The 4th International Conference on Control and Robotics Engineering

The Design and Implementation of GPS Controlled Environment Monitoring Robotic System based on IoT and ARM

Hasan Salman, Md Sezadur Rahman ,Md Abu Yousuf Tarek
College of Overseas Education
Nanjing University of Posts and Telecommunications, Nanjing 210023, China
e-mail: hasansalman@ieee.org, rahmansezad@gmail.com, yousuf_tarek@yahoo.com

Jun Wang
College of Communication and Information Engineering
Nanjing University of Posts and Telecommunications, Nanjing 210023, China
e-mail: wang_jun@njupt.edu.cn

***Abstract*—Environmental monitoring systems are often designed to measure and log the current status of an environment or to establish trends in environmental parameters. In this paper, We proposed an autonomous robotic system that is designed and implemented to monitor environmental parameters such as temperature, humidity, air quality, and harmful gas concentration. The robot has GPS coordinates, and it can store data on the ThingSpeak IoT platform. The mobile robot is controlled by a smartphone which runs an app built on the Android platform. The whole system is realized using a cost-effective ARM-based embedded system called Arduino and Raspberry Pi which communicates through a wireless network to the IoT platform, where data are stored, processed and can be accessed using a computer or any smart device from anywhere. The system can update sensor data to IoT server every 15 seconds. The stored data can be used for further analysis of the reduction of pollution, save energy and provide an overall living environment enhancement. The robotic system has designed for cost-effective remote monitoring environmental parameters without any human intervention to avoid health risk efficiently. A proof-of-concept prototype has been developed to illustrate the effectiveness of the proposed system.**

Keywords— ARM and Embedded system, Raspberry pi, Arduino, ThingSpeak, IoT, Air quality, Gas Sensor, GPS, Android App, Remote Monitoring..

I. INTRODUCTION

Environment monitoring is the collection of data and information on environmental parameters. Monitoring and evaluating the health of our natural resources is also essential for effective environmental planning, policymaking and solving environmental pollution. For the extremely polluted region, it carries the health risk for monitoring manually. To avoid these risks, remote monitoring techniques along with a robotic system that has intelligent data acquisition, communication and processing are crucial in revolutionizing monitoring and protection. For remote monitoring, developing a system will be an efficient solution so that the monitoring can be done without any human intervention.

Recently, robotic systems are utilized as data-gathering tools by scientists for a greater understanding of environmental processes [1]. Robots are also being designed to explore areas with harmful gases, monitor climatic conditions, and to study about a remote place that is quite risky for the human [2]. Keeping the above statement in the forefront, the new trending wireless sensor and ARM-based embedded system technology are getting integrated on a single board, intended towards the advancement of this system. The core part of our designed system is based on the ARM (Acorn RISC Machine) which presents a high-cost performance, code density, excellent period interrupt response and low electricity consuming with a small piece of a silicon chip. Specifically, the ARM is an ideal option for the embedded system that might assume additional significant functions while other simple SCM (Single Chip Micyoco) cannot, as an instance, The Raspberry Pi 3 model B includes Broadcom BCM2837 64-bit ARMv8 Quad-Core Processor powered Single Board and Arduino. It also has enough pins for GPIO and serial communication pin that can be connected to the number of sensors. All those benefits make ARM the most effective selection for completing the system [3].

In order to deploy a scalable and remote monitoring system, an efficient platform that enables users to monitor their daily exposure to air pollutants by giving air quality information provided by various sensing infrastructure is proposed. The sensors periodically monitor air quality. The data can be monitored and accessed from anywhere using mobile phones or PC with Internet access. The implementation has sensors for air quality, CO, CO2, and temperature and humidity to monitor the environment around. The Raspberry Pi has been used to interact with the IoT platform and sensors. The Arduino Mega microcontroller is used for control and navigation of the robot. The system has been developed by python and embedded C programming language. The robotic system with GPS controlled feature enables to move according to user's instruction autonomously and collects sensor data from targeted locations. An Android app has been developed for the user-friendly interface. All collected data is sent to the ThingSpeak [4] IoT platform in order to be accessed by the

978-1-7281-1593-1/19/$31.00 ©2019 IEEE

user from a wireless connection. Real-time cloud graphical visualization is performed to analyze the collected data. This multipurpose robotic system is capable of remote monitoring without any human intervention and keeping away environmental hazard risks.

