International Journal for Multidisciplinary Research (IJFMR)

E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

IOT-Controlled Explosion

Anas A P

Teaching Assistant, Department of Computer Science, Rajagiri College of Social Sciences, Cochin

Abstract

This paper delves into the study and implementation of IoT-controlled explosions, exploring their relevance in various industrial sectors such as quarrying, mining, and building demolition. Additionally, the paper examines the application of controlled explosions in domestic settings, such as the ignition of firecrackers and the use of small explosions to deter animals from farmlands and fields. The term "controlled explosion" in this context refers to the ability to manage detonations remotely, mitigating the risks associated with potential injuries or fatalities.

Conventional explosion methods pose significant hazards to both human life and property. In response to this, the research paper introduces a novel approach to controlled explosions utilizing an open-source IoT development board named NodeMCU. This board, powered by the ESP8266 Wi-Fi System-on-Chip (SoC) from Espressif Systems, enables remote control of explosions through devices such as mobile phones or computers. The proposed method enhances safety by providing a secure and efficient means of managing explosive events from a distance.

Keywords: IoT, Explosion, NodeMCU, ESP8266, Microcontroller, Arduino

1. Introduction

This project focuses on remotely detonating various types of explosives using IoT technology. The chosen platform for this endeavor is the open-source IoT development board NodeMCU. The motivation behind this initiative stems from the alarming rates of injuries and fatalities associated with the traditional methods of igniting explosives, particularly during festivities and ceremonies involving firecrackers in India. The prevalent practice of firing fireworks in close proximity significantly elevates the risk of accidents, especially considering the high explosive capacity of many of these pyrotechnics.

Evidence from hospital statistics in the Delhi National Capital Region reveals a concerning surge in firecracker-related injuries from 2002 to 2010. During this period, the hospital recorded an average of one patient per 100,000 populations, with 73.02% of the victims falling within the 5–30 age group. A tragic incident at the Puttingal Temple in Paravur, Kerala, India, on April 10, 2016, further underscores the gravity of the situation. Following a mishap during firework celebrations, an explosion and fire resulted in 111 fatalities and over 350 injuries, some of which involved severe burns. The blast also caused extensive damage to the temple and approximately 150 surrounding houses.

In light of these alarming statistics and tragic incidents, this paper aims to provide a comprehensive overview of the application of IoT in the controlled detonation of explosives, specifically in industrial, mining, and domestic settings.

2. Existing system

Fireworks and explosives in the existing system are typically ignited manually with bare hands or through the use of a timer mechanism. In cases where a timer is employed, the system ignites the explosive by generating a spark through a connected fuse at a predetermined time.

However, this approach has notable drawbacks. Direct manual ignition poses a safety risk, increasing the likelihood of injuries due to unexpected explosions. Additionally, it lacks precision. In the case of using a timer, the system's limitations become apparent. The explosion occurs solely at the pre-set time, offering limited control over the system. Adjusting the timing is not possible, and stopping the explosion requires physical contact with the system, rendering it impractical for remote control.

In summary, the current system has safety concerns associated with manual ignition and lacks flexibility and remote control capabilities, particularly in terms of adjusting the timing and stopping the explosion remotely.

3. Proposed system

The proposed system involves the utilization of the Arduino IDE to program the NodeMCU. Following the sketch upload, a 5-volt relay module is connected to the NodeMCU. Notably, GPIO Pin 4 is configured as HIGH, with an understanding that for the ESP8266, a LOW voltage level corresponds to the Pin being ON (as it operates on an active-low logic). The positive terminal of the 5-volt relay module is linked to GPIO Pin 4, while the negative terminal is connected to the ground reference of the NodeMCU.

Further, a 4-volt 2 Ah lead-acid battery is integrated into the system, with its positive terminal connected to the Common contact (CO) terminal of the relay module. Simultaneously, the negative terminal of the battery is connected to one end of a Nichrome (NiCr) wire. The choice of Nichrome wire is deliberate, as its high resistance facilitates the efficient conversion of electrical energy into heat energy, resulting in the wire becoming red hot. The opposite end of the Nichrome wire is intricately linked to the fuse of explosives such as crackers or other fireworks.

To control the system remotely, an Android App has been developed specifically for activating the relay connected to GPIO Pin 4. This Android App is crafted using MIT App Inventor, providing a user-friendly interface for seamless operation of the proposed explosive ignition system.

4. Technology used

The proposed system incorporates various technologies, such as MIT App Inventor, Arduino IDE, ESP8266 NodeMCU, and a 5-volt relay module. MIT App Inventor facilitates the creation of an Android app, enabling user control over the NodeMCU to activate the relay and initiate an explosive event.

