

NodeMCU RC Car

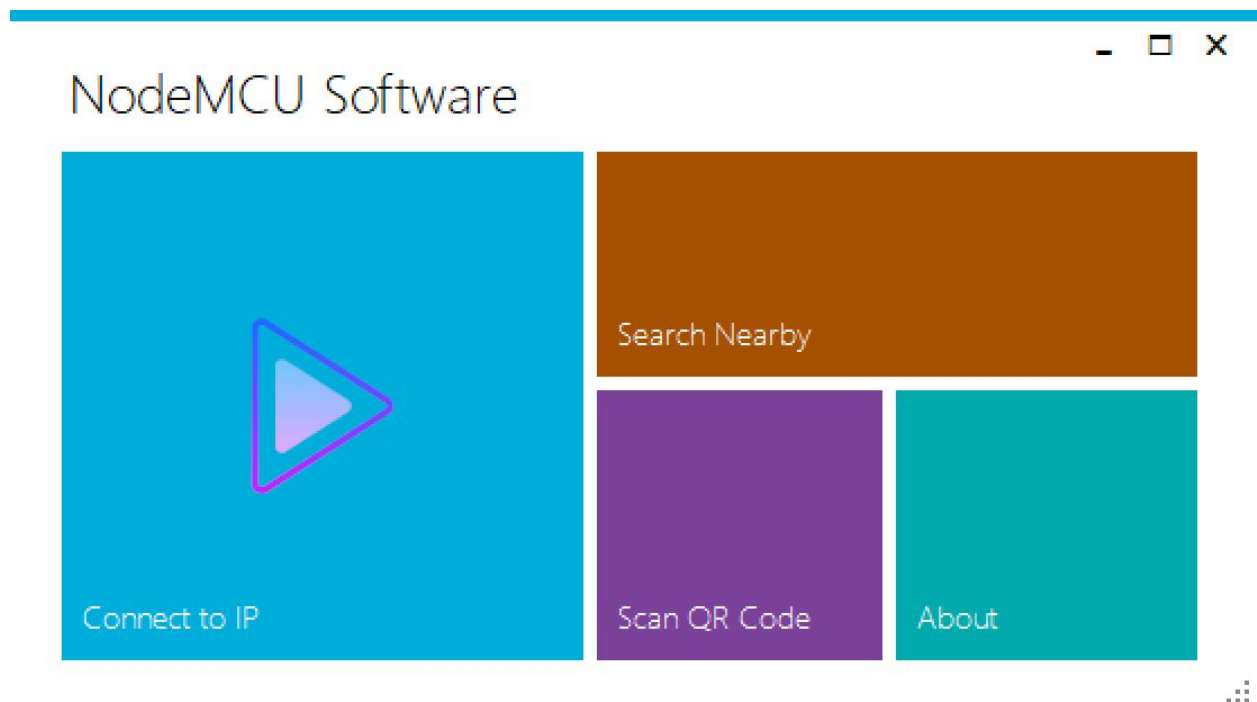
Software Usage Guide

ShangJin Li (SYEP)
shli001@uw.edu

Introduction

In this tutorial, we will go over the usage of the NodeMCU softwares. We start with a brief description of what each software does and proceed to demonstrate their basic functionalities.

