IoT Based Smart Multi Application Surveillance Robot

**ABSTRACT**

## Aims

The aim of this project is to design and develop an IoT-based Smart Multi- Application Surveillance Robot to reduce casualties on the battlefield by enhancing security and monitoring capabilities. The robot is intended to capture intruder information, neutralize threats, and perform multiple applications such as bomb detection in public places.

## Study Design

* Controlling the robot in manual mode using IoT technology via Android/PC.
* Implementing a wireless vision camera for live video streaming.
* Integrating PIR (Passive Infrared) and metal detection sensors.
* Utilizing location tracking technology.

## Methodology

* IoT Control: The robot's movement is controlled remotely using an Android phone or PC, overcoming short-range communication issues.
* Surveillance and Detection: The robot uses a wireless vision camera for live video streaming, and PIR and metal detectors for identifying intruders and potential threats.
* Threat Neutralization: Upon detecting an intruder or metallic object, the robot halts and activates its laser gun to neutralize the threat.
* Location Tracking: The robot tracks and reports the location of detected threats using location tracking technology.

## Results

The robot successfully captured and transmitted real-time video of its surroundings, detected intruders and metallic objects, and neutralized threats using the laser gun. Additionally, it accurately tracked and reported the location of detected objects, demonstrating its effectiveness in various surveillance applications.

## Conclusion

The IoT-based Smart Multi-Application Surveillance Robot significantly enhances battlefield safety and reduces the risk to soldiers by capturing critical information and neutralizing threats before engagement. Its ability to perform multiple functions, including bomb detection and location tracking, underscores its versatility and potential impact on modern security and surveillance practices**.**

# INTRODUCTION

In an era marked by rapid technological advancements, the realm of surveillance and security has witnessed a paradigm shift (Venu, 2022). Traditional methods of monitoring and safeguarding assets, premises, and individuals have been supplemented, if not surpassed, by innovative solutions powered by the Internet of Things (IoT) (Sanneman et al., 2021; Romeo et al., 2020; Zhang et al., 2020). Among these cutting-edge innovations stands the IoT-based Smart Multi- Application Surveillance Robot – a marvel of engineering ingenuity and technological integration poised to redefine the landscape of surveillance and monitoring.[1]

# METHODOLOGY

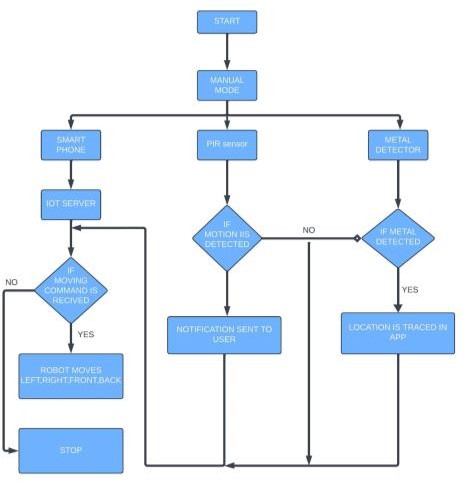


Fig .1 Flowchart of Iot Based Smart Multi Application Surveillance Robot

The robotic system described operates in two primary modes: manual and automatic[2]. In manual mode, users control the robot's movement via a Smartphone or PC, directing it forward, reverse, left, or right. Alternatively, in automatic mode, an Internet of Things (IoT) server communicates movement instructions to the robot, initiating designated actions upon receipt. Automatic

operation commences with the IoT server transmitting specific movement commands to the robot, prompting it to execute the instructed actions promptly.

The integration of safety features ensures the robot operates efficiently and safely in its environment. One such feature is the Passive Infrared (PIR) sensor, which continuously scans for motion. If no motion is detected, the robot proceeds with its tasks; however, upon identifying movement, it halts to prevent potential collisions Additionally, a metal detection system enhances safety protocols. A dedicated sensor scans the surroundings for metal presence. If no metal is detected, the robot follows its instructions.

Conversely, upon metal detection, the robot stops abruptly, prioritizing safety over task completion. The accompanying app provides users with remote control capabilities, facilitating visual inspection of the robot's surroundings through camera rotation. Additionally, it enables remote activation of the robot's built-in laser gun, suggesting potential security applications. The system's flowchart concludes with a "STOP" function, indicating a method to power down the system, ensuring controlled shutdown procedures. While the provided information suggests the robot's primary purpose is surveillance, certain aspects hint at broader security applications. Notably, the system prioritizes metal detection over motion detection, emphasizing safety measures to prevent potential hazards.

