ESP-IR

This document describes the realization of an universal remote control based on an ESP8266 WiFi module.

The ESP8266 is a remarkable thing. It has a Wi-Fi interface and a powerful processor with enough memory to store a complex application. ESP_IR universal remote control described here uses a ESP-12 and an optional RF433 sender to control TV/DVD/lightning.

Features:

- Simulates one or more IR remote controllers for TV/DVD/Interactive TV.
- Simulate RF sender to control lights in living room.
- Uses a minimal number of components.
- Can be controlled by a tablet or other device through a build-in webserver.
- Pictures of original remote controls are used to control the ESP-IR.
- The strongest available WiFi network is automatically selected. Passwords are kept in the SPIFFS filesystem.
- Heavily commented source code, easy to add extra functionality.
- Debug information through serial output.
- Update of software over WiFi (OTA).

Software:

The software for the ESP-IR is supplied as an Arduino sketch that can be compiled for the ESP8266 in de Arduino IDE version 1.6.8, esp8266 software 2.2.0. No Arduino is required in this project.
The following libraries are used:

- ESP8266WiFi – for establishing the communication with WiFI
- ESPAsyncWebServer – for remote controlling the radio via http.
- ESPAsyncTCP – Needed for webserver.
- ArduinoOTA for software update over WiFi.
- IRremoteESP8266 for driving the IR leds.

The webpage is written in HTML using JavaScript.

Webpage:

The webpage that will be supplied by the ESP8266 webserver shows a page like this:

The 3 remote controls are scans of the originals. The buttons are HTML areas that react on a mouse click. The power button for the Samsung DVD-player is defined like:

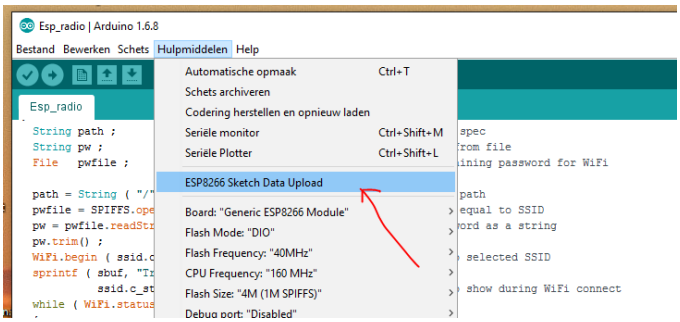
<area shape="rect" coords="16, 22, 46, 50" onclick="smklik('power')" href="#">

“smklik” refers to a JavaScript function that converts the area to a IR code and sends the code to te ESP8266.

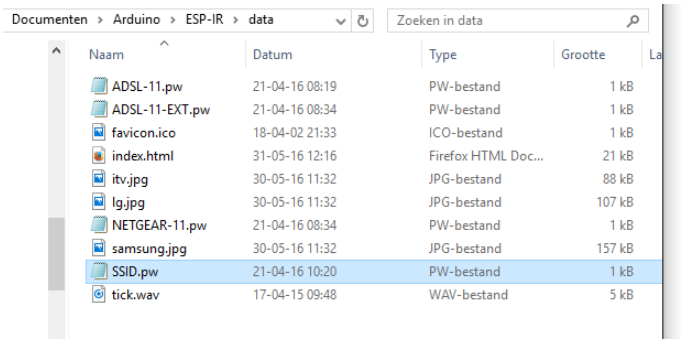
The 10 white buttons at the bottom are the switches that control the living room lights.

Configuration:

In order to work properly, the software needs some configuration.
The filesystem (SPIFFS, set to 1 MB) of the ESP8266 must contain the files necessary for the webserver and a password file for every acceptable network SSID. The plug-in for Data Upload must be present in your Arduino IDE:



If you don't see this feature, check the installation guide at the end of this document.
The files must be present in a map "data" in the Arduino project map, For Example:



Note: If only one password file (*.pw) is present in the filesystem, the radio will connect to that SSID. This is very useful when using hidden SSID.

.pw file(s)

A ".pw" file must be present in the filesystem for every WiFi network you intend to connect to. The content is in readable format, editable by any text editor like "Notepad".
Example: Assume your SSID is "PS4" and your password is "welcome123". In this case you must prepare a file with the name "PS4.pw" that contains the string "welcome123". The file SSID.pw in my Github repository is an example. Rename this file according to your SSID and edit the first line in it.

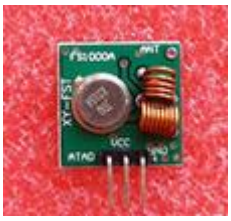
.jpg file(s)

The ".jpg" files hold the images of the remote control(s).

