A Surveillance Mobile Robot based on Low-Cost Embedded Computers

Krisanth Tharmalingam[1, 2] and Emanuele Lindo Secco[2] [0000-0002-3269-6749]

[1] Azenta Life Sciences, Manchester, UK
[2] Robotics Lab, School of Mathematics, Computer Scinece and Engineering, Liverpool Hope University, UK
19010816@hope.ac.uk, seccoe@hope.ac.uk

Abstract. Robot technology is everywhere from health to retail. It is very rare to see the use of robots in developing countries. It is also a big task for companies to hire staffs for the purpose of monitoring as it requires to work especially in the night. It is also dangers for human. It is a good idea to have some robots for the purpose of monitoring. This paper focuses on the development of surveillance mobile robot using an Arduino board, an ESP32 board and a Raspberry PI system. The robot is a proof of concept of possible robotic application in warehouses where monitoring technologies need to be embedded in the system. In this context, the device is equipped with a servo motor and an on-board camera. The end-user can monitor the robot remotely by wirelessly controlling the position of the servo (i.e. the camera). The camera module is controlled by the Raspberry PI and incorporate a motion detection system which provides an alarm and capture the image frame as soon as the motion is detected. Finally, the image is sent to the end-user. A set of preliminary test of the system has been performed in order to proof the reliability of the proposed prototype.

Keywords: Mobile Robotics, Monitoring Robotics, Inspection Robotics

1 Introduction

Use of robots for the purpose of surveillance has been increasing. Companies are expressing their interests on implementing robots for their business. There are some benefits on using robots for this task, such as, for example, the fact that the robot can work continuously, the maintenance cost could be reasonable, the communication can be embedded in the system and allows the human operator to remotely monitor the plant [1].

In some countries like Asia, companies mainly hire human for the security purpose as they can’t afford to buy a robot. Securities come and secure the property. If that is a large place, they hire multiple securities. They make a schedule between them and walk around the factory or warehouse according to their schedule. For example, if a person walks around the factory this time, another one goes after some minutes and hours.

There are clearly some problems that security systems have to face, such as, for example:

1. Most of the time they walk around the warehouse lonely which is not safe for them.
2. It is very hard for them to walk continuously as they need some rest.
3. Even if they encounter any suspicious activities, it is very hard for them to alert others. Before they alert others, possibility to be attacked.

According to these scenarios, we want to develop a security robotic system, given the demand to make this kind of robot is high, especially in some countries. A literature review has been also performed in order to analyze some existing similar system which are currently available in the market. The following section presents some of these systems.

1.1 Security Robotic Systems

There are different systems for security and surveillance which are currently available in the market: here we present some details of the most significant ones.

Leo-Rover Robot

The Leo-Rover has a cost of £ 3024.79, it is not autonomous vehicle that can be controlled remotely. Its main software is based on Robotics Operating System (ROS) with video streaming feature, thanks to a camera at the front of the vehicle. It is a device suitable for outdoor explorations with a built-in Raspberry PI controller, 4 DC motors and Wi-Fi connectivity and an overall weight 6.5 Kg [2].

The robot is quite big in size. Wheels are properly suitable for outdoor activity, however the camera is fixed and there is no option to turn the camera position and monitor what is happening around the vehicle.

Si-Surveillance Robot

This robot provides video surveillance remotely with a panoramic camera at 360°. It is capable of detecting people up to 100 m in any direction; it also embeds an Artificial Intelligence based system to analyze the contents of the video. The device is suitable for indoor and outdoor explorations with a total weight up to 185 Kg, facial recognition up to 50 meters. The robot can also detect weapons and identify who has the permission to stay inside the premise that you are going to monitor [3].

Some of the features in this robot are interesting with respect to the system which we are going to develop and present here. For example, one of the main purpose of the device could be the surveillance after the opening hours of a warehouse or of a factory. Therefore, it is of interest to have the possibility to detect persons and/ot objects in that practical scenario. Nevertheless, the weight of the SI robot is very high and it maybe more interesting to approach a design which involve a lower weight of equipment.

