# ANALYZING WIFI NETWORKS USING ELECTRONIC HARDWARE DEVICES

**POPA SORIN EUGEN [1*], PRUTEANU EUSEBIU [1]**

[1] *“Vasile Alecsandri” University of Bacau, Calea Marasesti 156, Bacau, 600115, Romania*

**Abstract:** The paper presents the programming and development of a portable device for detecting and analyzing wireless networks. It is able to determine the main parameters of an access point, that is, to be able to set our own access point to achieve maximum performance in wireless data transfer.

**Keywords:** wifi networks, .

## 1. INTRODUCTION

Nowadays Wi-Fi networks have a widespread spread. Almost all electronic devices have the ability to connect to a Wi-Fi network, starting with laptops and continuing with mobile phones, tablets, smart clocks and more. More recently, all kinds of modules and development placards have been created that allow us to connect to any Wi-Fi any other electronic device or a wide range of sensors and transducers to transmit the measured values over the Internet.

Consequently, the ether around us is very crowded with 2.4 GHz and 5 GHz signals, resulting in the phenomenon of interference and distortion of radio signals, hence a Wi-Fi connection with much lower performance than standard.

Thus, various techniques and methods have emerged that allow us to measure the performance of wireless networks and to set up network access points accordingly so that we achieve maximum connection performance.

## 2. PROBLEM ANALYSIS

In all 802.11 network variants, the maximum pass-through power is given either based on ideal measurements or on OSI layer 2 transfer rate. However, this does not apply to typical applications in which data is transferred between two endpoints, at least one of which is usually connected to a wired infrastructure and the other endpoint is connected to an infrastructure through a terminal, a wireless connection.

This means that data frames typically go through an 802.11 (WLAN) environment and are converted to 802.3 (Ethernet) or vice versa. Due to the difference between the frame lengths (headers) of these two environments, the size of the application package determines the data transfer speed. This means that applications that use small packages (for example, VoIP) create data streams with high-speed traffic (ie, good low value). Other factors that contribute to the overall rate of application data are the speed at which the application transmits the packets (i.e., the data rate) and, of course, the energy with which the wireless signal is received. The latter is determined by the distance and the output power of the communication devices.

[*] Corresponding author, email sorinpopa@ub.ro

## 3. DESIGN AND IMPLEMENTATION OF THE NEURAL NETWORK

In principle, any device that comes with a wireless network card can be turned into a wireless network analyzer with the right software application. So any PC, laptop, mobile phone that has a wireles receiver and a processor / microcontroller that can run a software code may be able to scan the ether to determine if we have access points in the adjacent area.

To make a portable wifi analyzer, we opted for the following technical solution: a NodeMCU WiFi module with ESP8266, connected to a 2.4 "LCD with SPI. They also add an external battery for mobile phones and wires.

The ESP8266 WiFi module is a SOC with integrated TCP / IP protocol that can give it access to the WiFi network of any microcontroller. ESP8266 is capable of hosting an application or taking over all Wi-Fi network functions from another application processor. Each module comes pre-programmed, with firmware with AT commands.

This module has a powerful enough processing and storage capability that allows it to be integrated with specific sensors and devices through GPIO pins. ESP8266 supports APSD for VoIP applications and Bluetooth interfaces and contains a self-calibrated RF that allows it to work in all operating conditions and does not require external RF tracks.

All components were connected according to the general electrical scheme of Figure 1.

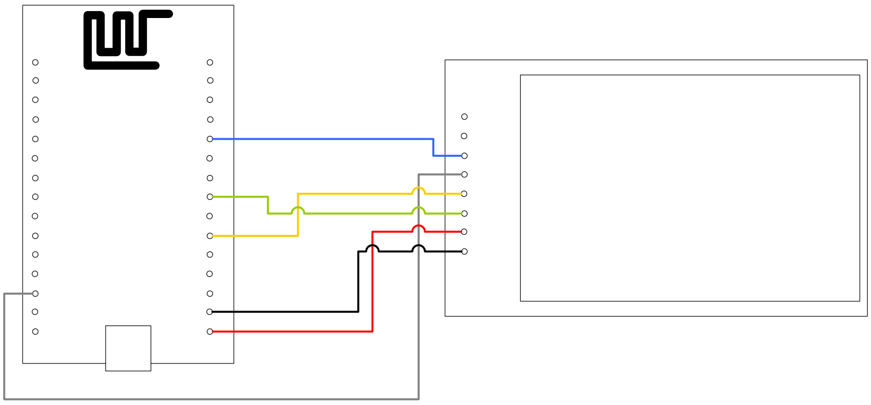


Fig 1: Electrical connection diagram of the ESP8266 module to the TFT LCD screen.

