Full Wireless Communication Device

Project documentation.

Caio Dutra

May 2023

Contents

Preliminaries

i Project overviewls

The purpose of this project was to develop a firmware solution as part of a job interview for the position of Firmware Engineer at Tractian. The project involved designing and implementing hardware and firmware for data transmissions through a wireless protocol. For this task, the WiFi protocol was chosen. Although it wasn't necessary, it was added to the project the capability of data collection, so there was actual information to be sent later. The device was equipped with a temperature sensor to capture data, which was then stored in a flash EEPROM memory.

ii Project objectives

The main objective of this project is to demonstrate the capabilities of a battery-powered device by successfully transmitting a 500kB file across a challenging 100-meter open-air gap. This task requires efficient wireless communication and optimized power management to ensure reliable and uninterrupted data transmission. By accomplishing this objective, the project aims to showcase the device's ability to overcome distance limitations and deliver data wirelessly, even in challenging environments.

1 Introduction

This project aimed on the full development of a wireless communicating capable device, from scratch all the way up to the final design. Throughout the development process, careful consideration was given to hardware requirements and specifications. The firmware was optimized to ensure accurate temperature data collection and efficient data transmission, while minimizing resource utilization and power consumption. The firmware development encompassed various aspects, including the design of the firmware architecture, algorithm implementation, and integration of communication protocols.

The following sections of this documentation provide detailed insights into the project's hardware and firmware requirements, operation flow, device usage, limitations, and potential future improvements. The schematics and PCB layout diagrams offer a comprehensive view of the hardware design, while the firmware section provides an in-depth analysis of the software implementation.

1.1 Hardware Requirements

The chosen wireless protocol for this project was WiFi, and so, the ESP8266EX microcontroller was selected for its robustness, affordability, widespread acceptance in the firmware market and strong community support. Additionally, this MCU has the capability to work with RF signals at power levels of up to +20dBm in its TX, and sensitivities of -91dBm and -75dBm for 11Mbps and 54Mbps on its RX, respectively. Based on these values, under ideal conditions, signals can travel up to 300m. However, it is important to validate the project concept in real-world scenarios, where noisy environments and obstacles are almost always present.

For the temperature sensor, the TMP35 was chosen due to its low power consumption, ability to be powered off when not in use, good accuracy, and similarity to other widely used components in the market, such as the LM35.

As for the flash EEPROM memories, the MX25R3235F (32Mbit) and AT25SF081 (8Mbit) chips were selected for program and collected data storage, respectively. These ICs were chosen because they meet the size requirements (minimum 2MB for program and 500kB for data) and operate under the serial SPI communication protocol.

1.2 Firmware Requirements

For the firmware, the development was chosen to be carried out using the Arduino IDE, as it works with C++, a language accepted by the chosen microcontroller. Additionally, this IDE provides pre-developed board configurations, which can be helpful during the prototyping phase. Moreover, it makes firmware uploading easier since we can utilize ready-made modules for these tasks, such as the well-known ESP8266 NodeMCU 1.0.

For the WiFi protocol, the "ESP8266WiFi.h" library is used, licensed under the LGPL (Lesser General Public License) Version 2.1. In broad terms, the license can be interpreted as follows:

> *"Primarily used for software libraries, the GNU LGPL requires that derived works be licensed under the same license,but works that only link to it do not fall under this restriction."*

The candidate chose to use this library because they do not have solid knowledge of the WiFi protocol (from the physical layer to the application layer) to develop their own library or subroutines. A full copy of the LGPL Ver 2.1 can be found at [4].

For the communication between the microcontroller and the Flash Data Memory, the SPI communication protocol is used. Also, the it was implemented an algorithm for a linear time-invariant system, namely the Moving Average Filter. Such algorithm is required to ensure that the measured temperature remains in a stable state when it is saved in the flash data memory integrated into the hardware.

1.3 Operation Flow

The final device workflow is simple: it stabilizes the reading/collecting of temperature data, stores it on the flash EEPROM, and then opens communication with a access point, access a localhost webserver url and transmits the collected data. Finally, it erases the last $512,000$ samples of temperature and starts the process again.