ID-2868 Robot

This device has a weatherproof chassis for outdoor activities, a camera that covers 360°, infrared cameras for night activities, object detection capability, as well as it can detect number plates and human beings. The robot is autonomous and can plan its path automatically. The robot can also send notifications with photos.

The robot's navigation system is equipped with a LIDAR which helps to adjust the path when it detects obstacles on its way. The main software is based on ROS. One of the main feature of this robot is its autonomous function and navigation capability [4].

ID-2729 Robot

The ID-2729 is a small size device, which can easily reach areas that are not easily accessible to people. A remote access to its camera allows taking pictures and videos while the robot moves. It has a weight of 3.63 Kg and a wheelie bar which provides the ability to climb obstacles without rolling over. A built-in SD card reader allows saving videos while the camera keeps recording even if the wireless communication drops down. Interestingly, it also embeds a swappable battery that can run the robot continuously for about 2 hours [5].

According to the aforementioned characteristics of some surveillance robots, this paper presents the development of a mobile robot for security and surveillance which will be designed to perform its activity on daily time and, moreover, on nigh time.

The proposed system will be characterized by four wheels and a camera in the front. We would also like to integrate a buzzer on the top of the robot (Figure 1).

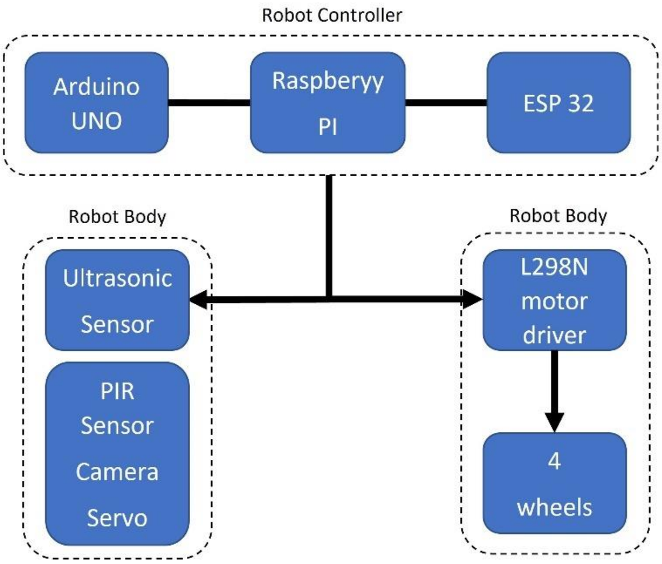

Fig. 1. - Overall architecture of the robot

In order to keep the cost of the system as low as possible, an Arduino Uno will be used to control the wheels by means of a low-cost L298N motor driver.

An Ultrasonic sensor - which is connected to the Arduino board – will also be included, since we want the robot to be able to explore the environment using obstacle avoidance technology.

Finally, a PIR Sensor - which will be connected to a further control board, namely a Raspberry PI, will be embedded. Whether the PIR sensor will perform a motion detection, then the system will trigger the camera (to capture a picture of the scene) and the buzzer (to provide an acoustic alarm feedback).

The end-user will then be able to wirelessly receive the picture over an IEEE 802.11 communication protocol (e.g., wi-fi). Moreover, the camera will be connected to a servo motor which is wirelessly controlled by an ESP32 board. By changing the position of the servo, the orientation of the camera will be changed (Figure 1).

2 Materials & Methods

This section refers to the materials that were used in the project, how the hardware and parts were assembled and organised. A summary about the Programming Code which has been integrated in the system is also reported.