5. IoT (Internet of Things)

The term "Internet of Things" (IoT) refers to a network of tangible devices, gadgets, appliances, and other physical devices that contain sensors, software, and network connections that enable them to store and exchange information. These products are often referred to as "smart organizations" and range from "smart home" products such as smart thermometers to wearable devices such as smartwatches and RFID-enabled clothing. Additionally, IoT continues its impact on technology and transportation. The vision in the technology field is also to see the realization of "connected cities" based on IoT technology.

6. NodeMCU

NodeMCU is a versatile open-source electronics platform that combines hardware and software to facilitate the development of Internet of Things (IoT) applications. It is based on the popular ESP8266 microcontroller, equipped with built-in Wi-Fi capabilities, and comes with an easy-to-use Lua-based scripting language.

NodeMCU simplifies IoT prototyping and development by offering a range of pre-built functions and libraries, allowing researchers and developers to quickly create and deploy connected devices without the need for extensive low-level programming. Its compact form factor, low cost, and rapid development capabilities make it an attractive choice for various research projects involving IoT, home automation, sensor networks, and more.

Features of NodeMCU

- ESP8266 Microcontroller: NodeMCU is built around the ESP8266 microcontroller, which is a cost-effective and power-efficient chip with integrated Wi-Fi capabilities, making it suitable for IoT projects.
- Wireless Connectivity: NodeMCU provides built-in Wi-Fi connectivity, enabling easy communication with the internet and local networks using the 802.11 b/g/n standards.
- Scripting Languages: Originally, NodeMCU used Lua as its scripting language, simplifying development. However, it supports multiple programming languages, including Arduino C/C++, Python, and JavaScript, providing flexibility to developers.
- GPIO Pins: NodeMCU offers multiple GPIO pins, supporting digital input/output, analog input, and PWM functionality, allowing for interfacing with various sensors, actuators, and peripherals.
- I2C and SPI Support: It supports I2C and SPI communication protocols, facilitating connectivity with a wide range of sensors, displays, and other devices.
- Analog-to-Digital Converter (ADC): An onboard ADC allows NodeMCU to read analog sensor data and convert it into digital form for processing.
- USB Connectivity: NodeMCU can be powered and programmed via USB, making it convenient for development and debugging.
- Node.js and MicroPython Support: In addition to Arduino-based programming, NodeMCU can be programmed using Node.js and MicroPython, expanding its development possibilities.
- Community and Libraries: NodeMCU has a thriving community, offering a wealth of libraries, tutorials, and resources for developers to leverage.
- Open-Source: Both its hardware and firmware are open-source, fostering customization and modification for specific project requirements.
- Compact Form Factor: NodeMCU boards are compact, allowing for easy integration into IoT devices and prototypes.
- Low Power Consumption: The ESP8266 chip is designed for low-power applications, ideal for IoT devices that may operate on battery power.

7. ESP8266

The ESP8266 chip is a small, cost-effective, and Wi-Fi-enabled microcontroller widely used in IoT and embedded systems. It allows devices to connect to the internet, making them smarter and more connected.

8. Arduino IDE

The Arduino Integrated Development Environment (IDE) is a widely used open-source software platform designed to simplify the programming and development of embedded systems and microcontroller-based projects. It provides a user-friendly interface for writing, compiling, and uploading code to Arduino microcontrollers, which are prevalent in the field of electronics and embedded systems research.

9. MIT App Inventor

MIT App Inventor is a user-friendly and educational platform for creating mobile applications without the need for extensive coding skills. Developed by the Massachusetts Institute of Technology (MIT), it provides a visual, block-based programming interface that allows individuals, especially beginners and educators, to design and develop Android apps. MIT App Inventor simplifies app creation by enabling users to drag and drop code blocks, making it accessible for students, hobbyists, and aspiring app developers to bring their app ideas to life through an intuitive and educational environment.

10. NodeMCU Code

```
#include <ESP8266Wifi.h>
WiFiClient client;
WiFiServer server(80);
const char* ssid = “yourssid”;
const char* password = “password”;
String data =””;
int cracker = 4;
void setup()
{
        pinMode(cracker, OUTPUT);
        server.begin();
}
void loop()
{
        client = server.available();
        if (!client) return;
        data = checkClient ();
        if (data == "fire") fire();
        else if (data == "off") offi();
        else if (data == "timer") timer();
}
void fire(void)
{
        digitalWrite(cracker,LOW);//LOW because relay is active low component
```

```
}
void offi(void)
{
	digitalWrite(cracker,HIGH);
}
void timer(void)
{
	delay(9000);//9 second delay
	digitalWrite(cracker, LOW);
}
String checkClient (void)
{
	while(!client.available()) delay(1);
	String request = client.readStringUntil(‘\r’);
	request.remove(0, 5);
	request.remove(request.length()-9,9);
	return request;
}
```

11. References

1. https://doi.org/10.36948/ijfmr.2023.v05i02.2338
2. https://doi.org/10.36948/ijfmr.2023.v05i03.2914
3. https://www.arduino.cc/
4. https://nodemcu.readthedocs.io/en/release/