1.4 Limitations and Future Improvements

To reduce the code complexity, it was assumed that the server the board communicates with via the Access Point (AP) is a "localhost," and therefore, no SSL-HTTPS security configurations were implemented. Additionally, no ACK verification is performed to ensure that the server has indeed received the transmitted data. Furthermore, no HTTP POST messages containing SQL commands for writing to a hypothetical database are implemented. It is assumed that this set of business rules would be on the server side. Also, to avoid further increasing the code length (which has already been extended due to an unnecessary feature - reading temperature with precision and storing it in memory), no routines were implemented to ensure that the data was successfully written to the PCB memory. Finally, the device does not have any interface; it is simply plug and play. If there were any specific requirements, such an interface could be added subsequently.

2 Circuitry

2.1 Block Diagram

The block diagram of the device on the Figure 1 consists of several main components that play essential roles in its operation. Here is a brief overview of each of these parts:

- Microcontroller (ESP8266EX): The ESP8266EX microcontroller is the central component of the device. It is responsible for controlling all operations and functionalities, including wireless communication via WiFi.
- Program Memory (MX25R3235F): The program memory is where the device's firmware is stored.
- Data Memory (AT25SF081): The data memory is used to store the collected data in the device.
- Temperature Sensor (TMP35): The TMP35 temperature sensor measures the ambient temperature.
- RF Antenna (Texas SWRA117D 2.4GHz - Left): The RF antenna plays a crucial role in the wireless communication of the device. It enables the transmission and reception of radio signals, allowing WiFi communication. The RF antenna ensures a stable and reliable connection between the device and the access point (AP), facilitating the transmission of collected data. More information on this type of PCB antenna can be found at [2].

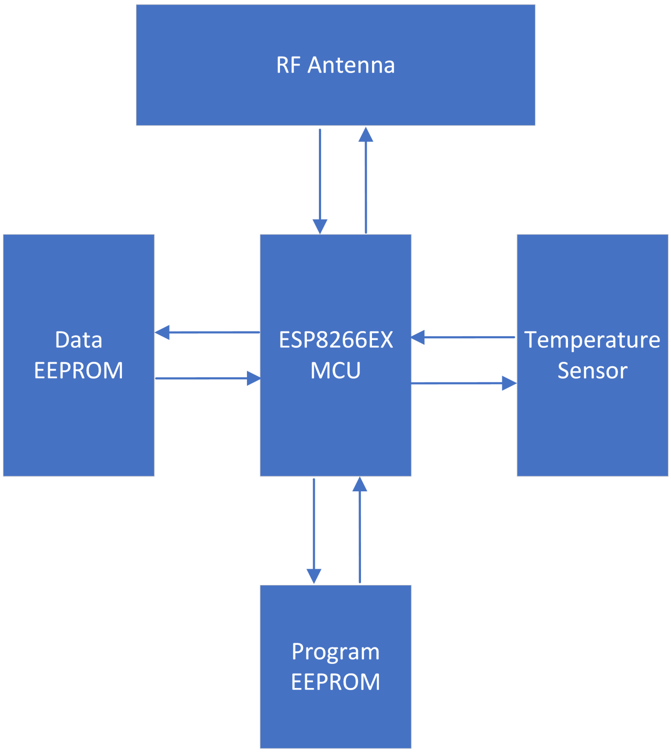

Figure 1: Device's block diagram.

2.2 Schematics

To further delve into the circuit's details, on Figure 2 is the schematic diagram of the circuit. It is also divided into four blocks, each serving a specific purpose in the device's functionality as follows:

- Power Supply Block: The power supply block provides a stable +3.3V voltage to the entire circuit. It ensures that all components receive the necessary power to operate efficiently and reliably. The power supply block includes a 3.7V (mAh to be decidade, depending on the longevity needed) and a voltage regulators. Decoupling capacitors and other necessary components to regulate and distribute power are cointained in the "Main Circuitry" (see Figure 2).

- Program and Data Memory Block: This block encompasses the program and data memories of the device. The program memory, represented by the MX25R3235F chip, stores the firmware responsible for controlling the device's operations. The data memory, represented by the AT25SF081 chip, stores collected data for future transmissions. This block includes the necessary connections between the microcontroller and the memory chips.

- Temperature Sensing Block: The temperature sensing block includes the temperature sensor component (TMP35) and its associated circuitry. The sensor measures the ambient temperature and then converts it temperature reading into a eletctrical analog format for processing by the microcontroller. This block is responsible for accurate temperature measurement and data acquisition.

