduino not to do anything until a signal from serial communication is online. The easiest way to accomplish this is to use the if() function. Inside the parenthesis we will write "Serial.available()" which will tell the Arduino exactly what is needed: "if serial is available, then"

```
If (Serial.available()) {…}
```

"Command" will be the variable that is read from serial (from the PC). In this case:

```
Int Command = Serial.read();
```

To not risk overloading the serial buffer and get lags or hang-ups, we will split the speed slider into 9 levels of speed (removing 0rpm, reasons of redundancy).

```
Int Speed = Command%10;
```

The Arduino Digital PWM pin 5 will output to the MOSFET values between 0 and 255 (in this case between 28 and 255), which transforms to values between 0 and 5 Volts, which the MOSFET can easily interpret. Using the "map" function, we transform the Speed variable that the PC is sending to the Arduino into a PWM variable for the MOSFET to interpret. This will optimize wireless serial communication.

```
Int PWM_Speed = map(Speed,1,9,28,255);
```

The last lines are comprised of a case loop, that will make the Arduino understand when to start spinning the motor and when to stop spinning it. For the case loop to work, we need to initialize another variable, RUN in this case.

Using the switch command, we will be able to set two cases, one in which the motor is turned off, and one in which the motor has the speed set by PWM_Speed. The syntax requires a "break;" line after each case.

```
int RUN;
switch (RUN)
 {
  case STOP: analogWrite(Motor, 0);
    break;
  case FORWARD: analogWrite(Motor, PWM_Speed);
    break;
 }
```

After this, the program is compiled successfully, and can be uploaded on the Arduino board. Now it's time to design and program the Windows Software.

5.2 Visual Studio Software

Fig. 18 DC-Motor Windows Application

After installing Visual Studio Community, we have to create a new Windows Form. Then, a design can be created using the Design Editor. The application UI resembles an industrial machine control panel, for this study, but it can be changed by dragging and dropping new elements or just rearranging content on the window

by dragging the mouse. The application is present in fig. 18: By double-clicking on any present element, the IDE enters the code editor, where element behavior can be changed. For example, double clicking the UPB logo will send to the following lines:

```
private void pictureBox2_Click(object sender, EventArgs e)
        {
System.Diagnostics.Process.Start("http://www.upb.ro/");
        }
```

This means that when the user clicks the UPB logo, the computer will redirect him to the specified website.

To make the motor spin, the user has to specify the port through which the computer communicates with Arduino, we'll take as an example COM1. The code is:

```
private void BLUETOOTHstartButton_Click(object sender,
EventArgs e)
        {
                if (serialPort.IsOpen)
                {
                    serialPort.Write(STOP, 0, 1);
                    serialPort.Close();
                    BLUETOOTHstartButton.Text = "Start";
                }
                else
                {
                    serialPort.Open();
                    serialPort.Write(FORWARD, 0, 1);
                    BLUETOOTHstartButton.Text = "Stop";

                }
        }
private void tBar_ValueChanged(object sender, EventArgs e)
        {
            SPEED_TMP = (byte)tBar.Value;
            FORWARD[0] =    (byte)(SPEED_TMP);
            if (serialPort.IsOpen)
            {
                serialPort.Write(FORWARD, 0, 1);
            }
            labelSpeed.Text = "Revolutions per minute:
[rot/min]";
        }
```

To explain this part of code: At first, when START button is clicked, check if serial port is open. If it is open, then show button as START and send "STOP" to Arduino to keep the motor from spinning. Otherwise, spin the motor by sending the chosen SPEED_TMP to Arduino and show STOP on the button. SPEED_TMP is the value of the dragging bar, which is sent to Arduino while the Serial port is opened. So, if the port is open (motor is spinning), send the chosen value on the bar (no. 3 at Fig. 18) to Arduino. The rotations per minute shown on the bar represent the mapped values on the Arduino code using the "map" function.

To show the available serial ports on the PC, a simple function is used: "`SerialPort.GetPortNames()`" – Gets an array of serial port names for the current computer. If the user clicks the "?" next to the port dropdown box, the following message is shown:

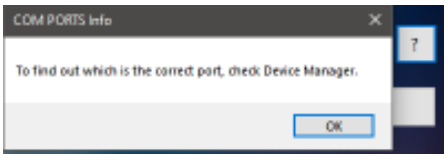

Fig. 19 COM PORTS Info

The correct port can be easily found after plugging the Arduino with a USB cable to the PC, or successfully pairing it via Bluetooth, accessing Device Manager and checking Ports (COM & LPT).

6. CONCLUSIONS

This article presented an easy to program software application to operate a DC-Motor in a cost-effective way, using easy-to-use and open-source software, for users with beginner's programming skills. The Integrated Development Environments used to design the platform were Arduino IDE and Visual Studio Community Edition in order to program an Arduino UNO R3 to send a digital signal, on demand, to an IRF540N MOSFET, set by the user using either USB Connection or Bluetooth to communicate with the electronic assembly.

The purpose of this electronic assembly is for using it as-is, by attaching a useful system on the DC-Motor's shaft, or for use in bigger projects, for a great variety of purpose.

REFERENCES

[1] *What is Arduino?* Available at: https://www.arduino.cc/en/guide/introduction, *Accessed:* 2019-11-11.

[2] *Arduino UNO R3, available at:* https://store.arduino.cc/arduino-uno-rev3, *Accessed:* 2019-10-10.

[3] *C99 library support in Visual Studio 2013,* available at: https://devblogs.microsoft.com/cppblog/c99-library-support-in-visual-studio-2013/ *Accessed:* 2019-11-18.

[4] *12VDC Reversible Metal Gear Head Motor, available at*: https://www.jameco.com/z/HN-GH12-1632T-R-M-12VDC-Reversible-Metal-Gear-Head-Motor-246mA_151442.html, *Accessed:* 2019-10-15.

[5] *L298N dc and stepper motor driver,* available at https://www.aliexpress.com/w/wholesale-l298n.html, *Accessed:* 2019-11-25.

[6] *IRF540N MOSFET Datasheet,* available at: https://media.digikey.com/pdf/Data%20Sheets/Fairchild%20PDFs/IRF540N.pdf, *Accessed:* 2019-11-12.

[7] *HC-05 Bluetooth modules,* available at: https://www.aliexpress.com/wholesale?SearchText=hc+05, *Accessed:* 2019-11-10.

Authors:

Popescu Theodor-Andrei, Eng., University Politehnica of Bucharest, E-mail:theodorandrei.popescu@yahoo.com **Assistant prof. Ph.D** student **Eng. Ioana Teodora COSTACHE**, University Politehnica of Bucharest, Department of Engineering Graphics and Industrial Design, E-mail: teodora.cos94@yahoo.ro