

Remote fireworks system with Wi-Fi communication protocol

Gregoire Le Faou, Leo Ozenne, Killian Gomes, Anousith Phoutthasak,
Mouhamadou Toure

gregoire.lefaou@uphf.fr

leo.ozenne@uphf.fr

killian.gomes@uphf.fr

anousith.phoutthasak@uphf.fr

mouhamadou.toure@uphf.fr

Abstract— This project aims to develop a secure system for remotely triggering fireworks using the Wi-Fi communication protocol. The motivation behind this system is to enhance safety by enabling remote ignition and providing the capability to coordinate and synchronize multiple fireworks while detecting any errors during ignition. The system comprises three main components: the control board, the ignition board, and Arduino Nano boards. To establish communication, Wi-Fi protocol was chosen due to its adaptability, sufficient range, and ease of incorporating encryption for secure communication. The communication involves a control board acting as a relay between Bluetooth instructions from the control console and the ignition board. The ignition board receives Wi-Fi instructions from the control board and communicates with Arduino Nano boards via I2C. The report details the selection of microcontrollers and development environments, ultimately opting for the ESP8266 card for the control and ignition boards due to its compatibility with Wi-Fi and Bluetooth modules. The communication between the control console and the control board is established using Python for the PC application, and a data frame system is implemented to standardize communication between different hardware and programming languages. Safety encryption is implemented using the AES library on the Arduino IDE to protect the system from cyber-attacks. The report further discusses the Wi-Fi communication between the control board and the ignition board, and the I2C communication between the ignition board and Arduino Nano boards for individual tube control. The presented system offers a secure and synchronized approach to fireworks ignition, emphasizing safety and control. The communication protocols, encryption measures, and hardware integration contribute to a comprehensive solution for remote firework displays.

Keywords— fireworks, remote, communication protocol, UART, I2C, Wi-Fi, Bluetooth, control console, ignition board, control board, microcontroller, safety, encryption, ignition tubes, relays, programming

I. INTRODUCTION

We aim to develop a secure system to remotely trigger the lighting of fireworks using the Wi-Fi communication protocol. Since the lighting of fireworks presents risks for the safety of users, it is better to opt for a system that allows remote triggering, thus providing better security. In addition, our ignition system would offer the possibility to coordinate and synchronize the ignition of different fireworks, as well as detect any errors during the ignition.

To establish the communication between the ignition system and the control system, we chose to use the Wi-Fi

protocol. This communication protocol is mainly used for home automation and connected objects applications. The advantage of Wi-Fi is that this protocol has a range adapted to what is required for our application (approx. 50 m outdoors), that it is easily adaptable to communicate with a microcontroller and that it is easy to incorporate encryption features to communicate securely.

Our system consists of five ignition tubes, each equipped with five electric igniters (relays), allowing individual control of rockets.

Here is the diagram of our system illustrating its operation and components:

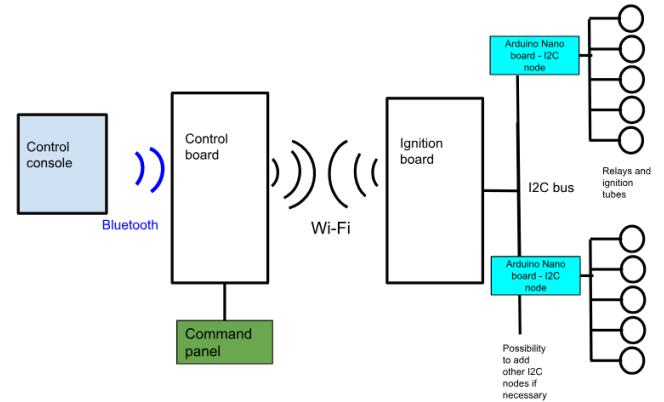


Fig. 1. Explanatory diagram of the functioning of our system

Thus, we have to program three boards: the control board, the ignition board and the Arduino Nano board. The control board acts as a relay between the instructions from the control console (Bluetooth) and the control board. As for the ignition board, it will receive instructions via the Wi-Fi protocol of the control board and will communicate with our Arduino Nano boards via an I2C communication.

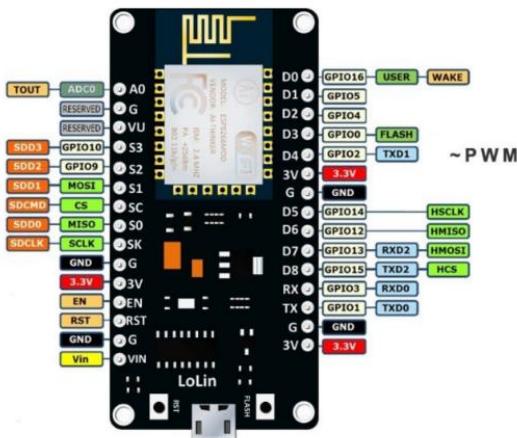
II. CHOICE OF MICROCONTROLLER AND INTEGRATED DEVELOPMENT ENVIRONMENT (IDE)

For the choice of the control and control board, we first studied the 2 most consistent tracks for our application: a programmable microcontroller with the Arduino IDE and a Silicon Labs development board adapted to Z-Wave communication (the Z-Wave Thunderboard 800 Series).