- Main Block (Microcontroller and Components): The main block contains the microcontroller (ESP8266EX) and its associated components, in addition to the 2.4GHz RF antenna. The microcontroller serves as the central processing unit of the device, executing firmware instructions and coordinating various operations. This block includes components such as crystal oscillators, capacitors, resistors and inductors; these are the supporting elements necessary for the microcontroller's functionality.

These four blocks work together to ensure the device's proper operation. The power supply block provides the necessary voltage, the program, and data memory block stores and retrieves essential information, the temperature sensing block enables temperature measurement. Last but not least, the main block with the microcontroller controls and manages the device's overall functionality.

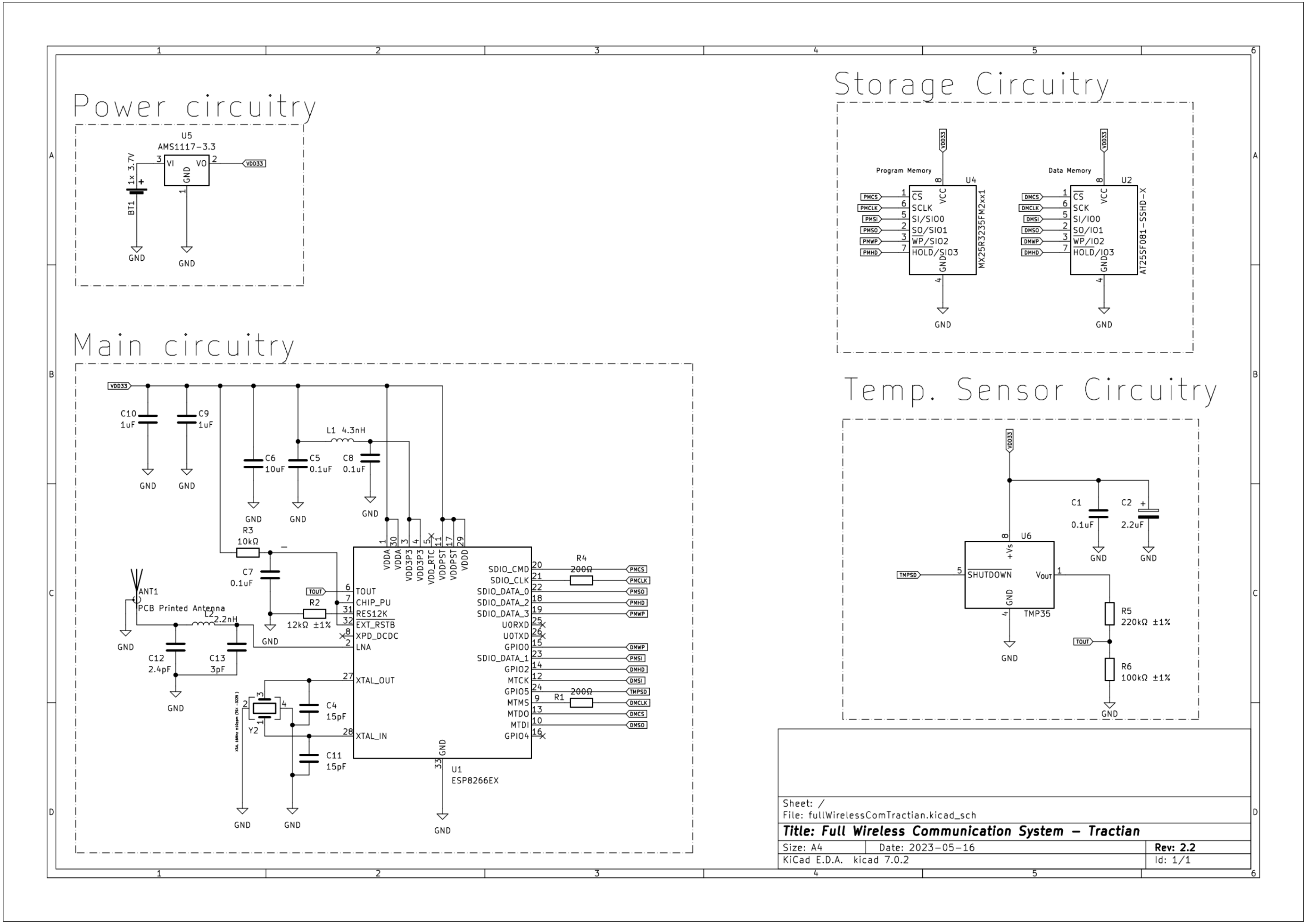

Figure 2: Circuit Schematics.