The inclusion of a laser gun function underscores the system's potential suitability for security-related tasks However, a comprehensive understanding of the system requires further clarification on specific components and their functionalities [9]. Additional details regarding the integration of various sensors, communication protocols, and safety mechanisms would provide deeper insights into the system's capabilities and operational procedures. In summary, the described robotic system represents a sophisticated blend of automation, safety features, and remote control functionalities, catering to surveillance and potentially broader security

# RESULTS

A multi-purpose robot that can perform metal detecting, Obstacles detection, live location tracking and live video can provide several benefits in different scenarios. Here are some specific results that can be achieved by using such a multi-purpose robot:

* + **Metal detecting:** The robot can be equipped with metal detectors to detect metal objects in public places such as airports, train stations, and public events. This can help to improve security and prevent potential threats.
  + **Obstacles detection:** The robot can be equipped with ultrasonic sensor metal to detect any Obstacles in the direction of the Robot movement and if any Obstacle detected Robot sent a message to android phone
  + **Live location tracking and live video:** The robot can be equipped with Bluetooth and WiFi module to track the live location of the Robot. And also Robot equipped with Camera for live video also we can record the live streaming video

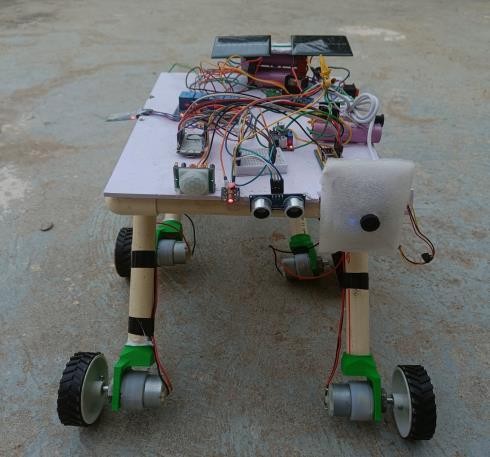


Fig .2: Snap of IoT Based Smart Multi Application Surveillance Robot.

# CONCLUSION

In conclusion, the IoT-based Smart Multi-Application Surveillance Robot represents a pinnacle of technological innovation and engineering excellence, poised to redefine the landscape of surveillance and monitoring across diverse applications and environments. From its advanced sensor technologies to its intuitive user interface, each facet of the robot's design and functionality embodies a commitment to excellence and efficacy, thereby empowering users with unparalleled surveillance capabilities and peace of mind. As society continues to grapple with evolving security challenges, the role of innovative surveillance solutions such as the Smart Multi-Application Surveillance Robot becomes increasingly indispensable. By harnessing the power of IoT-based technologies and sustainable energy sources, this revolutionary robot exemplifies a harmonious synthesis of technological innovation, environmental stewardship, and societal resilience, thereby paving the way for a safer, more secure future for generations to come

# REFERENCES

[1].Lavanya K N, Ramya Shree D, Nischitha B R, T Asha and C Gururaj "Vision Interfaced War Field Robot With Wireless Video Transmission"Proceedings of the 2nd International Conference on Trends in Electronics and Informatics (ICOEI 2018) IEEE Conference Record: # 42666; IEEE Xplore ISBN:978-1-5386- 3570-4.

[2].Ahsanul Hoque, Md. Baijid Hasan Shorif Shekh Nuruzzaman and Md. Eftekhar Alam, "Arduino based Battlefield Assistive Robot" 2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC) 21 - 23 Dec 2017, Dhaka, Bangladesh.

[3].Jignesh Patoliya, Haard Mehta and Hitesh Patel, "Arduino Controlled War Field Spy Robot using Night Vision Wireless Camera and Android Application" 2015 5th Nirma University International Conference on Engineering (NUiCONE).

[4]. Mohammad Shoeb Shah, Borole. P.B. “Surveillance And Rescue Robot Using Android Smart Phone And Internet”. International Conference on Communication AndSignal Processing, India. (2016).

[5].Ahn, H., Park, J., & Kim, K. (2019). Development of a Smart Surveillance Robot with Autonomous Navigation for Safety Monitoring. International Journal of Precision Engineering and ManufacturingGreen Technology, 6(1), 199-207.

[6].Liu, Y., Ma, S., & Tang, W. (2020). Design and Implementation of a Smart Surveillance Robot Based on Raspberry Pi. IEEE Access, 8, 114846-114856

[7] Sanneman, L., Fourie, C., & Shah, J. A. (2021). The state of industrial robotics: Emerging technologies, challenges, and key research directions. *Foundations and Trends® in Robotics*, *8*(3), 225-306.

[8] Romeo, L., Petitti, A., Marani, R., & Milella, A. (2020). Internet of robotic things in smart domains: Applications and challenges. *Sensors*, *20*(12), 3355.

[9] Zhang, Q., Jin, T., Cai, J., Xu, L., He, T., Wang, T., ... & Lee, C. (2022). Wearable triboelectric sensors enabled gait analysis and waist motion capture for IoT‐based smart healthcare applications. *Advanced Science*, *9*(4), 2103694.

[10] Venu, D. N. (2022). IOT Surveillance Robot Using ESP-32 Wi-Fi CAM & Arduino. *IJFANS International Journal of Food and Nutritional Sciences*, *11*(5), 198-205.