II. RELATED WORK

Existing environmental monitoring systems discussed in this section, with a focus on environmental sensors, robotic systems, as well as IoT, are also reviewed to clarify the essence of this work.

A. Cyber-physical system for environmental monitoring

With recent advances in wireless sensor technology, low power single-board computers, and short-range communication technologies, remote sensing applications have improved towards solutions that encompass ubiquitous computing. A Cyber-Physical device was once proposed for environmental monitoring of ambient stipulations in indoor spaces [5].

B. Climate monitoring using Raspberry Pi

Shete., R. and Agrawal S. [6] presents the framework for monitoring the metropolis environment. Low-cost Raspberry Pi used for implanting the system. However, no emphasis has given on particulate matter which left the environment monitoring system incomplete.

C. Cloud-based smart device

Biao Jiang and Christian F. Huacón developed a Cloud-based Environment Monitoring Smart Device (CEMSD) [7] that monitors different environmental parameters such as air quality, noise, temperature, and humidity. The device collects and sends data from targeted measurement locations through a wireless network or cellular network to a cloud server.

D. Motivations of this Paper

The discussion mentioned above merely lists a few of the numerous options that have been proposed for remote monitoring of environmental conditions, mainly using wireless sensing methods, GPS, robotics, IoT-based technologies. However, most of these solutions only address data collection and data observation. In order to tackle the problem of remote environment monitoring with avoiding health risks, it is imperative that the system should collect data via a self-sufficient robotic system and applicable observations transfer to a cloud server remotely.

III. SYSTEM ARCHITECTURE

The proposed robotic system has designed incorporating the embedded hardware, software, and IoT components. The system architecture is shown in fig. 1 which illustrates the block diagram of the IoT and ARM-based embedded robotic System. The whole robotic system has two parts:

1. Environment Monitoring System: This system is responsible for collecting data from the sensor and uploading collected data to the IoT platform.

2. Navigation and Control System: The primary function of this system is to navigate and control the movement of the robotic system according to the instruction from the app.

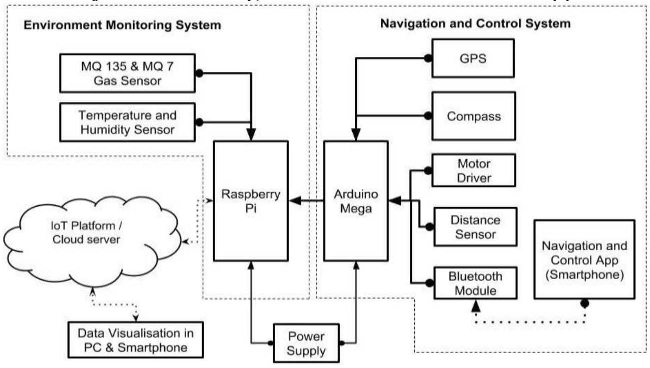

Figure 1. The block diagram of proposed system design.

The environment monitoring system utilizes the Raspberry Pi to communicate with three sensors such as the DHT11, the MQ135, and MQ7 Gas sensor. The Raspberry Pi collects sensor data from targeted locations and uploads data into the IoT platform which can be accessed directly by the user. The robot is part of the Internet of Things because it requires network connectivity through a GPRS module connected with raspberry pi, it uses sensors to collect environmental parameters data, so the system needs minimum human intervention. The robot navigation and control system consists of an Arduino mega microcontroller, GPS and Compass module, DC motor and robot chassis. Arduino motor shield with L293D motor driver has been used with Arduino to control the DC motor. The navigation system utilizes the Arduino to communicate with the GPS module and the compass to navigate and move in a fixed path from the initial location to the destination. For accurate navigation, We added an ultrasonic distance sensor so that it can avoid obstacles. An App has been developed to give the instruction which communicates through the Bluetooth connection.

IV. HARDWARE COMPONENTS

A. Raspberry Pi 3B:

A Raspberry Pi 3B (RP) is an ARM-based single board computer. It has Broadcom BCM2837 64bit ARM Cortex-A53 Quad Core Processor SoC running at 1.2GHz and 1GB RAM. It has 40 GPIO pins used for the general purpose [8]. Additionally, it adds wireless LAN and Bluetooth connectivity making it the ideal solution for powerfully connected designs.