2.3 Circuit Diagrams

We start highlitning the main points of the Main Circuitry, putting important design requirements for the properly function of the ESP8266EX.

2.3.1 Digital and Analog Supplies Circuitry

For the analog power supply, ESP8266EX features five analog pins dedicated to power supply, encompassing Pin1, Pin3, and Pin4, which specifically cater to the power requirements of the internal PA and LNA. Furthermore, Pin29 and Pin30 serve as power inputs for the internal PLL. The acceptable voltage range for these analog power supply pins spans from 2.5 V to 3.6 V [3].

As for the digital power suplly, ESP8266EX is equipped with two digital pins, namely Pin11 and Pin17, dedicated to power supply. When it comes to the digital power supply, the inclusion of supplementary filter capacitors is unnecessary. The operational voltage range for the digital power supply pins spans from 1.8 V to 3.3 V [3] Both Analog and Digital power supply connections can be seen on Figure 3

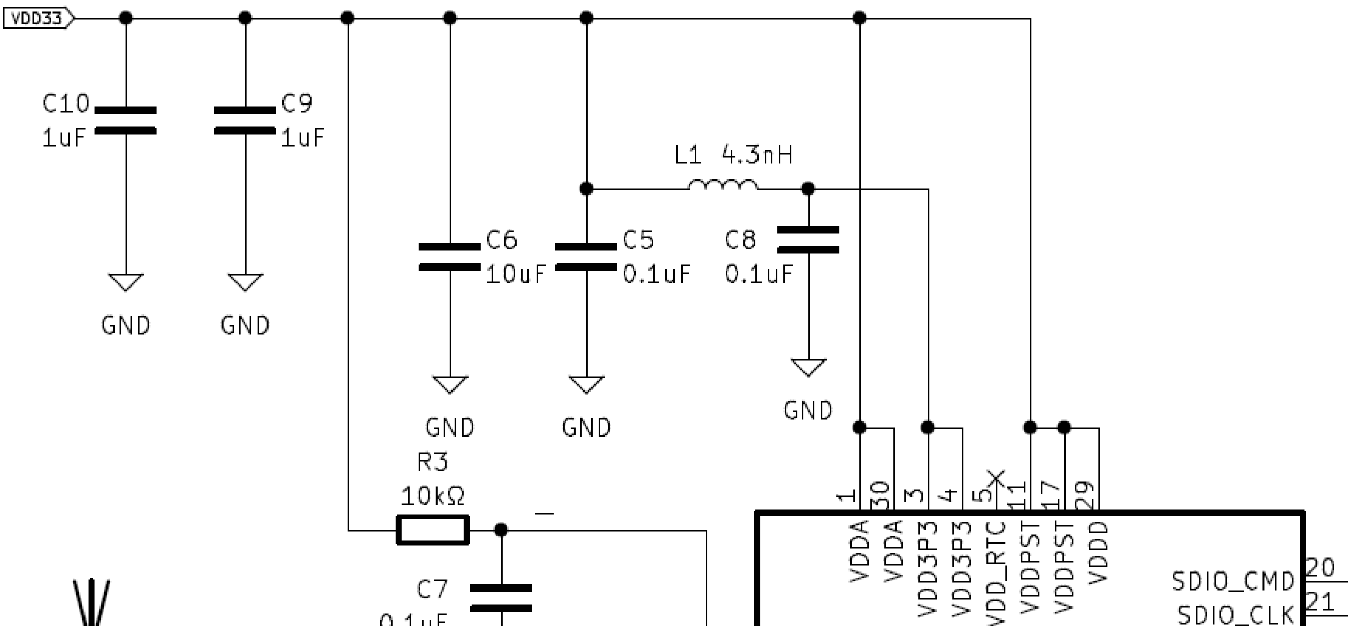

Figure 3: Digital and Analog Supplies Circuitry.

2.3.2 RC Delay Reset Circuitry

The reset pin of ESP8266EX is denoted as Pin32 EXT_RSTB and plays a vital role in the device's operation [3]. This pin possesses an internal pullup resistor and operates on an active low mechanism. As for Pin7 CHIP_EN, it functions as the enable pin for ESP8266EX. When held at a low state, ESP8266EX powers off [3]. Its connections can see on the Figure 2.3.2 shwon in the section (2.3.3) below.