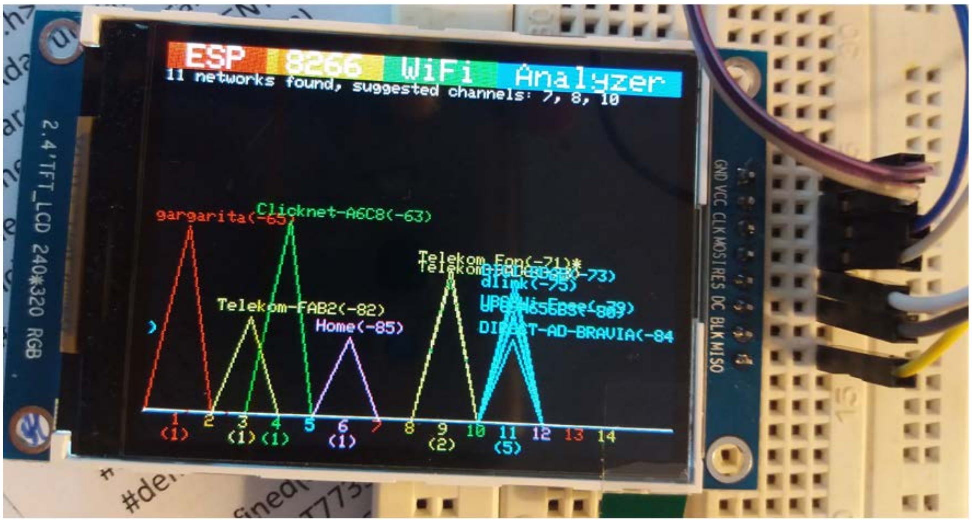


Fig. 2: Wi-Fi networks scanned by the device created with the ESP8266 module and the TFT LCD.

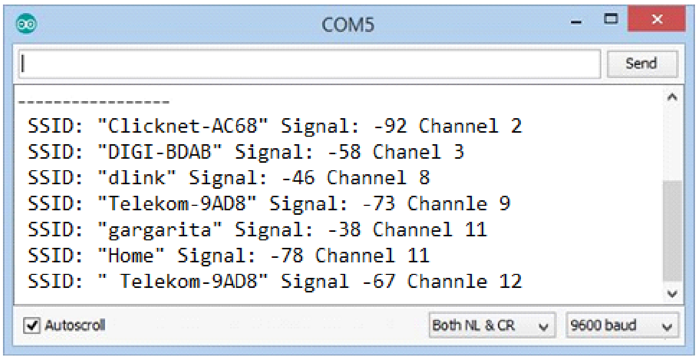
**The setup()**

The content of the block in setup () is explained.

- Start the tft.begin () display and set the text orientation tft.setRotation (3).
- Then, the following message appears on the screen: "ESP 8266 WiFi Analyzer", each word being written in a different color and on a different background.
- Set ESP mode in station mode and disconnect from Access Point if previously connected: WiFi.mode (WIFI_STA); WiFi.disconnect ();
- Enter a delay for execution time and then start the serial communication at 115,200 bps, sending the message that the settings were made.

**The loop ()**

- Defines an ap_count string that will count how many access points are on each of the 14 channels. The number of AP / channel is initialized with zero; uint8_t ap_count [] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
- Defines for each channel the RSSI signal to the maximum value -100; int32_t max_rssi [] = {-100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100};
- Network scanning starts;
- The whole n stores the number of WiFi networks found; int n = WiFi.scanNetworks ();
- Deleting the old graph on the screen:
- If n == 0, the message "no network found" is displayed, ie no network was found;
- Otherwise, graphically plot information about WiFi networks;
- Displays SSID, signal strength and if unencrypted;
- Displays the status of WiFi networks: the number of networks found and the remaining channels;
- Draw the axle of the chart;
- The following messages are displayed on the serial (to be seen in text and on the computer) as shown below:



```
-----------------
 SSID: "Clicknet-AC68" Signal: -92 Channel 2
 SSID: "DIGI-BDAB" Signal: -58 Chanel 3
 SSID: "dlink" Signal: -46 Channel 8
 SSID: "Telekom-9AD8" Signal: -73 Channle 9
 SSID: "gargarita" Signal: -38 Channel 11
 SSID: "Home" Signal: -78 Channel 11
 SSID: " Telekom-9AD8" Signal -67 Channle 12
```

Fig 3: Results sent on the serial post.

