ABNUS JOURNAL
Your Research Partner

Jurnal Teknika dan Informatika (JTI)
E-ISSN: 2797-0000 | Vol. 1, No. 1 (2021) | pp. 1–18

IP Camera Surveillance System Using an Android Application Based on Arduino

Belinda Ayuningtyas[1*], Syaeful Ilman[2]

[1,2] *Faculty of Engineering and Informatics, Electrical Engineering Study Program, Dian Nusantara University, Jakarta, Indonesia*

***Corresponding author:** ayuningtyas@undira.ac.id |

Received: 3 March 2021 | Revised: 15 April 2021 | Published: 16 May 2021

Abstract

Purpose: This study designs an IP camera surveillance system using motion detection with a PIR sensor, integrated with an Android application for real-time monitoring. The system aims to improve security by notifying users of detected movement and allowing them to view captured images.
Research Methodology: The system utilizes an Arduino microcontroller, Ethernet shield, IP camera, PIR sensor, and Android application. A quantitative approach was employed to test the effectiveness of the system's components. The system detects motion, sends notifications to an Android app, and uploads images to a website using FTP.
Results: The system successfully detects motion within a 480 cm range using the PIR sensor, triggering an alert on the Android application. The IP camera captures images when motion is detected and uploads them to a website, where users can access and download them via FTP links. The system demonstrated real-time monitoring capabilities through the Android app.
Conclusions: The surveillance system effectively enhances security by allowing real-time monitoring and image capturing. It demonstrates how Arduino and IP cameras can be combined for an affordable, functional security system. However, the system's data transfer speed was limited when using cellular routers, affecting performance.
Limitations: This study is limited by the reliance on a single test environment and the slow data transfer speed using cellular routers. Future research could address these limitations, improve data transfer efficiency, and explore broader applications.
Contributions: The study contributes to developing affordable DIY surveillance systems, providing a practical approach for security monitoring using accessible technologies like Arduino and IP cameras.

Keywords: *Brand Image, Client Satisfaction, Service Quality, Trust*

How to Cite: Belinda, A., & Ilman, S. (2021). IP Camera Surveillance System Using an Android Application Based on Arduino. *Jurnal Teknika dan Informatika (JTI)*, *1*(1), 1–18. https://doi.org/10.52909/jti.v1i1.6

1. Introduction

Security is a major concern for everyone, especially for those who have valuables to protect. Various methods are used to prevent theft, including storing valuables in secret rooms, using coded doors, safes, and surveillance cameras (Tao et al., 2019). Closed-circuit television (CCTV) is a type of surveillance camera that is often used because it can record anything monitored within range and store the recordings in a Digital Video Recorder (DVR) located in the control room. However, if a theft occurs, the owner will only find out after the theft has occurred because the owner cannot monitor the room in real-time (Welsh et al., 2020). The worst scenario is if the thief takes the DVR from the control room, then his actions will be difficult to trace because all the stored recordings will be lost (Ashby, 2017). As an alternative to this problem, another type of surveillance camera can be used, namely, an IP camera. An IP camera is a surveillance camera that connects directly to the Internet (Jang et al., 2018; Piza et al., 2019). It does not require a control room for monitoring, as it uses a Network Video Recorder Recorder (NVR) as a storage medium. However, IP cameras do not store videos but images. With NVR storage, owners do not lose data recorded by IP cameras. Furthermore, IP cameras have a motion detection feature that can detect movement, capture it, and then The resulting images are sent to the web, where they can be accessed directly by the owner (Muthusenthil & Kim, 2018; Wu et al., 2017).

In this study, a security system was created utilizing the Motion Detection feature of an IP camera. Additionally, a PIR sensor was used to detect motion and was connected to an Arduino Uno microcontroller and Ethernet shield. This system also utilizes an Android smartphone application to generate alerts. If the PIR sensor detects motion, a notification appears via the app on the Android smartphone. This notification is a web link to download images captured by the Motion Detection feature of the IP camera. This allows the owner to monitor the room in real time.

2. Literature Review

2.1 Arduino Microcontroller

The Arduino microcontroller is an open-source, single-board microcontroller designed to simplify electronics applications in a wide range of fields (Furter & Hauser, 2019). Its hardware uses an Atmel AVR processor, and its software uses a proprietary programming language. Arduino is an open hardware platform aimed at anyone who wants to create interactive electronic devices based on flexible, easy-to-use hardware and software (Nayyar, 2016; Ray, 2017). This microcontroller was programmed using the Arduino programming language, which has a syntax similar to that of the C programming language. Because of its open nature, anyone can download Arduino hardware schematics and build them. Arduino uses the Atmel ATMega family of microcontrollers as its base, but some individuals and companies have created clones using other microcontrollers while maintaining the hardware compatibility (Martinez-Santos et al., 2017). For flexibility, programming was performed through the bootloader, although there was an option to bypass the bootloader and use a downloader to program the microcontroller directly through the ISP port. Similar to many types of microcontrollers, Arduino was born and developed, and then appeared with various types (Chaudry, 2020). Among them are:

1. Arduino Uno
   Arduino Uno is a circuit board based on the ATmega328 microcontroller. This integrated circuit (IC) has 14 digital inputs/outputs (six outputs for PWM), six analog inputs, a 16 MHz ceramic crystal resonator, a USB connection, an adapter socket, an ICSP header pin, and a reset button. This is what is needed to support the microcontroller easily connected with a USB power cable or an AC-to-DC adapter power supply cable or even a battery is used (Tt & Oo, 2018).