2.3.3 12Kohm RES

An external GND 12kΩ showed on Figure 2.3.2 resistor should be connected with the ERS12K pin (pin 31) [3]. This component must have high accuracy when controlling the bias current. For this, a $12\text{k}\Omega \pm 1\%$ is required.

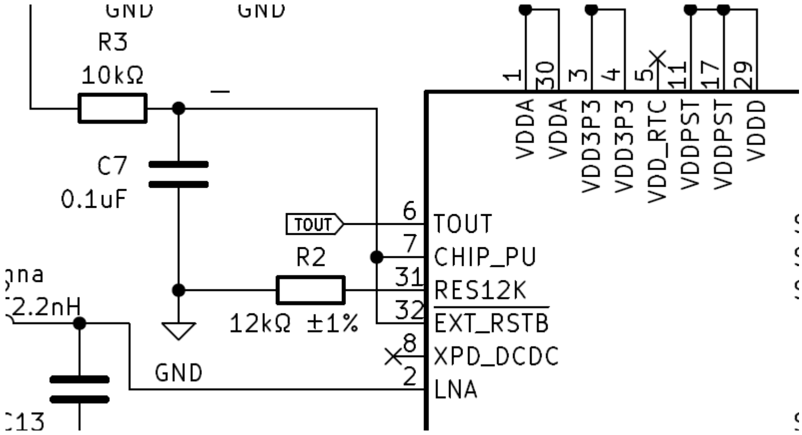

Figure 4: RC Delay Reset circuitry and 12K RES Pin.

2.3.4 RF Antenna Circuitry

For the 2.4Ghz RF antenna, in terms of "abstract" circuit itself, the only requirements is that the the impedance at the output terminal of the ESP8266 PA is $39 + j6\Omega$, which necessitates a corresponding matched impedance of $39 - j6\Omega$ (from the antenna to the chip). For such implementation, see the Figure 5 below.

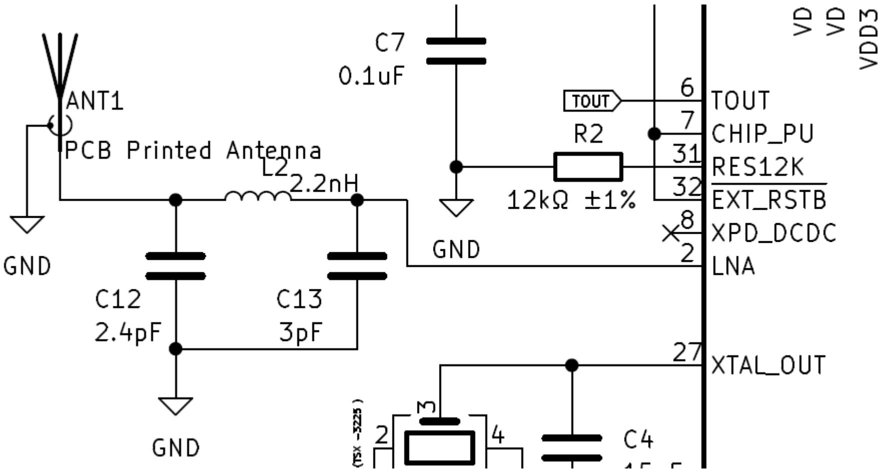

Figure 5: 2.4GHz Antenna circuitry.

2.3.5 XTAL Circuitry

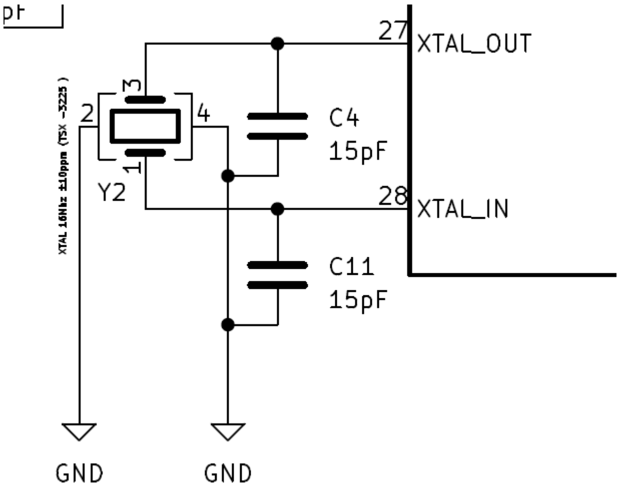

Figure 6: Crystal Oscillator Circuitry.