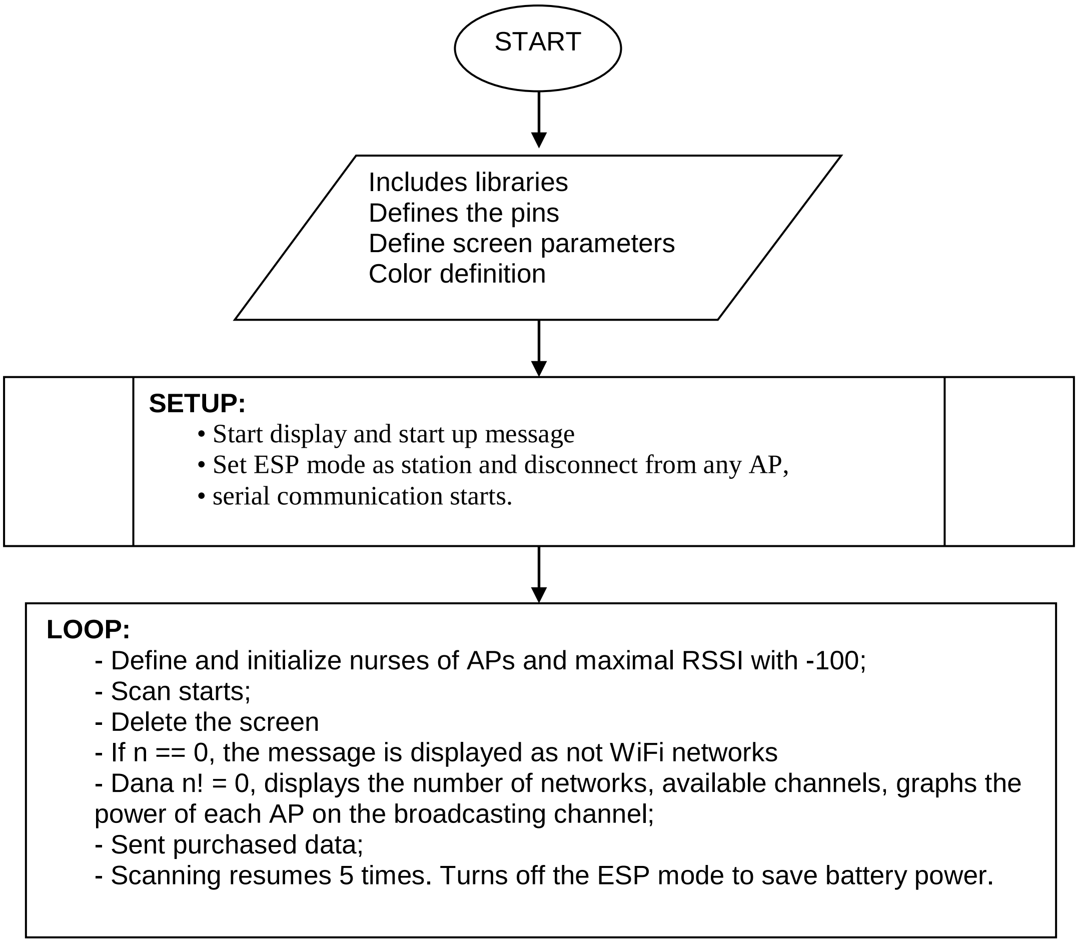

START
Includes libraries
Defines the pins
Define screen parameters
Color definition
SETUP:
• Start display and start up message
• Set ESP mode as station and disconnect from any AP,
• serial communication starts.
LOOP:
- Define and initialize nurses of APs and maximal RSSI with -100;
- Scan starts;
- Delete the screen
- If n == 0, the message is displayed as not WiFi networks
- Dana n! = 0, displays the number of networks, available channels, graphs the power of each AP on the broadcasting channel;
- Sent purchased data;
- Scanning resumes 5 times. Turns off the ESP mode to save battery power.

Fig. 4: The logical scheme of the Arduino program.

## 4. CONCLUSIONS

This paper succeeds in fulfilling the objectives of the paper, namely the development of a dispatch for the analysis of Wireless networks.

The chosen solution uses state-of-the-art equipment and techniques, and requires poor programming of microprocessor programming, C ++ programming language, and hardware used.

The device uses a NodeMCU Wi-Fi module with ESP8266 that can be programmed with Arduino IDE in C ++ and which transmits the data to a 2.4 inch color display that has a resolution of 240x320 pixels.
All of this is powered by an external battery.

## 5. REFERENCES

[1] http://www.instructables.com/id/Using-the-ESP8266-module/
[2] https://learn.adafruit.com/adafruit-huzzah-esp8266-breakout/using-arduino-ide
[3] http://www.komodolabs.com/product-downloads/
[4] https://electronicsforu.com/electronics-projects/hardware-diy/arduino-esp8266-wireless-web-server
[5] https://nib.ro/05/22/monitorizare-si-optimizare-retele-wireless-cu-inssider/
[6] https://www.binefa.cat/php/dam/m09uf3/doc/esp8266wifi-library.pdf
[7] https://alselectro.wordpress.com/2016/11/29/esp8266-wifi-library-on-arduino-ide/
[8] https://cdn-learn.adafruit.com/downloads/pdf/adafruit-gfx-graphics-library.pdf