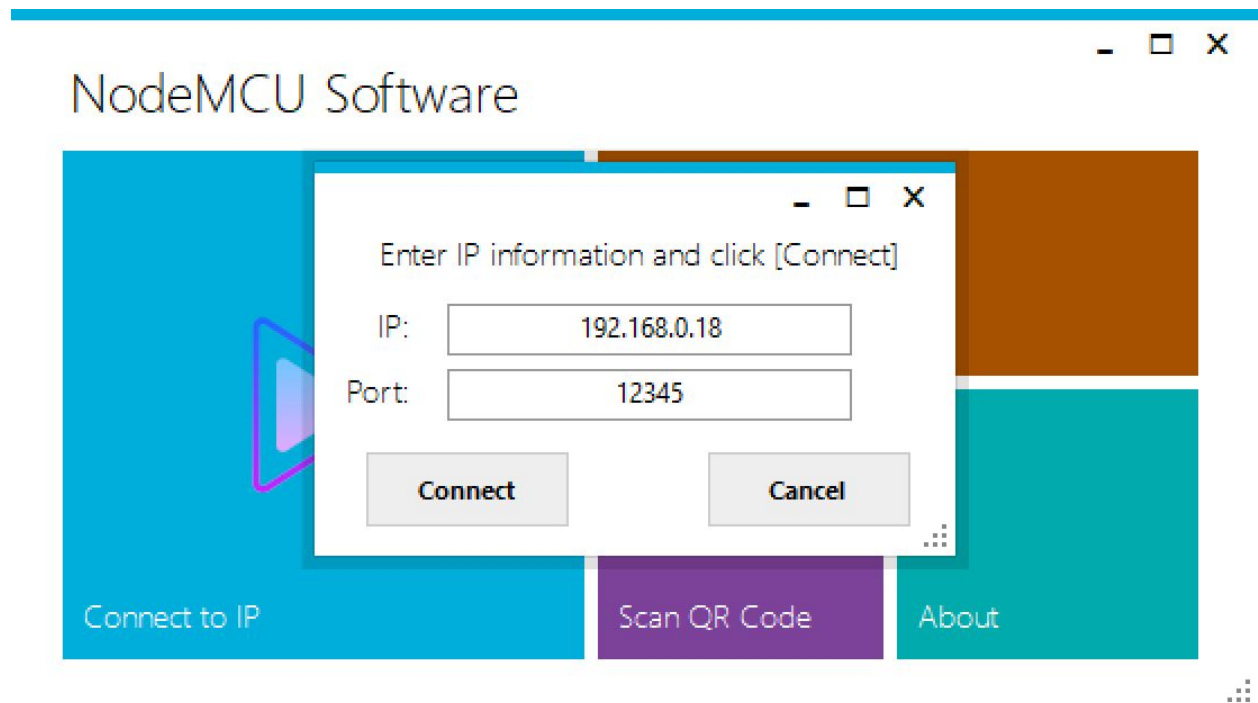
Part 1: C# Application



Overview

The NodeMCU Software provides a user-interface for controlling the ESP8266 RC car over the network. It features both “Connect by IP” and “Search Nearby” as options for establishing a connection.

Usage (Connect by IP)



When you click on the big, blue tile labeled “Connect to IP,” you should see this box pop up asking for inputs. IP stands for IPv4 address, and port stands for the networking port of which a WiFiServer is listening on a remote device (RC car). You may find these information (IP & port) from the Arduino source code uploaded to the RC car, and the Serial output printed when the car has connected to WiFi. They must match exactly in order for a successful connection to be established. If you are ready, click “Connect” to establish a connection.