	Assets	Drawbacks	Features
ESP8266	<ul style="list-style-type: none"> -Dimension 58 x 31 x 12 mm -Used for home automation -Range (up to 50m) -Coding with the Arduino IDE -Higher computing capacity than Arduino 	High energy consumption	<ul style="list-style-type: none"> -Tensilica 32-bit RISC CPU Xtensa LX10 -Memory: 512 KB to 4 MB flash, 64 kB RAM -Interface: 29 pins UART, SPI, I2C, GPIO, PWM, ADC, Wi-Fi -Power supply 3.3V
Serie 800 Thunderboard Silicon Labs	<ul style="list-style-type: none"> -Fairly compact size: 100*70mm -Long range (up to 150m) -Low energy consumption -Used for home automation -Much better performance than Arduino board 	Development in C more complex than the Arduino language	<ul style="list-style-type: none"> -ARM Cortex M33 32 bits -Memory: 512 KB flash, 64 kB RAM -Wireless transmission with SMA antenna -Multiple integrated sensors -Interface: 20 pins GPIO, UART, SPI, I2C, USB -Power supply 3.6 to 6V DC

TABLE I. COMPARISON BETWEEN TWO BOARDS

Regarding the communication between the control board and the ignition board, we started with the Z-Wave communication protocol which was integrated into the Thunderboard 800 Series card. However, given the complexity of the IDE and programming language, we finally opted for the ESP8266 card and to use the Wi-Fi protocol (integrated) for communication. We will integrate a Bluetooth module (HC-05 module) to our control board in order to receive data from our control panel.



For a sequence of more than 60 characters we then have frames of the form: S01TCb3Ca1... / S02TD1000Cc2E.

The last flag defined is the one representing the start instruction of the ignition sequence. This flag is A.

B. Safety encryption

In order to protect the functioning of our system from a cyber-attack, it is necessary to set up a safety encryption.

To do this, we need to install Advanced Encryption Standard (AES) library on Arduino IDE.

The principle of encryption consists on converting data into an unreadable form for those who do not have the decryption key. This helps protect sensitive data from unauthorized access. By using encryption algorithms, we can secure communications and stored data. In our case, the AES encryption process uses one key. We generate one key to encrypt and another to decrypt. The length of the key determines the level of security. To encrypt data, we use the corresponding function of AES library and we provide it with unencrypted data as well as the encryption key. Concerning secure transmission, encrypted data can be securely transmitted over the network or stored securely. Only people or devices that have the decryption key can recover the original data. For data decryption, on the receiver side, the encrypted data is decrypted using the decryption key, this will recover the original data.



Fig. 4. Capture of a non-encrypted frame with Wireshark : we can recover the data coded in ASCII

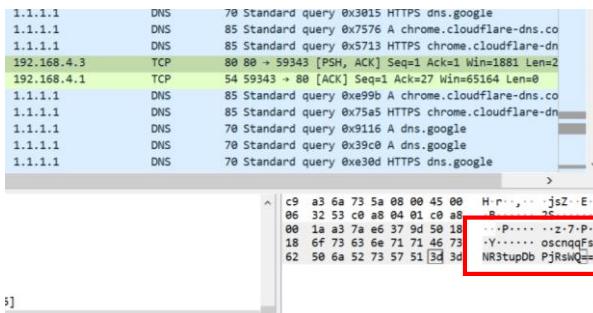


Fig. 5. Capture of an encrypted frame with Wireshark : we can see that the intercepted data are not usable

C. Wi-Fi communication between the control board and the ignition board

Once the Bluetooth communication and safety encryption are established, the control board must read the received message in order to transmit it to the ignition board by Wi-Fi. In our case, we use the ESP8266 card as an ignition card and control card.

So, we established a code to manage the Wi-Fi connection between a control card and an ignition card. By configuring the control board as an access point, the code will allow the ignition board to connect to the control board via Wi-Fi. The Wi-Fi credentials (SSID and password) have been set, and the IP address of the access point is displayed. Our code also integrates a web server and WebSocket to facilitate bidirectional communication between the two cards.

Once the data is received, the control board will then communicate with the Arduino Nano boards.