# APPENDIX

#include "stm32f4xx\_hal.h"

#define MOTOR1\_PIN GPIO\_PIN\_0 #define MOTOR2\_PIN GPIO\_PIN\_1 #define SERVO1\_PIN GPIO\_PIN\_2 #define SERVO2\_PIN GPIO\_PIN\_3 UART\_HandleTypeDef huart;

void SystemClock\_Config(void); static void MX\_GPIO\_Init(void);

static void MX\_USART2\_UART\_Init(void); int main(void) {

HAL\_Init(); SystemClock\_Config(); MX\_GPIO\_Init(); MX\_USART2\_UART\_Init();

while (1) {

}

}

void SystemClock\_Config(void) { RCC\_OscInitTypeDef RCC\_OscInitStruct;

RCC\_ClkInitTypeDef RCC\_ClkInitStruct;

HAL\_RCC\_PWR\_CLK\_ENABLE();

HAL\_PWR\_VOLTAGESCALING\_CONFIG(PWR\_REGULATOR

\_VOLTAGE\_SCALE1);

RCC\_OscInitStruct.OscillatorType = RCC\_OSCILLATORTYPE\_HSI;

RCC\_OscInitStruct.HSIState = RCC\_HSI\_ON;

RCC\_OscInitStruct.HSICalibrationValue = RCC\_HSICALIBRATION\_DEFAULT;

RCC\_OscInitStruct.PLL.PLLState = RCC\_PLL\_ON; RCC\_OscInitStruct.PLL.PLLSource = RCC\_PLLSOURCE\_HSI; RCC\_OscInitStruct.PLL.PLLM = 16;

RCC\_OscInitStruct.PLL.PLLN = 336; RCC\_OscInitStruct.PLL.PLLP = RCC\_PLLP\_DIV4; RCC\_OscInitStruct.PLL.PLLQ = 7; HAL\_RCC\_OscConfig(&RCC\_OscInitStruct);

RCC\_ClkInitStruct.ClockType = RCC\_CLOCKTYPE\_SYSCLK | RCC\_CLOCKTYPE\_PCLK1 | RCC\_CLOCKTYPE\_PCLK2;

RCC\_ClkInitStruct.SYSCLKSource = RCC\_SYSCLKSOURCE\_PLLCLK;

RCC\_ClkInitStruct.AHBCLKDivider = RCC\_SYSCLK\_DIV1; RCC\_ClkInitStruct.APB1CLKDivider = RCC\_HCLK\_DIV2;

RCC\_ClkInitStruct.APB2CLKDivider = RCC\_HCLK\_DIV1; HAL\_RCC\_ClockConfig(&RCC\_ClkInitStruct,

FLASH\_LATENCY\_5);

}

static void MX\_GPIO\_Init(void) { GPIO\_InitTypeDef GPIO\_InitStruct;

HAL\_RCC\_GPIOA\_CLK\_ENABLE();

HAL\_RCC\_GPIOB\_CLK\_ENABLE();

GPIO\_InitStruct.Pin = MOTOR1\_PIN | MOTOR2\_PIN; GPIO\_InitStruct.Mode = GPIO\_MODE\_OUTPUT\_PP; GPIO\_InitStruct.Pull = GPIO\_NOPULL; GPIO\_InitStruct.Speed = GPIO\_SPEED\_FREQ\_LOW; HAL\_GPIO\_Init(GPIOA, &GPIO\_InitStruct); GPIO\_InitStruct.Pin = SERVO1\_PIN | SERVO2\_PIN; GPIO\_InitStruct.Mode = GPIO\_MODE\_OUTPUT\_PP; GPIO\_InitStruct.Pull = GPIO\_NOPULL; GPIO\_InitStruct.Speed = GPIO\_SPEED\_FREQ\_LOW; HAL\_GPIO\_Init(GPIOB, &GPIO\_InitStruct);

}

static void MX\_USART2\_UART\_Init(void) {

huart.Instance = USART2; huart.Init.BaudRate = 9600;

huart.Init.WordLength = UART\_WORDLENGTH\_8B; huart.Init.StopBits = UART\_STOPBITS\_1; huart.Init.Parity = UART\_PARITY\_NONE; huart.Init.Mode = UART\_MODE\_TX\_RX; huart.Init.HwFlowCtl = UART\_HWCONTROL\_NONE; huart.Init.OverSampling = UART\_OVERSAMPLING\_16; HAL\_UART\_Init(&huart);

}

void HAL\_UART\_MspInit(UART\_HandleTypeDef\* uartHandle) { GPIO\_InitTypeDef GPIO\_InitStruct = {0};

if(uartHandle->Instance == USART2) {

HAL\_RCC\_USART2\_CLK\_ENABLE();

HAL\_RCC\_GPIOA\_CLK\_ENABLE();

GPIO\_InitStruct.Pin = GPIO\_PIN\_2 | GPIO\_PIN\_3; GPIO\_InitStruct.Mode = GPIO\_MODE\_AF\_PP; GPIO\_InitStruct.Pull = GPIO\_PULLUP;

GPIO\_InitStruct.Speed = GPIO\_SPEED\_FREQ\_VERY\_HIGH;

GPIO\_InitStruct.Alternate = GPIO\_AF7\_USART2; HAL\_GPIO\_Init(GPIOA, &GPIO\_InitStruct);

}

}

void USART2\_IRQHandler(void) { HAL\_UART\_IRQHandler(&huart);