B. Arduino Mega:

The Arduino Mega is a microcontroller board based on the ATmega2560 containing 54 digital input/output pins, 16 analog inputs, 4 UARTs (hardware serial ports). ATmega2560 is our choice as it is an easy option for use in prototyping with ease in robotics. [9].

C. *Environment Sensing*

The robotic system uses sensors such as the DHT11 Temperature and Humidity Sensor, MQ-7 Carbon-monoxide Gas Sensor and MQ135 Air Quality Sensor. The system also can monitor environmental parameters such as CO2 and smoke as well. The DHT11 is a digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air and spits out a digital signal on the data pin [10]. The air quality sensor is the MQ-135 sensor for detecting toxic gases that are present in the air. Being a very dangerous gas Carbon monoxide (CO) is odorless, colorless, so it cannot be smelt, seen, or tasted which makes it difficult to detect. MQ7 Carbon Monoxide (CO) gas sensor detects the concentrations of CO in the air and outputs its reading as an analog voltage anywhere from 20 to 2000ppm. This sensor has a high sensitivity and fast response time. There is a screw potentiometer that enables manual adjustments to the output gain of the sensor. The ADS1015 is used to convert analog data to digital data received from both sensors.

D. *Navigation and Control Hardware*

For navigation, the system requires the GPS module U-box NEO-6M, and the compass HMC5883L with Arduino mega. Global Positioning System (GPS) is a satellite-based navigation system that provides critical positioning capabilities to the robot. Along the way, The HC-SR04 ultrasonic distance sensor is used for obstacle detection. L293D motor shield controls the robot's movement according to its navigation. The HMC5883L is a triple axis magnetometer that uses the basic principle behind electromagnets to sense magnetic north pole. This compact sensor uses I2C to communicate. The L293D, A motor driver is an integrated circuit chip which is usually used to control motors in robots.

E. *ThingSpeak*

Open-source IoT platform ThingSpeak is used to collect and store sensor data in the cloud and helps in the development of IoT application [11]. Read and write API keys were generated on ThingSpeak. The Raspberry Pi connected with Internet-enabled GPRS module that sends the data value from the sensors to the IoT platform. ThingSpeak performs real-time visualization by using MATLAB software. The data can also be extracted directly from the platform, and anyone can process and visualize the information using any statistical software.

F. *Network System hardware:*

The system uses an HC-06 Bluetooth module for building personal area networks (PANs). It has been connected to Arduino to communicate with the App. SIM800L GPRS module allows raspberry pi to connect with wide area network (WAN) using a corresponding pin connection. The technique used for communication with the peripheral device in this project is I2C (Inter-Integrated Circuit) [12].

V. Hardware Design

The robotic system is built by using mechanical and electrical components, and it is of the wheeled type. The Arduino mega serves as the central part for the navigation and control system. GPS module is mounted on the Arduino to tag the GPS coordinates of the robot's position. The robot is equipped with an ultrasonic sensor to measure the distance between the robot and an obstacle. The system moves in a fixed path according to the app's instructions. Fig. 2 shows the circuit diagram of the whole system. Two DC motors are used to control the movement of the robot. HC-06, HMC588L, and Neo 6M GPS Modules are connected to the Arduino. Note the HMC588L compass should be placed away from any ferromagnetic element; otherwise, it may misdirect the robot.

For this reason, the compass has been placed in a surface made of plastics and wood that is away from the main body which is made of a ferromagnetic element. This ensures the compass to get the accurate direction of the robot. The GPS module searches for at least four satellite and gets a GPS location. After getting a GPS location, it follows the program flow of the embedded C code and moves to its destination GPS location that will be given to the robot by App. For all modules VCC is 5V, and the GND is connected to the GND on Arduino board. Rest of the pin connections are followed from table 1. In the design of the environment monitoring system, the most critical part is to sense analog value from the environment. The Raspberry Pi supports I2C serial communication and only digital input. To solve this issue, the ADS1015 ADC has been utilized which supports I2C communication. The GPRS module is attached with Raspberry Pi ensuring connection with WAN. Using API key provided by the ThingSpeak, the Raspberry Pi follow the program flow of fig. 3 and starts uploading sensor data using internet connection. The GPRS module requires a SIM card for internet service. The sensors MQ7, MQ135, DHT11, should be connected through ADS1015 with raspberry pi. For all modules VCC is 5V, and the GND is connected to the GND on board. Rest of the pin connections followed from Table 2.