Hardware:

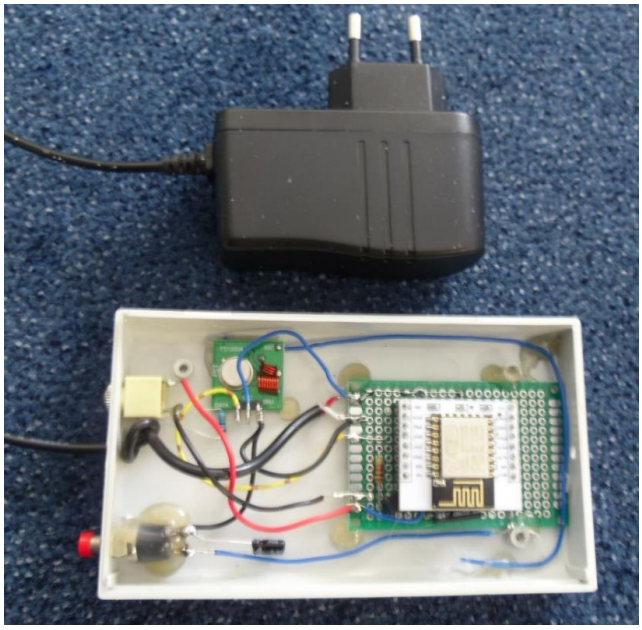
The ESP-IR is built with the following hardware:

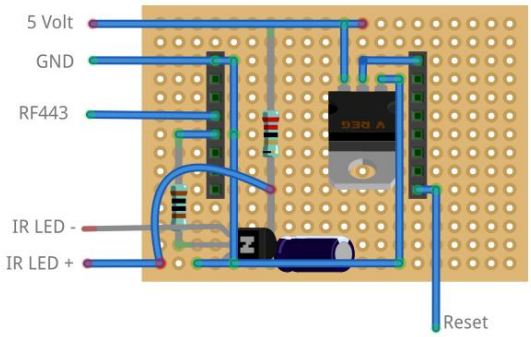
- An ESP-12 module. This is basically an ESP8266 on a small print. There is pull-up on CH_PD and a pull-down on GPI015. If the ESP8266 is used without this module you have to provide at least the pull-up to set the ESP8266 to work. The ESP8266 is running on 80 MHz.
- A RF433 transmitter to control living room lights.
- A 3.3 volt LDO to provide power for the ESP8266.
- 1 to 3 IR LEDs. I used 3 in series and mounted them close to the TV/DVD/Interactive TV box.
- An NPN transistor like BC237 to drive the IR LEDs.



The ESP-IR is powered by a 5 V adapter. For the ESP8266 a small regulator (LD1117S33TR), 3.3 Volt 800 mA is used.

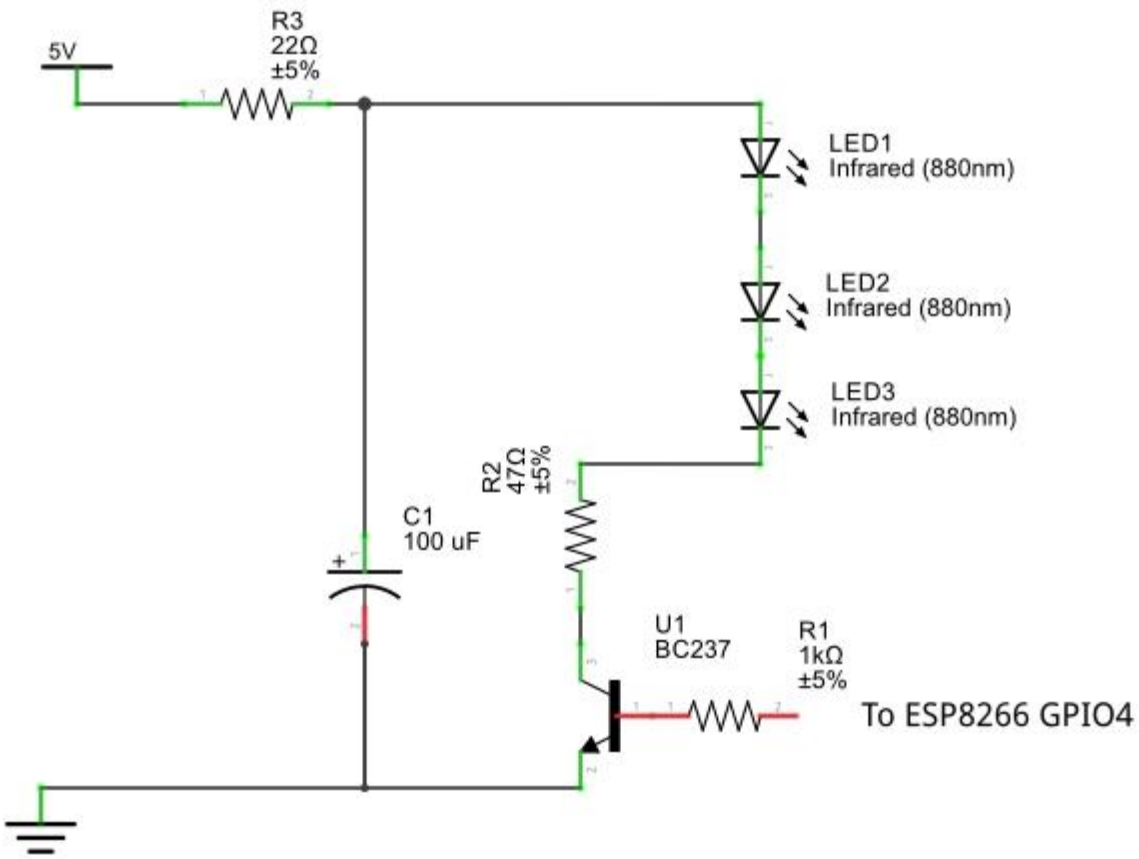
I used a small perforated board to mount the ESP8266 module.