NodeMCU Controller



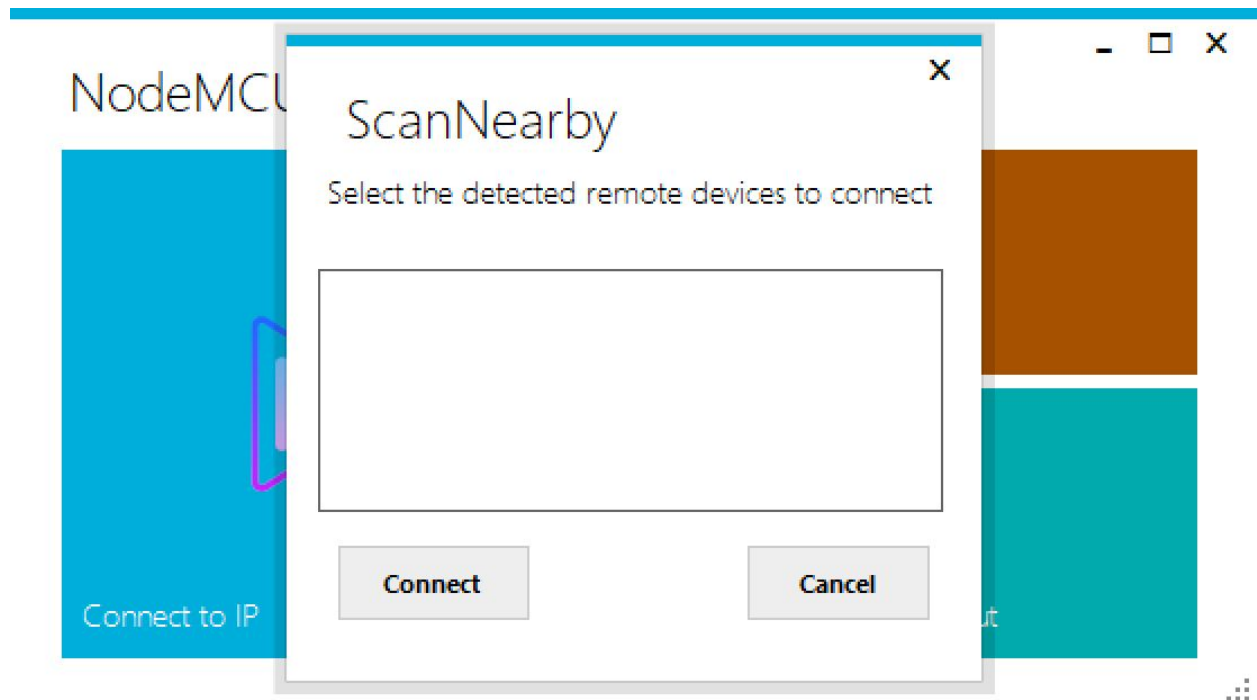
After you have clicked “Connect,” a window that looks like this should pop up. Let us focus our attention on the lower-right corner, a label called “Status.” If your label does not say “Connected,” you should close this window and double check your IP and port. A “Disconnected” status implies the remote device is either 1) offline, or 2) does not exist. You must be connected to the ESP8266 module before you get to control it!

By now you should have the status labeled “Connected.” Try to press on your keyboard for the keys shown (W, A, S, D)¹, and observe the movement of your car as it responds to your instructions. Be aware that when you press on key A and D, the car might not move at all. This is due to the fact that you must specify a direction, either forward or backward, for a turn to be considered valid. In other words, you must press on either W or S when you press A or D for the corresponding turn instructions to be executed.

After you have finished controlling the car, you may disconnect via the “Disconnect” button, or simply close the program in its entirety. Your car will come to a stop after you have disconnected.

¹ The [F] key is not included because horn is not installed in the basic version of this car. You may, however, use the [F] key to enable the horn after you have installed the hardware and updated the Arduino code (see `remote_handler_improved.ino`) for the horn.

Usage (Scan Nearby)



The “Scan Nearby” feature is provided as a compensation for users who do not know the IP address of their ESP8266 car. After clicking on the rectangular, brown tile labeled “Scan Nearby,” a window like this should pop up, listing each and every IP address in the WiFi environment that have address near the computer that is running the software. Note that it is *not guaranteed* that any online devices be detected by this feature because Ipv4 addresses are sometimes unpredictable due to the fact that modem configuration could affect the distribution of Ipv4 addresses for devices connected to the same WiFi. In case your device (ESP8266) does show up, simply double-click the entry to highlight it, and then press “Connect.” The rest will be the same as that demonstrated in the “Connect by IP” part of this tutorial.

Part 2: Android App

Download



Mirror



After you have installed the .apk file for the app and run it, you should see this window.