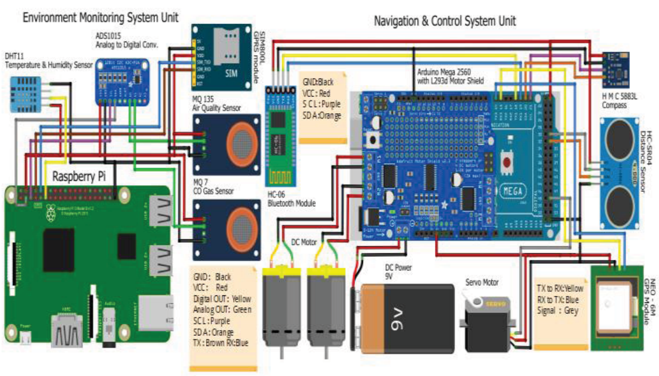

Figure 2. Circuit diagram of the system

TABLE I. MODULE'S PIN CONNECTIONS WITH ARDUINO

Module	Connection
HC-06 Bluetooth Module	Tx to pin 18 and Rx to pin 19
HMC5883L Compass	SDA to pin 20 and SCL to pin 21
Neo 6M GPS Module	Tx to pin 15 and Rx to Tx pin 14
HC SR-04 Distance Sensor	Trig to pin 31 and Echo to pin 33
Servo and DC Motor	Data line to pin 49 & M1 , M2

TABLE II. MODULE'S PIN CONNECTIONS WITH RASPBERRY PI

Module	Connection
MQ-7	Analog out to pin A1 on ADS1015
MQ-135	Analog out to pin A0 on ADS1015
SIM800L GPRS	TXD to pin 10 & RXD to pin 9
ADS1015	SCL to pin 5(SCL1, I2C)
DHT11	SDA to pin 3(SDA1, I2C)

VI. SOFTWARE DESIGN & PROGRAMMING

A. Android App Development Tools

MIT AppInventor is a visual, block-based development environment which does not require any prior programming knowledge. We choose MIT AppInventor as the android application development tool. MIT and Google jointly develop AppInventor [13], and it has gained popularity as a learning tool and a way by which students can practice creative innovations.

B. Embedding Code in Raspberry Pi

For the realization of the environment monitoring system in software, the python code is used according to the program flow of fig. 3 to program raspberry pi. The primary function of the code is to collect sensor data and upload the data to the IoT platform using API keys. At first, the system initializes libraries and modules of that is needed to run some built-in function. At the same time, we stored the API key as a variable. Then I2C communication has been enabled so that it can communicate with ADC pins to get analog value from the sensor. After that, we have defined one function to read ADC value. The ADC pin will read sensor data and store that value in a variable. Another function has been defined to upload sensor data using the API key variable to the IoT platform. Finally, both functions are called in a loop which repeats calling the function if the internet is available.

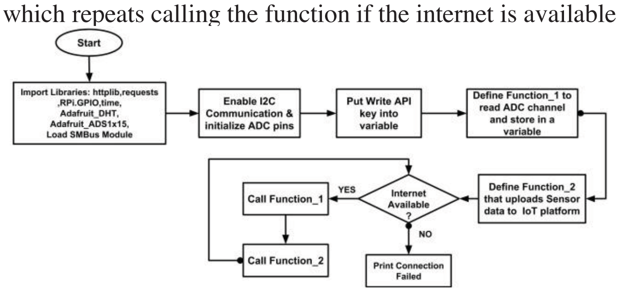

Figure 3. Program flow of the sensors data collection and storage to the IoT platform.