Figure 1. Arduino Uno Main Board

Figure 1 show the Arduino UNO R3 ATmega328 differs from all earlier microcontroller boards in that it does not use a dedicated FTDI USB-to-serial driver chip. Instead, the USB-to-serial implementation is the ATmega16U2 version R2 (previously ATmega8U2). The Arduino Uno Rev.2 version includes a resistor to the 8U2 to ground line, making it easier to place into DFU mode.

Table 1. Arduino Uno Rev.2

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g

Based on Table 1, the Arduino Uno operates with a 5V voltage, and its recommended input voltage is between 7V and 12V, with a maximum range of 6V to 20V. It has 14 digital I/O pins, 6 of which support PWM output, and 6 analog input pins. The board provides a maximum current of 40 mA per I/O pin, with 50 mA available for the 3.3V pin. It is equipped with 32 KB of flash memory (0.5 KB used by the bootloader), 2 KB of SRAM, and 1 KB of EEPROM. Operating at a 16 MHz clock speed, it measures 68.6 mm in length and 53.4 mm in width, weighing 25 g. The Arduino Ethernet Shield, which connects the Arduino board to the internet, uses the Wiznet W5100 Ethernet chip, expanding the board's capabilities for networked applications.

2. Arduino Ethernet Shield
   The Arduino Ethernet Shield is a module that connects the Arduino board to the Internet, as it is based on the Wiznet W5100 Ethernet chip (datasheet).

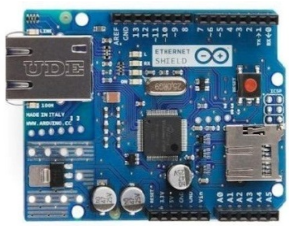

Figure 2. Arduino Ethernet Shield

Figure 2 shows the Arduino Ethernet Shild. To connect and use the module so that it can connect to the Internet is quite easy; it only takes a few minutes. The method is to install the module on the Arduino board, connect it with an RJ45 network cable, follow the programming tutorial (using the

Ethernet library that is already available in the Arduino IDE software package), and the Arduino is ready to be controlled via the Internet. In the Arduino Ethernet itself, there is a micro SD slot that functions as a file storage place, while to access the micro SD card, the SD library is used. For the type of Arduino board that can be paired with the W5100 Ethernet Shield, namely Arduino Uno and Mega (Kyeremeh & Peprah, 2017).

2.2 IP Camera

An IP camera is a type of digital video camera commonly used for security monitoring that can send and receive data over a computer network and the Internet. Although webcams can also perform this function, the term "IP camera" is used. "IP Cameras" or "Network Cameras" are typically used only for security surveillance systems. The first IP cameras were used in 1996 (Bhuyan & Hasan, 2020).

Types of IP cameras:

1. Fix IP camera.
2. Dome IP Camera.
3. PTZ (pan, tilt, zoom).

Figure 3. Physical form of CCTV (IP Camera)

Figure 3illustrates the physical forms of various types of IP cameras, including fixed, dome, and PTZ (pan-tilt-zoom) cameras. These cameras are commonly used in surveillance systems, with each type offering unique advantages based on their specific application, such as fixed cameras for focused monitoring or PTZ cameras for dynamic coverage. Essentially, connecting an IP camera to a network is not much different from connecting other devices to a computer network. First, it must have an IP address. The method for assigning (setting) an IP address varies by the brand. Instructions can be found in the respective user's manual (Lahfaoui et al., 2017).

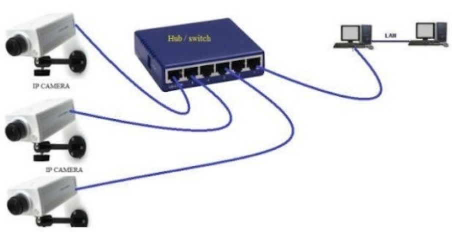

Figure 4. Example of an IP Camera Hub Network

Figure 4 demonstrates an example of how multiple IP cameras are connected in a network. The figure shows a hub that connects various cameras to a central system, allowing them to be monitored and controlled remotely. This setup highlights how the cameras are linked through an Ethernet connection, ensuring the easy transfer of video data to a centralized monitoring system. The easiest way to access an IP camera is through a web browser, such as Mozilla, Internet Explorer, or Google Chrome. The camera's IP address was typed into the web browser, and a simple menu appeared to activate the IP camera.