The following parts were adopted for the project:

- Arduino Uno
- ESP32
- Raspberry PI
- Raspberry PI camera module
- 16 GB Micro SD card
- Servo Motor
- 4 DC Motors
- L298N Motor Driver
- HC-SR04 Ultrasonic Sensor
- PIR Sensor (Passive Infrared Sensor)
- LED light
- 4 Wheels
- Buzzer
- Router
- Jumper wires
- Other consumables/lab materials

The Arduino board is used to control the vehicle wheels and the ultrasonic sensor. The ESP32 board is used to control the servo motor position wirelessly over a wi-fi communication protocol. The Raspberry PI is used to detect the motion and then capture a frame and trigger an alert to the end-user by means of a beeping buzzer.

All the microcontrollers in the vehicle are designed to work independently. The main reasons for using such a few microcontrollers are:

- It makes the system reactive (in case of many components connected to a single microcontroller, it may take more time to read all the sensors and then react accordingly)
- It makes the system versatile and it will be easy to add some extra features in the future (such as, for example, an object detection system)
- It simplifies the debugging process and enhance modularity

The development of the project is divided into three main stages, namely

- Step 01 - Communicate to the motors using the Arduino Uno and the Ultrasonic sensor
- Step 02 - Communicate to the servo motor by using the ESP32 board
- Step 03 - Communicate to the Raspberry PI and to the camera by using the Raspberry PI board

All these steps are developed and tested individually, then all the systems are connected to the vehicle and fully integrated at the end. The following paragraphs detail the above-mentioned three integration steps.

2.1 Controlling the Motors & the Ultrasonic Sensor (i.e. the Arduino board)

Motors should not be connected to microcontroller directly. They are then connected to a L298N motor driver. An HC-SR04 Ultrasonic sensor to avoid obstacles is also incorporated. The ENA, ENB, IN1, IN2, IN3, IN4 are the outputs and Motor Terminal 1, whereas Motor Terminal 2 are inputs in the L298N motor controller. Therefore the ENA, ENB, IN1, IN2, IN3, IN4 are connected to the Arduino Uno board, while the Motor Terminal 1 and Motor Terminal 2 are connected to motors.

The design foresees two motors in the left hand side of the robot - which are connected to the Motor 1 Terminal – and two motors in the right hand side - which are connected to the Motor 2 Terminal [6].

Table 1. Motor control strategy.

Robot Movement	Left Motors/Wheels	Right Motors/Wheels
Forward	Forward	Forward
Backward	Backward	Backward
Stop	Stop	Stop
Turn Right	Forward	Backward/Stop
Turn Left	Backward/Stop	Forward

To turn the vehicle to the right or to the left, there are two basic approaches. One is based on turning off one side motors and activate the other side motors to go forward. A second method is based on turning both the side motors in the opposite side (i.e., the tank mode). In this project, second method is used. Table 1 summarizes the motor strategy according to this approach.

The ultrasonic sensor is placed at the front of the robot to scan the environment. The sensor is directly connected to the Arduino Uno board in order to measure how far are the obstacles from the front of the robot. In this project, a threshold distance is fixed at 20 cm. Whenever the distance is more than 20 cm between the robot and any objects, then the robot will keep moving forward. On the contrary, the vehicle will stop moving, it will then go back slightly and then it will randomly choose whether to turn right or left. This approach allows the robot to be unpredictable vs any possible presence which is trying to avoid the robot itself. The following structured code summarises the rational of the approach.

```
Int randNum = rand() % 2; //Generate number 0 or 1
   if (randNum == 0) {     //To turn Right
     turnRight();
     delay(1000);
   }
   else {
     turnLeft();        //To turn Left
 }
```

2.2 Controlling the Servo Motor of the Camera (i.e. the ESP32 board)

A servo motor is connected to the ESP32 microcontroller directly. The reason for using an ESP32 for this step is that the Arduino Uno board does not straightforward integrate a Wi-Fi communication module.

A bracket is also mounted at the end of the servo motor in order to position the camera and the PIR Sensor.

A webserver is available for the end-user by entering the IP address of the ESP32 in the Internet Browser. From the mask, the user can choose whether to turn the camera to the RIGHT or to the LEFT. A static IP address of ESP32 is also set on the board.