2.3.6 Drive Current Serial Clock RES

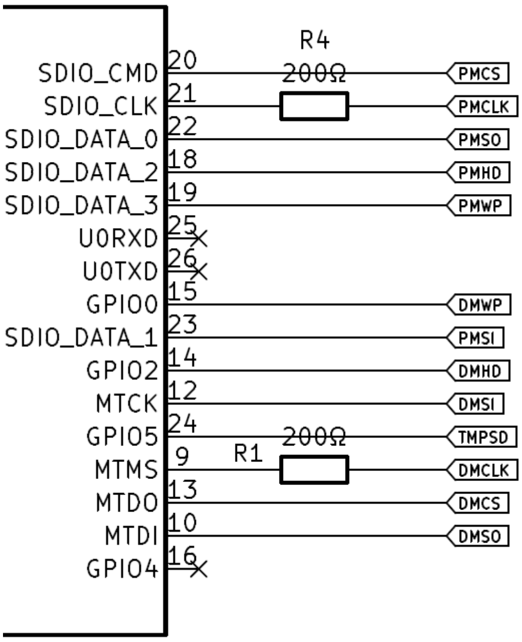

Figure 7: 200Ω RES Drive Current.

As for the Nets connected on each of the pins shown in Figure 7, "P" stands for "Program", "D" is "Data", "SI" and "SO" are "Serial IN" (MOSI) and "Serial Out" (MISO) respectivily (notice the "IN" "Out" refers to the slave pins). Moreover, "SCLK" means "Serial Clock", "WP" is "Write Protect', "HD" refers to "Hold". Finnaly, "TEMPSD" stands for "TMP 35

Shutdown". So, for instance, "DMCLK" refers to the "Data Memory Serial Clock" Net. From this point, the rest of the blocks are described.

2.3.7 Power Circuitry

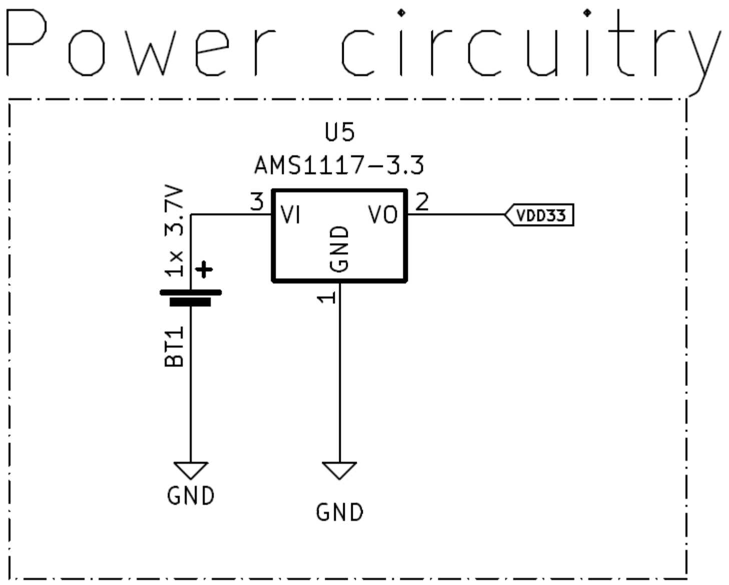

Figure 8: 3.3V Power circuitry.

2.3.8 Storage Circuitry

As for the program memory, since there is no internal memory available, an external flash memory is required for the ESP8266EX to store the user program. It is important to consider that the minimum capacity of the memory should be 512kB for disabled OTA (firmware updates Over The Air) or 1MB for enabled OTA. For firmware programming into the memory, the ESP8266EX supports Standard SPI, Dual SPI, and Quad SPI, which need to be selected prior to the programming process. The chosen memory, MX25R3235F [5], meets all the aforementioned requirements.

