Source: http://educ8s.tv/c-application-arduino-communication/

Hello guys, I am Nick and welcome to educ8s.tv a channel that is all about DIY electronics projects with Arduino, Raspberry Pi, ESP8266 and other popular boards. Subscribe to the channel now if you don’t want to miss any future videos. In this video we are going learn how to build our own computer programs, in order to communicate with Arduino. I have built a simple project to demonstrate how easy it is to build your own Windows applications that can control an Arduino.

I have connected an LCD shield to an Arduino Mega, and I have also connected three LEDs to it. The application that I have developed connects to the Arduino using the Serial connection, and allow us to write some text to the LCD display and control the state of the LEDs. As you can see this is a very useful project if you need to send some data from your computer to an Arduino. Let’s see how to build this project.

The parts needed in order to build this project, are the following:

——————–

WHERE TO BUY

——————–

Arduino Mega: http://bit.ly/ArduinoMega

LCD Shield: http://bit.ly/ArduinoKeypadDisplay

LEDs: http://bit.ly/375LEDS

Resistors: http://bit.ly/ResistorsB

Small Breadboard: http://bit.ly/SmallBreadboard

3 in 1 wires: http://bit.ly/Wires3InOne

Power Bank: http://bit.ly/PowerBank\_XiaoMi

Full disclosure: All of the links above are affiliate links. I get a small percentage of each sale they generate. Thank you for your support!

First of all, let’s build the hardware part of the project. It is very easy to build since we are using a shield. We connect the LCD shield to the Arduino like this, and then we connect the LEDs. I am using pins, 19,20,21 for the LEDs. Connect a resistor to the long leg of each LED and the other leg to GND. You can find the schematic diagram of the project in the description of the video. If we program the Arduino, we can see, that the LCD display shows the message “Ready to connect”, which means the Arduino is programmed successfully and it is waiting for commands from the computer. Then we open the Computer application, we select the serial port that the Arduino is connected to and we press connect.

Now we can control the Arduino from the computer! How cool is that! Let’s now see how to develop the computer software to send commands to the Arduino.

The software that we are going to develop today, runs only on Windows, it won’t work on Linux or Mac. We are going to use Visual Studio 2017 community edition in order to develop the application. Visual Studio 2017 community edition is absolutely free. It is great development environment for Windows. With Visual Studio, you can literally build simple Windows Applications in minutes. First of all you have to download the installer, and install it on your machine. When prompted, select to install the .NET desktop development package and then press the install button. After the installation is complete you can launch the development environment.

Now you have two options, you can create a new project and build your own application from scratch using the C# programming language or you can download my project and modify it to achieve your desired results. Of course you want to just test the project and can just run the executable that you can find in the description of the video. If this is the first time you are using Visual Studio, please watch some tutorials on YouTube to help you getting started. Since Visual Studio is very popular you can find great tutorials online. I won’t show how I built this Windows Application in detail in this video but if you are interested I can prepare another very detailed tutorial on that. Are you interested in a detailed tutorial on how to build a Windows Application like this on Visual Studio? Please vote by clicking on the card here.

Let’s see how you can build your own Application. From the Menu on the Start Page we Select File-> New -> Project and we select to create a Visual C# -> Windows Classic Desktop -> Windows Forms App. We define a name for the Project and we press OK. That’s it a new project has been created and we can start to build our UI. As you can see, there are two tabs on the main project window. In the Form1.cs file we write our C# code. In the Form1.cs [Design] tab we design the User Interface of our Application. From the toolbox on the Left, we can select any tool we like and we can drag and drop it on our form. It is that easy. Each User Interface element has some Properties. We can modify them from this window. For example we can change the button text here. If we now press the Start button we can see our application running.

Let’s now open the ComputerToArduino project to see how it works. This is the UI design and this is the code of the project. Before we start let me explain you how this project works. I have created a simple communication protocol for this project which consists of 6 commands. Each command is 4 characters long and it can be followed by some data. Each command is starts with the # character and it is terminated by the \n symbol. This symbol will tell the Arduino that the command is complete so it can act. So we have the following commands:

STAR for Start

STOP

TEXT

LED1

LED2

LED3

The STAR command tells the Arduino to wait for commands from the computer.

The STOP command tells the Arduino that the computer has disconnected from it.

The TEXT command is followed by the text we want to display on the LCD. For example if we write #TEXTYOUTUBE\n on the serial port the word YOUTUBE will appear on the display.

The LED1, LED2 and LED3 commands control the state of each LED. So if we want to turn on the first LED we write the following command to the serial port: #LED1ON\n . If we want to turn off the second LED we send the command #LED2OF . The same thing applies to the third LED as well.

That’s the very basic protocol I use in this project. Of course you can design and use your own protocol which can use many more commands.

In the code now at first, after the User Interface is initialized we disable all the controls. Then we get the available serial ports on the computer and we load them on the combox. When we press the Connect button, the button1\_Click method is called and we try to connect to Arduino if we are not already connected. In order to connect to Arduino, we open the Serial port, and we send a command to it. We simply write #STAR\n to the serial port. Then we change the button text and we enable all the controls on the user interface. Now if we write a text on the textbox and press the write button, we get the text from the textbox and write it to the Serial port with the command TEXT. Also, when the state one of the checkboxes changes we send the command LED1 for example with the state, on or off. Lastly, when we press the connect button and we are already connected to Arduino we send the STOP command.

Now let’s take a quick look at the Arduino code. At first we initialize the display. Then we wait for serial data. When we receive data on the serial port we save them in the variable inputString until we receive the \n character. When we receive this character we know that command is complete, so we have to act. If we receive the TEXT command for example we know that we have to print something on the LCD display. So, we remove the first five characters from the inputString variable, because that’s the command, we delete then last 2 character which is the \n symbol and we print the remaining characters on the display. We do something similar when we receive an LED command. We read the next two characters after the command to see if we need to turn the LED on or off and we act accordingly. As you can see the Arduino code is very simple, we just use so if statements to respond to the commands sent by the computer. As always you can find the code of the project, both the Arduino Code and the C# code in the description of the video below.

From this demonstration project you can see that it is relatively easy to build your own Windows applications to control an Arduino. If you learn the basics of C# and Visual Studio you can build impressive projects. You can build your own computer peripherals! There are so many projects that we can build around a computer. I have already a project in mind that I am going to build soon. It is going to build an external Arduino computer monitor which will display the CPU usage, memory usage and so on in real time. I think this is going to be cool. I would love to hear your ideas as well. Do you find Computer to Arduino communication useful and if so what kind of projects are you going to build? Please post your comments in the comments section below and don’t forget to like the video and share it with your friends if you find it interesting!