C. For Embedding Code in Arduino Mega 2560:

The Arduino mega is programmed with embedded C language in an integrated development environment (IDE). In fig. 4, the program flow of the embedded C code is given. At first, the system initializes libraries and modules of that is needed to run some built-in function. At the same time, the system starts serial communications and I2C bus with Bluetooth, Compass and GPS module. At first, the GPS module starts looking for and acquire at least four satellite to get a GPS coordinate. After receiving the current GPS location, the system gets compass heading of the robot. After that, the user can set the GPS coordinate as a waypoint that the robot will follow to move. The waypoints are stored in an array so that the program can use them to provide the robot with five locations of which a waypoint array declares, and a path is planned according to a series of waypoints. The app can perform all of these steps. After setting the waypoint, we have to press "done" for confirmation in the app. Next, the system waits for Bluetooth signal for checking the latest GPS information, going through the waypoints or resetting the stored waypoint. After receiving "go to waypoint" command from the app, the robot begins moving directly to the first waypoint. During the movement to the destination, if it receives a "stop driving" command, it will freeze the robot immediately. Unless the stop signal is received the robot continues to update compass heading and GPS information as well as moving towards the destination.

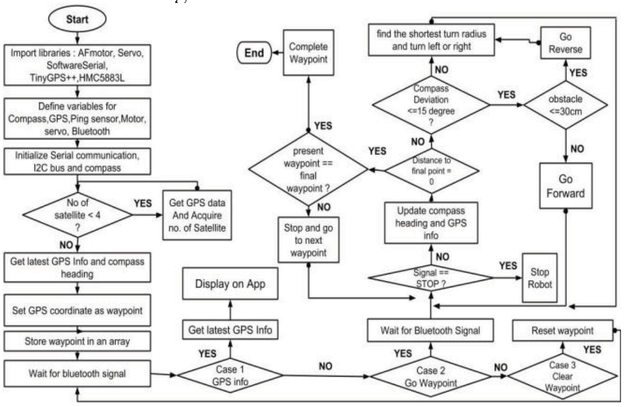

Figure 4. Program flow of the embedded C code .

The robot keeps going until the distance to destination equals zero. Moreover, the behavior control algorithm is employed to keep the robot on the right path for less than 15-degree deviation. For the deviation of fewer than 15 degrees, it prompts to avoid obstacles in less than 30 cm. On the contrary, the deviation of more than 15 degrees it adjusts its course or accurate pathfinding the shortest turn radius and turns left or right. When so with no obstacles detected within 30 cm the robot goes forward towards the destination. When obstacles found, again it finds the shortest turn radius and turns left or right to reach the destination. Finally, when it reaches a waypoint, the user is notified about the present location and checks if this is the final waypoint. If it is not the final waypoint, the robot stops and start to move toward the next waypoint. During this, all these procedures repeat until the final waypoint obtains. Finally, after reaching the final waypoint, the robot completes its ride. Along with all these things, there is also an option to clear the previous set waypoints to reset the robot and start the whole operation once again.

VII. RESULT

Fig. 5 shows the complete prototyping of IoT and ARM-based GPS controlled environment monitoring robot and Navigation and Control app. The prototype can work effectively in remote places to collect data, alone or in teams. The proposed system is quite cost-effective when compared with other existing methods [14-16] that require more number of hardware accessories. Fig. 6 shows the robot is capable of collecting and uploading environmental data to the ThingSpeak, IoT platform server efficiently. The field updating time at channel takes a minimum of 15 seconds. The sensor data stored in the platform can be used for visualization and analysis of the environmental parameters. Fig. 6 is also showing the carbon monoxide and carbon dioxide gas sensor data in ppm, temperature sensor data in degree centigrade and humidity sensor data in percent relative humidity (%RH), respectively.

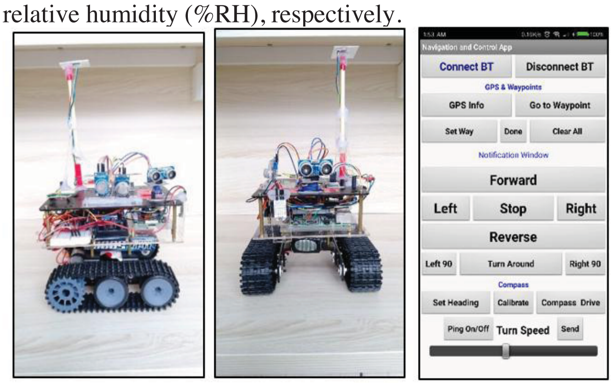

Figure 5. The complete prototype of IoT and ARM-based robotic system (right and front views) and Navigation and Control app for the robot (right side).

Figure 6. Real-time graphical visualization of the environmental parameters.