Then by means of the following code on the serial monitor, the IP address of the ESP32 board is communicated [7]:

```
#include <WiFi.h>
const char* ssid = "Your Router name";
const char* password = "Your Router's passward";
WiFiServer server(80);

void setup() {
  Serial.begin(115200);
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
```

```
  delay(400);
  Serial.print(".");
 }
 // Print local IP address
 Serial.println("");
 Serial.println("WiFi is connected.");
 Serial.println("IP address;: ");
 Serial.println(WiFi.localIP());
 server.begin();
}
void loop(){
}
```

2.3 Raspberry PI, Buzzer & Camera

The Raspberry PI camera module is connected to a Raspberry PI Zero W board. The PIR Sensor and the buzzer are also connected to the Raspberry PI, such as, in case the PIR sensor will detect any movements, the camera will take a photo and the buzzer will start beeping. Camera and PIR sensor are mounted on a bracket which is connected to servo motor, whereas the Buzzer is placed on top of the vehicle. It is possible to access the Raspberry PI by using a VNC Viewer or PUTTY directly from the computer.

After logging into the board, the user needs to type on the terminal "*sudo raspi-config",* then go to the interface option and enable VNC. A reboot of the Raspberry PI by using the *"sudo reboot"* command enables then the camera. A reboot will be needed every time changes are applied.

Finally, we need to connect one wire of buzzer into the Raspberry PI (at the pin 17) and connect the other wire to the ground. Connect the middle wire of the PIR Sensor at the pin number 4 of the board, the red wire to the 5 V pin, and the black one to the ground pin as well [8, 9]. The following code need also to be implemented

```
from gpiozero import MotionSensor
from picamera import PiCamera
import RPi.GPIO as GPIO
import time

buzz = 17
motion = MotionSensor(4)
GPIO.output(buzz, GPIO.OUT)
cam = PiCamera()

while True:
   motion.wait_for_motion()       //Wait for motion
   GPIO.output(buzz, GPIO.HIGH) //Start the buzzer after detect the motion
   time.sleep(1)
   GPIO.output(buzz, GPIO.LOW)   //Off the buzzer
   time.sleep(1)
```

cam.capture("/home/pi/image2.jpg")   //Capture the photo and save
time.sleep(10)

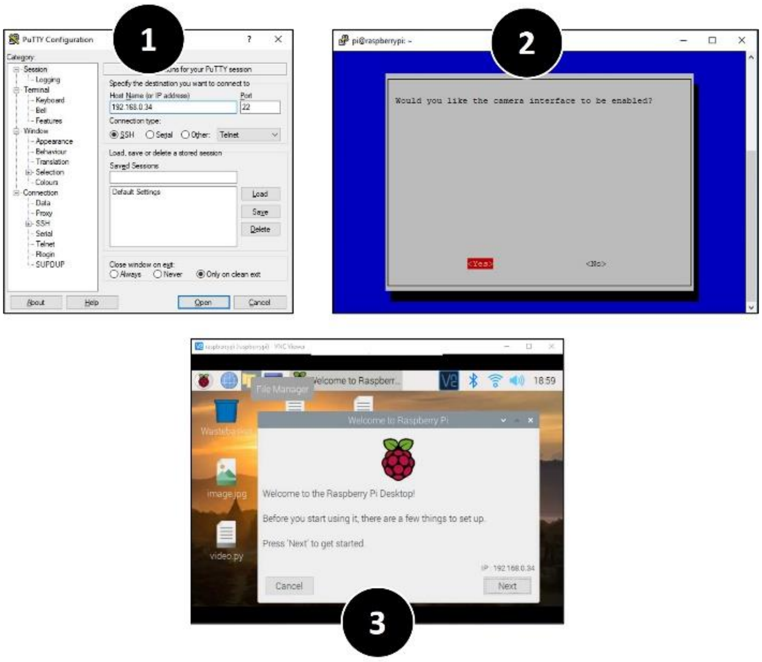

Fig. 2. – Connecting to the Raspberry PI via Putty (1), enabling the camera (2) and logging into the Raspberry PI board (3).