For the data memory, the requirements essentially boil down to having sufficient space (min 500kB) and supporting SPI communication. The chosen IC, AT25SF081 [1], fulfills these requirements. Additionally, this IC allows for serial data traffic on its I/Os at speeds of up to 50MHz (85MHz and 104MHz in specific cases). Since the communication with this chip was achieved through bit banging (see section 4.3), with an SCLK oscillation period of $6\mu s$, *i.e.*, a frequency of f = 166.666kHz, operating with this chip is entirely feasible, obviating the need for hardware-based SPI interface. Furthermore, The protection bits are set to zero by default [1], and there is no need to modify these bits. Therefore, the memory can be programmed and erased without any difficulties. Both program and data memory can be seen on Figure 9

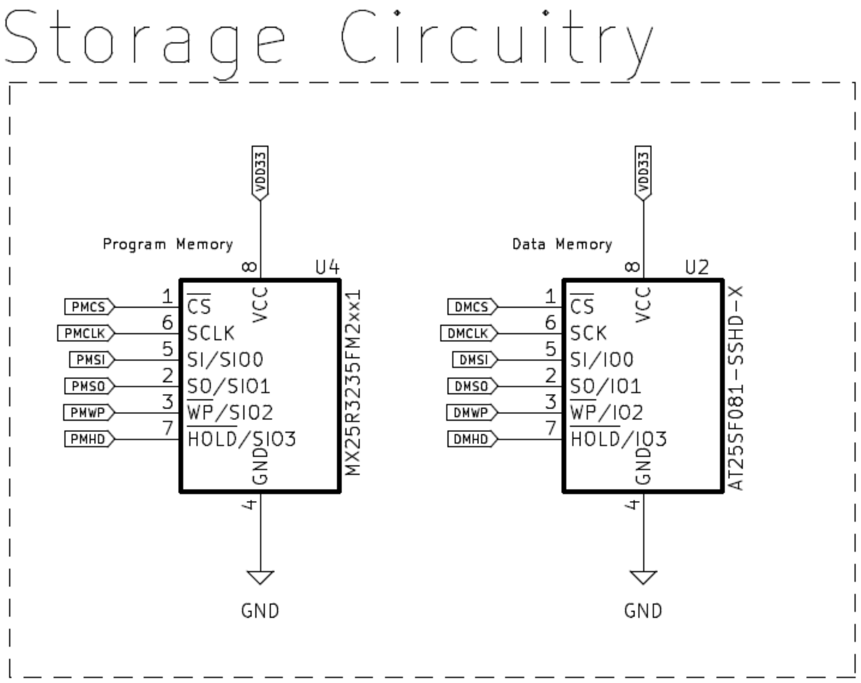

Figure 9: Storage circuitry - Program and Data Memory.

2.3.9 Temperature Sensor Circuitry

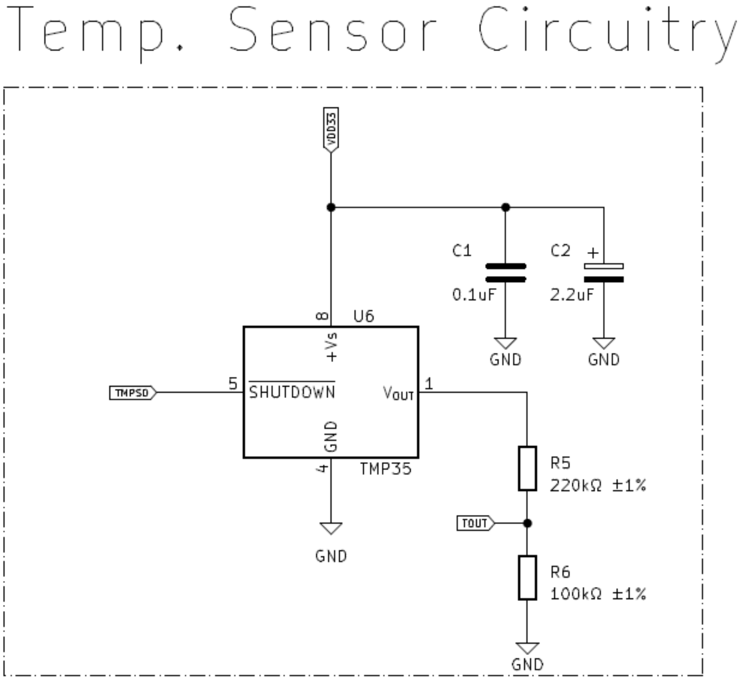

Figure 10: Temperature measuring circuitry.