D. I2C communication between the ignition board and Arduino Nano boards

Once the ignition information is received by the control board, it must be transmitted to the I2C bus to turn on the correct ignition tube.

To do this, we connected two Arduino Nano boards a and b to the ignition board (ESP8266) by connecting the SDA and SCL ports of the ESP8266 and the Arduino Nano boards to each other via the I2C bus.

Then, we connected three ignition tubes to each Arduino Nano via electrical relays; tubes number 1, 2 and 3 being respectively connected to the Arduino Nano a on ports D8, D9 and D10, and tubes number 4, 5 and 6 being respectively connected to the Arduino Nano b on ports D8, D9 and D10. As previously said in the Bluetooth communication part, typical frames Ma1, Ma2, Ma3, Mb1, Mb2 and Mb3 respectively correspond to the manual ignition of tubes number 1, 2, 3, 4, 5 and 6.



Fig. 6. Control console and control board

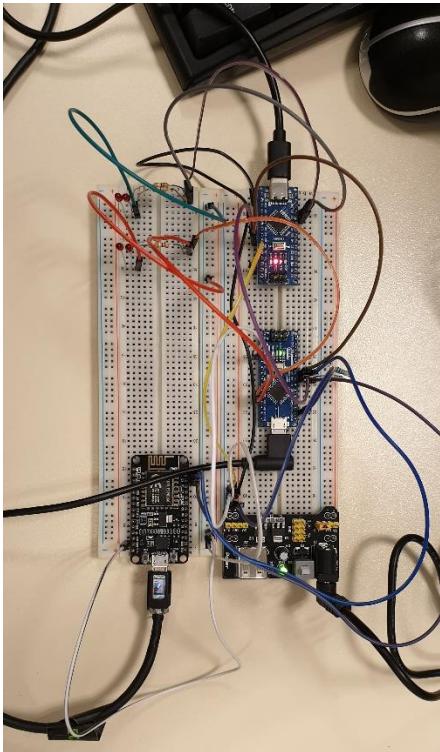


Fig. 7. Ignition board, Arduino Nano boards and LEDs

IV. CONCLUSION

The development and implementation of a secure remote ignition system for fireworks offer substantial advantages in terms of safety and coordination. By leveraging the Wi-Fi communication protocol, our system provides an adequate range for outdoor applications, approximately 50 meters, while allowing seamless adaptability for communication with microcontrollers.

The choice of the ESP8266 card for the control and ignition boards, coupled with the integration of a Bluetooth module (HC-05), enhances the versatility of the system. The utilization of Python for the control console facilitates a user-friendly interface, enabling the manual ignition of individual cylinders or the execution of pre-programmed sequences.

A robust safety encryption layer, implemented through the AES library on the Arduino IDE, safeguards the system against potential cyber threats. This encryption ensures the integrity of communication channels, mitigating risks associated with unauthorized access or tampering.

The structured communication framework, involving data frames and unique flags for instructions, streamlines interworking between different hardware supports and programming languages. This approach allows for clear and concise communication between the control console and the control board, ensuring efficient transmission of commands.

Incorporating I2C communication between the ignition board and Arduino Nano boards facilitates individual tube control, enabling precise and synchronized ignition sequences. The seamless integration of these technologies positions our system as a comprehensive solution for secure and coordinated firework displays.

As technology continues to advance, our work represents a meaningful contribution to the field of pyrotechnics, emphasizing safety, precision, and the potential for innovative applications in event coordination. The presented system aligns with contemporary trends in wireless communication and IoT applications, showcasing the adaptability and scalability of our approach.

ACKNOWLEDGMENTS

We are grateful to our supervisor Mr. Delebarre who provided insights and expertise that greatly assisted the project.

We also thank the whole team for the involvement and the advance of the project since the beginning of the project.

REFERENCES

- [1] “ESP8266 Technical Reference”, https://www.espressif.com/sites/default/files/documentation/esp8266_technical_reference_en.pdf.
- [2] “Arduino Nano Datasheet”, <https://docs.arduino.cc/resources/datasheets/A000005-datasheet.pdf>.
- [3] “Connecting Two Nano Every Boards Through I2C”, <https://docs.arduino.cc/tutorials/nano-every/i2c/>.
- [4] “HC-05 Bluetooth module datasheet”, https://components101.com/sites/default/files/component_datasheet/H_C-05%20Datasheet.pdf.
- [5] “Bluetooth module connection Arduino HC-06/ HC-05”, <https://arduino-france.site/bluetooth-hc-05/>
- [6] “Pull up resistor value”, <https://docs.particle.io/reference/device-os/api/wire-i2c/pull-up-resistors-i2c/>.
- [7] “AESLib encryption”, <https://www.arduino.cc/reference/en/libraries/aeslib/>.