3 Preliminary tests & Results

This section reports all the preliminary trials which were run in order to evaluate the final performance of the robotic system. Initially, the testing was performed on single components, then tests were performed on the overall system. Each session of tests was made of 10 trials.

Test 01

In the current version of the system, the servo motor of the camera has been connected to the ESP32 board. Nevertheless, initially this servo was connected to the Arduino Uno board. This latter board was then taking care of the servo motor, of the Ultrasonic sensor, of the L298N Motor driver, and, finally, of the 4 DC Motors. This configuration made the system pretty low on reacting and, as soon as we performed 10 trials, we always noticed that the system was slow (4/10) if not very slow (6/10). As soon as the servo motor was removed from the Arduino board and connected to the ESP32, performances of the system changed (10/10).

Test 02

We evaluated the performance of Ultrasonic Sensor whether it works well in order to detect obstacles and allow the robot avoiding them. On 10 trials, 9 times (over 10) the system was detecting the obstacle. The failure was due to a blind spot, since the Ultrasonic sensor was positioned at the front of the vehicle. There is a possibility to avoid this issue using a minimum of 3 sensors. In a future development, we should adopt 3 Ultrasonic sensors in the vehicle: one at the front, the second sensor should be 45° to the left with respect to the middle sensor, while the third one should be 45° to the right.

Test 03

In this set of trials, the position of the servo motor was controlled by the ESP32 wireless communication by means of the webserver. The main purpose of this test was to find out whether the servo was reacting as it was instructed to do. For an example, when the “LEFT” button in the webserver was pressed, we checked whether the servo was turning to the right direction. Communication and servo behavior were observed as consistent and reliable over all the trials (10/10).

Test 04

In this section, some tests evaluated the performance of the Raspberry PI board. The main purpose of this test to find out whether it captures a photo and beep the buzzer when it detected a motion. It was necessary for the system to both capture the frame and beep the buzzer in case of the motion detection event. Otherwise the test was reported as a failure. Table 2 reports the result on a set of 10 trials.

Table 2. Testing the camera and the buzzer vs motion detection.

Trial	Camera performance	Buzzer performance
1	success	success
2	success	failure
3	success	success
4	success	success
5	success	success
6	success	success
7	failure	success
8	success	failure
9	success	success
10	success	success

When repeating the test quickly, the buzzer and camera did not respond properly on some tests: this may require some changes in the coding when where the “time.sleep” function is set. Future investigation should be done in order to improve the system on this aspect.

Test 05

In the previous test, the system was checked on taking photo frames. In this test, live video streaming function was added to the system and tested. During these trials, there was a latency in the live streaming. The latency was pretty relevant in the order of 5 s. Because of this latency, the live streaming function was removed from the system. This should be reconsidered in the future development of the system.

Test 06

A final session of trials was focusing on the performance of the whole integrated system. At this stage, all the components were installed into the vehicle and monitored.

The robot was observed in action while detecting obstacles, performing image capturing and beeping in case of motion detection. It was successfully performed 8 times out of 10: in two trials, problems were encountered while the robot was capturing the frame and beeping. In all the trials, the robot was properly avoiding the obstacles.

4 Discussion & Conclusion

The proposed system has multiple microcontrollers and therefore it will be very easy to add extra features to the system. The system also finds its path by scanning the environment throughout its Ultrasonic sensor and the vehicle is substantially moving randomly. In the future it would be of interest to use a better navigation strategy, such as for example *Simultaneous Localization and Mapping* (SLAM) technology. SLAM lets the vehicle to map the unknown environment and acts accordingly [10, 11]. SLAM may also use some information from the sensors of the vehicle, such as, for example, the data from the frontal camera, the wheel revolutions to determine its current location and so on [12]. Clearly there are also potentials in order to increment the visual recognition system of the robot and make it more interactive [13].