2.4 Components

List of componentes, its values, packages, etc

3 Printed Circuit Board

3.1 PCB Design

3.2 PCB Specifications

3.2.1 ESP8266EX Power Supply Design

Prior to the power traces reaching the analog power-supply pins (Pin1, 3, 4, 28, 29), the inclusion of a 10 μF capacitor, working in conjunction with a 0.1 μF capacitor, becomes necessary. Additionally, an arrangement of a C circuit and an L circuit is advised for the power supplies of Pin3 and Pin4.The C-L-C circuit should be positioned as proximate as possible to the analog power-supply pin [3]. For more information about these components, refer to Figure 3. It is crucial to acknowledge that the power supply pathway can be susceptible to damage as a consequence of abrupt current surges during the transmission of analog signals by ESP8266EX. Consequently, it is imperative to incorporate an additional 10μF capacitor, packaged in either 0603 or 0805 dimensions, to complement the existing 0.1μF capacitor [3].

3.2.2 Layers Design

On [3] there are two suggested models regarding the number and dispositions of layers on a PCB that uses ESP8266EX (and by extension, an 2.4GHz RF antenna). The model with four layers was chosen. Here are the specifications:

- The initial layer corresponds to the uppermost part and is designated as the TOP layer, utilized for signal lines and component placement.
- The subsequent layer represents the GND layer, dedicated solely to establishing a comprehensive ground plane without the inclusion of signal lines.
- The following layer constitutes the POWER layer, exclusively reserved for the positioning of power lines. In certain unavoidable situations, it is admissible to incorporate some signal lines within this layer.
- Lastly, the final layer denotes the BOTTOM layer, exclusively allocated for signal line placement. It is discouraged to mount components on this particular layer.

Also, the power tracks should have a minimum size of 15 mils. The value used was 20 mils.

3.2.3 Reset Track Design

The EXT_RSTB (Pin32) possesses an internal pullup resistor and operates on an active low mechanism. To mitigate the possibility of external disturbances causing unwarranted resets, it is advised to maintain a concise PCB trace for EXT_RSTB and incorporate an RC circuit at this pin [3] (see section 2.3.2).

3.3 Trace Routing

3.4 Ground Planes

3.5 Tracks and Vias

3.6 RF Antenna

[3] 5

4 Firmware

4.1 Firmware Description

The ultimate goal of this firmware is to establish a complete wireless communication between a data collecting station (ESP8266EX) and an access point located 100m away using the WiFi protocol. The hardware collects data on-site, stores it with a size of 500kB, and then sends it to the access point.

For the purpose of data collection, a temperature measuring circuit was implemented on the hardware using a TMP36 IC. A linear time-invariant system, namely the Moving Average Filter, is required to ensure that the measured temperature remains in a stable state when it is saved in the flash data memory integrated into the hardware.

For the WiFi protocol, the library ”ESP8266WiFi.h” was used.

For the communication between the microcontroller and the Flash Data Memory, it is used the SPI communication protocol.

4.2 Firmware Architecture

4.3 Features and Algorithms

4.4 Configurations and Parameters

4.5 Interfaces and Protocols

```
#include <iostream>

#include <Arduino.h>

void setup() {
    Serial.begin(9600);
    pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
    digitalWrite(LED_BUILTIN, HIGH);
    delay(500);
    digitalWrite(LED_BUILTIN, LOW);
    delay(500);
}
```

References

[1] *8-Mbit SPI Serial Flash Memory with Dual-I/O and Quad-IO Support - AT25SF081 Datasheet*. URL: https://pdf1.alldatasheet.com/datasheet-pdf/view/606550/ETC2/AT25SF081.html.

[2] *Application Note AN043 Small Size 2.4 GHz PCB antenna*. URL: https://www.ti.com/lit/an/swra117d/swra117d.pdf?ts=1684435789781&ref_url=https%253A%252F%252Fwww.google.com%252F.

[3] *ESP8266 Hardware Design Guidelines*. URL: https://www.espressif.com/sites/default/files/documentation/esp8266_hardware_design_guidelines_en.pdf.

[4] *The GNU GLPL full text*. URL: https://github.com/esp8266/Arduino/blob/master/LICENSE.

[5] *Ultra low power, 32Mbit [x 1/ x2/ x4] CMOS MXSMIO (Serial Multi I/O Flash Memory - MX25R3235F Datasheet)*. URL: https://www.macronix.com/Lists/Datasheet/Attachments/8755/MX25R3235F,%20Wide%20Range,%2032Mb,%20v1.8.pdf.