VIII. CONCLUSION

In this present work, design, and implementation of a GPS controlled robot for environmental parameters monitoring based on IoT and ARM have been accomplished. The developed ARM-based embedded system with the IoT platform can monitor the environmental parameters, and the measurement of air quality is compact and cost-effective. The results obtained are found to be useful for monitoring real-time environmental conditions. The developed App allows the user to control and navigate the robot easily. The GPS controlled feature allows it to travel autonomously to the remote places and submits the collected data to the IoT server as well as displays it on the web for a high-level data analysis and processing. Graphical visualization evidence shows that the robotic system works efficiently.

Moreover, the key advantages of the system are The intuitive user interfaces in the App and Autonomous movement after getting instruction from the user. Also, the system is cost-effective, and the costs are less than 80 USD. It updates sensor data to IoT server in every 15 seconds. Secured data in IoT platform and can be accessed from anywhere of the world. Future work includes several features including solar power, advanced communication solutions for rural areas. The system can be modified to detect radiation and even other kinds of harmful gas autonomously to avoid human health risks. Also, the design method can also be applied in drone technology to make it even more dynamic.

ACKNOWLEDGMENT

Special thanks for the guidance to our supervisor Associate Professor Jun Wang.

REFERENCES

[1] M. Trincavelli, M. Reggente, S. Coradeschi, A. Loutfi, H. Ishida and A. J. Lilienthal, "Towards environmental monitoring with mobile robots," 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, 2008, pp. 2210-2215.

[2] M. Dunbabin and L. Marques, "Robotics for Environmental Monitoring [From the Guest Editors]," in IEEE Robotics & Automation Magazine, vol. 19, no. 1, pp. 20-23, March 2012.

[3] Liu, F.H.: Research and Implementation of WiFi Wireless Communication Terminal Based on ARM. Wuhan University of Science and Technology, China (2010).

[4] The MathWorks, Inc, “Understand Your Things” Internet:https://thingspeak.com/

[5] G. Mois, T. Sanislav and S. C. Folea, "A Cyber-Physical System for Environmental Monitoring," in IEEE Transactions on Instrumentation and Measurement, vol. 65, no. 6, pp. 1463-1471, June 2016.

[6] R. Shete and S. Agrawal, "IoT based urban climate monitoring using sSignal Processing (ICCSP), Melmaruvathur, 2016, pp. 2008-2012.

[7] B. Jiang and C. F. Huacón, "Cloud-based smart device for environment monitoring," 2017 IEEE Conference on Technologies for Sustainability (SusTech), Phoenix, AZ, 2017, pp. 1-6.

[8] Grimmett, Richard. Raspberry Pi Robotic Projects. Third Edition, Packt Publishing, 2016.

[9] M. Schmidt, 2011, “Arduino: a quick start guide”, The PragmaticBookshelf, Raleigh, NC

[10] A. Industries, "DHT11 basic temperature-humidity sensor + extras ID: 386 - $5.00: Adafruit Industries, Unique & fun DIY electronics and kits." Internet: https://www.adafruit.com/product/386, 2017.

[11] A. Tapashetti, D. Vegiraju and T. Ogunfunmi, "IoT-enabled air quality monitoring device: A low cost smart health solution," 2016 IEEE Global Humanitarian Technology Conference (GHTC), Seattle, WA, 2016, pp. 682-685.

[12] R. Shantha Selva Kumari; C. Gayathri “Interfacing of MEMS motion sensor with FPGA using I2C protocol” 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), Coimbatore,2017,pp.1-5.

[13] http://appinventor.mit.edu/explore/sites/all/files/Resources/Thesis_FINAL_AnshulBhagi.pdf

[14] S. Zhi, Y. Wei, Z. Cao and C. Hou, "Intelligent controlling of indoor air quality based on remote monitoring platform by considering building environment," 2017 4th International Conference on Systems and Informatics (ICSAI), Hangzhou, 2017, pp. 627-631.

[15] WenJieTian and Y. Geng, "Environment Monitoring System of Household Security Robot Based on Wireless Mesh Network," 2009 International Conference on Networks Security, Wireless Communications and Trusted Computing, Wuhan, Hubei, 2009, pp. 176-180.

[16] L. R. Adrian and D. Repole, "Intelligent autonomous environmental monitoring based on the AMBOA robot sensory system," 2017 IEEE 58th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON), Riga, 2017, pp. 1-6