The proposed Raspberry PI camera captures a photo when it detects a motion and alert the end-user by beeping the buzzer. At present this buzzer is placed on board of the vehicle, but – in the future – the buzzer maybe also integrated in the device of the end-user.

At present we are providing separate power supplies for each microcontroller, however, in a further step, it would be nice to use one integrated power source for the whole system.

ACKNOWLEDGMENTS

This work was presented in dissertation form in fulfilment of the requirements for the MSc in Robotics Engineering for the student Krisanth Tharmalingam under the supervision of E.L. Secco from the Robotics Laboratory, School of Mathematics, Computer Science and Engineering, Liverpool Hope University

References

1. Guest, P., 2021. Advantages & Disadvantages of Robots | PlastikMedia News. [online] Plastikmedia.co.uk. Available at: <https://www.plastikmedia.co.uk/advantages-disadvantages-of-industrial-robots/> [Accessed 10 October 2021].
2. Robotshop.com. 2021. Leo Rover 4WD Developer Kit (Assembled). [online] Available at: <https://www.robotshop.com/uk/leo-rover-4wd-developer-kit-assembled.html> [Accessed 9 October 2021].
3. series, S., 2021. Deep learning surveillance | Artificial intelligence of a robot | Big data for law enforcement and security | SMP Robotics ®. [online] Smpsecurityrobot.com. Available at: <https://smpsecurityrobot.com/products/ai-surveillance-robot/> [Accessed 9 October 2021].
4. Superdroidrobots.com. 2021. SPAR - Autonomous Indoor/Outdoor Security Robot. [online] Available at: <https://www.superdroidrobots.com/index.php?route=product/product&product_id=2858> [Accessed 9 October 2021].
5. Superdroidrobots.com. 2021. *Security Robots*. [online] Available at: <https://www.superdroidrobots.com/shop/custom.aspx/security-robots/93/> [Accessed 9 October 2021].
6. Teachmemicro.com. 2021. *Account Suspended*. [online] Available at: <https://www.teachmemicro.com/use-l298n-motor-driver/> [Accessed 10 October 2021].
7. Randomnerdtutorials.com. 2021. *ESP32 Servo Motor Web Server with Arduino IDE | Random Nerd Tutorials*. [online] Available at: <https://randomnerdtutorials.com/esp32-servo-motor-web-server-arduino-ide/> [Accessed 10 October 2021].
8. Arduino Project Hub. 2021. *USE a BUZZER MODULE (PIEZO SPEAKER) USING ARDUINO UNO*. [online] Available at: <https://create.arduino.cc/projecthub/SURYATEJA/use-a-buzzer-module-piezo-speaker-using-arduino-uno-89df45> [Accessed 10 October 2021].
9. W, T., 2021. *Time Lapse Raspberry Pi Zero W*. [online] Hackster.io. Available at: <https://www.hackster.io/rjconcepcion/time-lapse-raspberry-pi-zero-w-25da6b> [Accessed 10 October 2021].
10. Resources, A., 2021. *What is Visual SLAM Technology and What is it Used For?*. [online] Automate. Available at: <https://www.automate.org/blogs/what-is-visual-slam-technology-and-what-is-it-used-for> [Accessed 10 October 2021].
11. V. Germanos, E.L. Secco, Formal Verification of Robotics Navigation Algorithms, 19th IEEE International Conference on Computational Science and Engineering, Paris, 177-180, 2016
12. M. Barrett-Baxendale, M.J. Pearson, M. Nibouche, E.L. Secco, A.G. Pipe, Self-adaptive context aware audio localization for robots using parallel cerebellar models, Towards Autonomous Robotic Systems (TAROS), Lecture Notes in Computer Sciences, Chapter 6, 1-13, 2017, DOI:10.1007/978-3-319-64107-2

13. D. McHugh, N. Buckley, E.L. Secco, A low-cost visual sensor for gesture recognition via AI CNNS, Intelligent Systems Conference (IntelliSys) 2020, Amsterdam, The Netherlands