Schematic diagram:

The IR LEDs are driven by a BC237 transistor. The circuit is like this:



Wiring:

The logic wiring as shown in the diagram is also presented in the table below. The analog amplifier and the speakers are not included.

NodeMCU	GPIO	Pin to program	Connect to
D1	GPIO0	0	433 MHz module
D2	GPIO4	4	IR leds driver

Web interface:

The basic idea is to have a html page with embedded javascript that displays the interface to ESP-IR as shown on page 2. Command to ESP-IR can be sent to the http server on the ESP8266 by clicking the buttons.

Capabilities of the webserver:

Let's assume that the IP of the Esp-radio is 192.168.2.12. From your browser you can show a simple root page by entering the following URL: http://192.168.2.12. This will display the /index.html file from the SPIFFS as well as /favicon.ico.

If your computer is configured for mDNS, you can also use http://ESP-IR.local in your browser.

Changing remote controls:

If you want to replace one or more controls, you have to take the following steps:

- Make a scan of the original remote and save it as a .jpg file. Mind that the resolution may not be very high because of the limited space in the SPIFFS.
- Edit the index.html in order to load to load the new .jpg file.
- Edit in index.html the area's for every button on the remote. I used "Paint" to get the position of the buttons.
- Edit the tables with the IR-codes for the remote in index.html.
- Edit the sketch for the new remote.

This is all not so easy to do, but you will end with a very nice ESP-IR WiFi controller and you can throw away all your old controllers.

Arduino IDE installation for ESP-IR.

- Download Windows installer for Arduino version 1.6.8 from https://www.arduino.cc/download_handler.php?f=/arduino-1.6.8-windows.exe and start the installation.
- Start Arduino and open Preferences window.
- Enter `http://arduino.esp8266.com/stable/package_esp8266com_index.json` into *Additional Board Manager URLs* field. You can add multiple URLs, separating them with commas.
- Open Boards Manager from Tools > Board menu and install *esp8266* platform (and don't forget to select your ESP8266 board from Tools > Board menu after installation).
- Download "Async TCP Library for ESP8266 Arduino" from https://github.com/me-no-dev/ESPAsyncTCP and install in the IDE (add .zip library).
- Download "Async Web Server for ESP8266 Arduino" from https://github.com/me-no-dev/ESPAsyncWebServer and install in the IDE (add .zip library).
- Download "IRremoteESP8266 library" from https://github.com/sebastienwarin/IRremoteESP8266 and install in the IDE (add .zip library).
- Load the sketch. You should be able to compile it.
- Install Python 2.7 for Windows. Select option "Add python.exe to Path".
- Download the tool: https://github.com/esp8266/arduino-esp8266fs-plugin/releases/download/0.2.0/ESP8266FS-0.2.0.zip and unpack it in your Arduino sketchbook directory, create `tools` directory if it doesn't exist yet. The path will look like `<home_dir>/Arduino/tools/ESP8266FS/tool/esp8266fs.jar`.
- Restart Arduino IDE
- Update boards.txt according to https://github.com/esp8266/Arduino/blob/master/boards.txt

Debug (serial 115600 Baud) output.

This is an example of the debug output.

```
D: FS Total 957314, used 392815
D: /ADSL-11-EXT.pw                  -     12
D: /ADSL-11.pw                      -     12
D: /favicon.ico                     -    894
D: /index.html                      -  21341
D: /itv.jpg                         -  90084
D: /lg.jpg                          - 108601
D: /NETGEAR-11.pw                   -     12
D: /samsung.jpg                     - 160228
D: /SSID.pw                         -     73
D: /tick.wav                        -   4616
D: Starting ESP Version 31-may-2016...  Free memory 40104
D: Sketch size 291992, free size 2850816
D: * Scan Networks *
D: Number of available networks: 6
D:  1 - NETGEAR-11                Signal: -64 dBm Encryption WPA2  Acceptable
D:  2 - Private_FON               Signal: -53 dBm Encryption WPA
D:  3 - FON_Roulet_11             Signal: -54 dBm Encryption None
D:  4 - ADSL-11                   Signal: -84 dBm Encryption Auto  Acceptable
D:  5 - Ziggo0A484                Signal: -81 dBm Encryption WPA2
D:  6 - Ziggo                     Signal: -78 dBm Encryption ????
D: -----------------------------------------------------------------
D: Selected network: NETGEAR-11
D: Try WiFi NETGEAR-11
D: IP = 192.168.2.16
D: Start server for commands
```