NodeMCU

10.133.0.75

12345

CONNECT

SAMSUNG

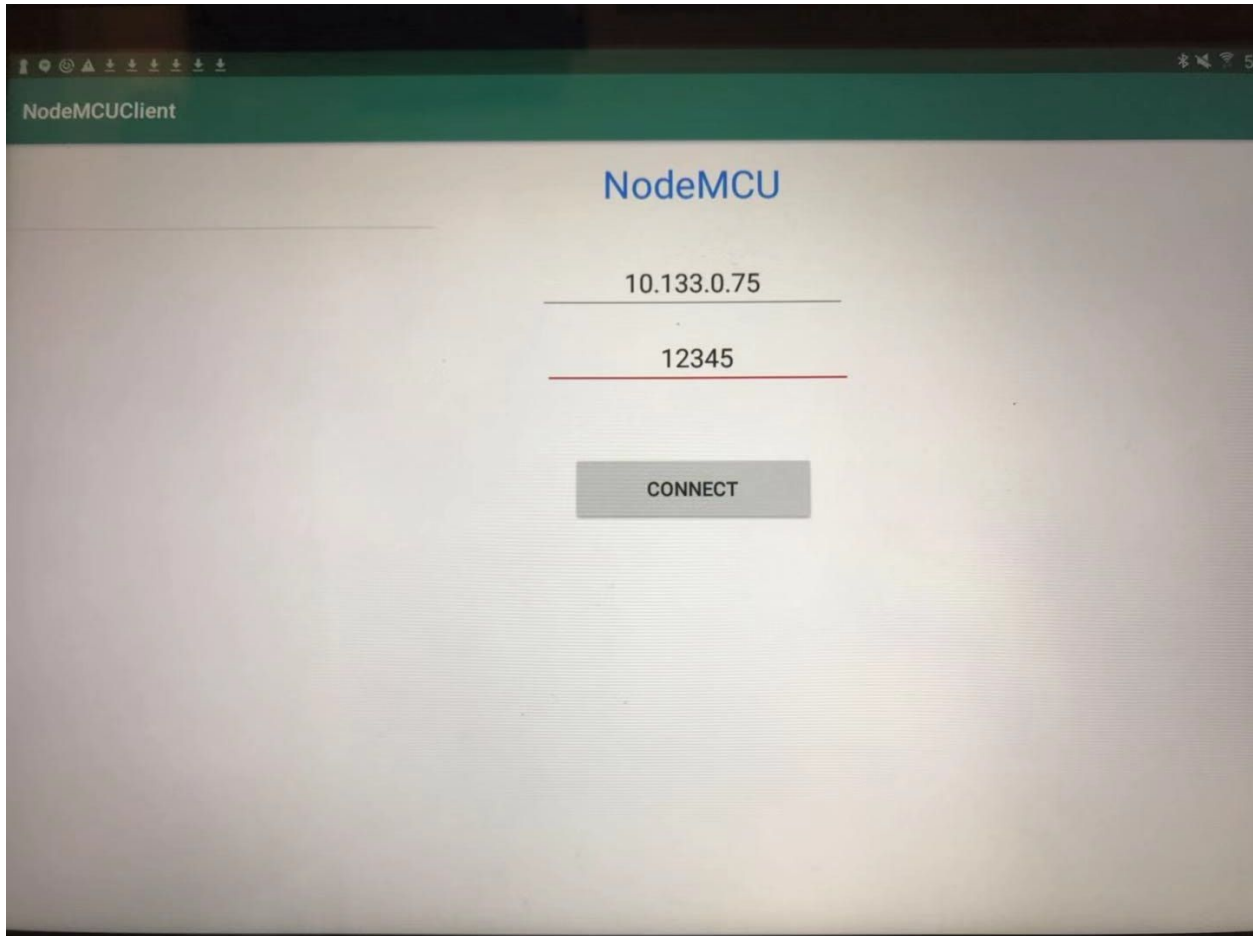
Text View



Alt

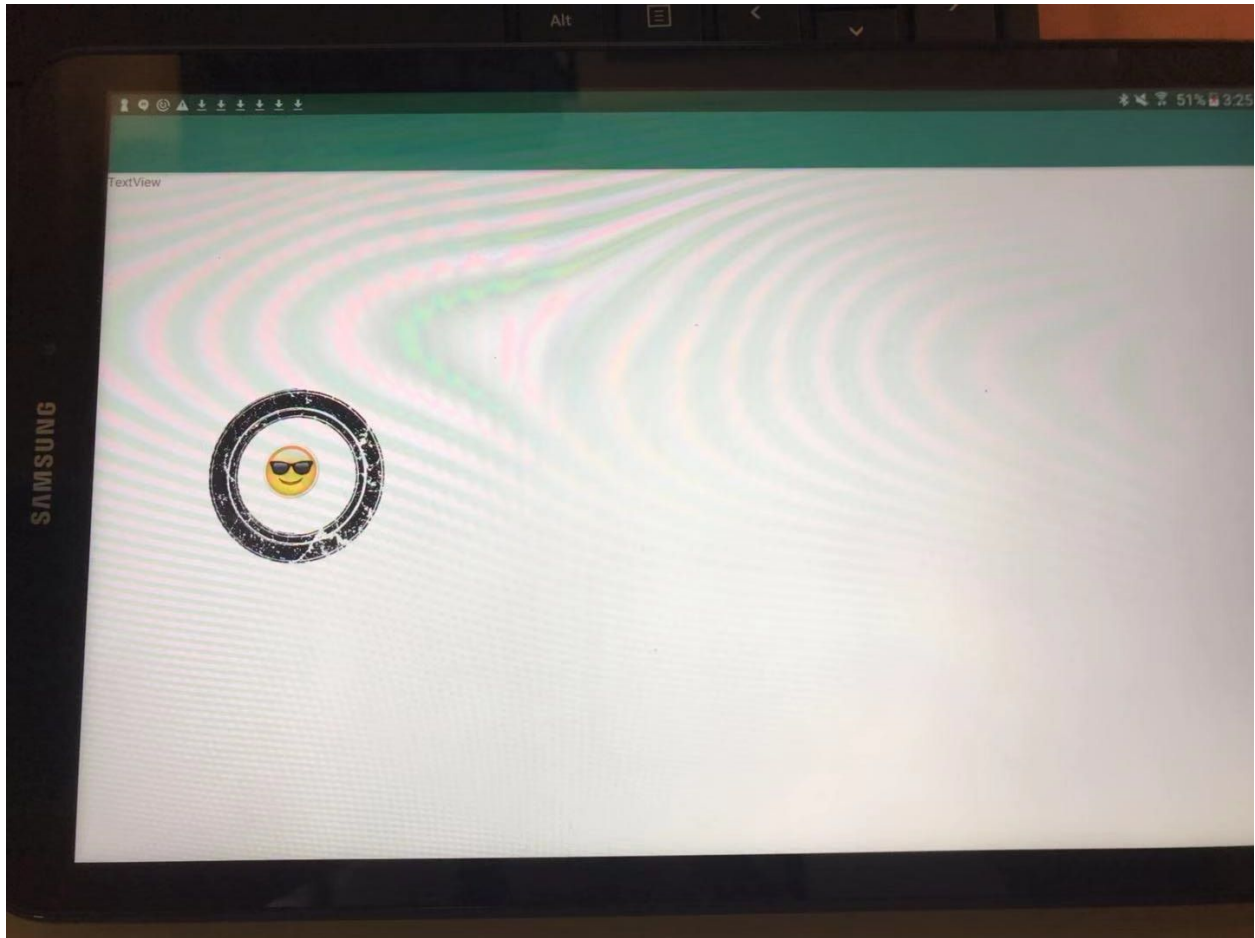


51% 3/25



Enter the IP address and port number for your ESP8266 car, then press “Connect.” After that, you should see this window pop up. The black circle with an emoji icon is the virtual joystick that you may use to control the car! Try holding the emoji icon and push it forward, backward, left, and right to see how the car responds to the corresponding instructions.

If your car is not moving, you may need to go back to the IP & port window by clicking on the return button of your Android device, and verify that the IP address and port number is correct.



Summary

We have gone over the usage of two different applications that can both control the ESP8266 car. If you wish to learn more about the development of such applications, check out the [NodeMCU Software Implementation Guide](#) where we will go over the structure and implementation for the two applications!