2.3 Android

Android is a Linux-based operating system designed primarily for touchscreen mobile devices, such as smartphones and tablet computers. It was originally developed by Android, Inc., with financial backing from Google, which acquired it in 2005. The operating system was officially released in 2007, coinciding with the founding of the Open Handset Alliance, a consortium of hardware, software, and telecommunications companies dedicated to advancing open standards for mobile devices. The first Android phone was sold in October 2008 (Hussain et al., 2018; Thakare et al., 2016).

The Android user interface is primarily direct manipulation, using touch gestures that mimic real-world actions, such as swiping, tapping, and pinching, to manipulate on-screen objects and a virtual keyboard for typing text. In addition to touchscreen devices, Google has developed Android TV for televisions, Android Auto for cars, and Android Wear for watches, each with its own distinct user interface. Variants of Android are also used in laptops, game consoles, digital cameras, and other consumer electronics devices (Maw et al., 2019; Patil et al., 2017).

Android is an open-source operating system, and Google releases its code under the Apache License. Android's open-source code and licensing license allow software to be freely modified and distributed by device manufacturers, wireless carriers, and app developers. Furthermore, Android has a large community of developers developing apps (apps) that extend the device's functionality, typically written in a customized version of the Java programming language. As of October 2013, there were over one million apps available for Android, and approximately 50 billion apps had been downloaded from Google Play is Android's primary app store. A survey in April–May 2013 found that Android was the most popular platform for developers, used by 71% of mobile application developers. At Google I/O 2014, Google reported over one billion monthly active Android users, up from 583 million in June 2013 (Abdulaali Jasim et al., 2020; Riadi et al., 2018).

2.4 PIR Sensor

A passive infrared (PIR) motion sensor functions as a motion detector that detects differences/changes in current and previous temperatures. Motion sensors using a PIR module are very simple and easy to apply because the PIR module only requires a DC 5V input voltage, which is effective enough to detect movement up to a distance of 5 meters. When it does not detect when motion is detected, the module output is LOW. When motion is detected, the output changes to HIGH. The HIGH pulse width is ±0.5 seconds. The PIR module's sensitivity, which can detect motion at a distance of 5 m, allows for a more effective motion detector (Mukhopadhyay et al., 2018; Yang et al., 2018).

Figure 5. PIR (Passive Infra Red) Motion Sensor Shape

Figure 5 shows the shape of a PIR (Passive Infrared) motion sensor. This sensor detects motion by identifying changes in infrared radiation within its environment. The sensor has two logic output levels: Low and High. When motion is detected, the output switches from Low to High, with a pulse width of approximately ±0.5 seconds. The PIR module's sensitivity allows it to detect motion from a distance of up to 5 meters, making it an effective motion detection solution. This type of sensor is commonly used in applications such as security systems, lighting activation, and automated systems, where it can trigger actions such as turning on lights or activating alarms when motion is detected.

With outputs that only provide two logic levels, High and Low, a variety of motion sensor applications can be created. For example, to directly apply it to an alarm, a driver circuit is created to activate the alarm (Wu et al., 2018). Or, if you want to use it to activate a light, simply create a driver to provide a voltage source for the light. The PIR motion sensor module has outputs that can be directly connected to TTL or CMOS digital components and can also be directly connected to microcontrollers (Welbourne et al., 2016; Wu & Wang, 2018).

The effectiveness of motion detection using this motion sensor is influenced by the placement of the PIR sensor. The motion The sensor must be placed in a location that can detect all movements in the room or area being monitored by the PIR motion sensor (Besteiro et al., 2018).

3. Methodology

3.1 System Block

His section discusses a monitoring system specifically for an empty room with a safe inside. This system uses an IP camera as the main input, and a Motion Detection feature on the IP camera can detect movement. If the feature detects movement, the IP camera captures an image of the movement and sends the image to the owner's website using the File Transfer Protocol (FTP). To allow the owner to know when the IP camera sends an image to the website, this system also uses a PIR sensor as a motion detector. Every time it detects movement, the PIR sensor will give a logic "1" to the Arduino microcontroller that has been connected to the Ethernet shield, then the Ethernet shield will send information to the Android application in the form of a notification that there has been movement and A web link is provided so that the owner can download the image (Chaudry, 2020; Kyeremeh & Peprah, 2017; Yang et al., 2018).

The monitoring system can be divided into several blocks based on its function. The system block diagram is shown in Figure 6.

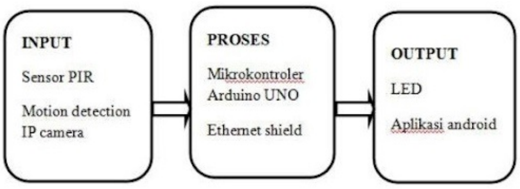

Figure 6. System Block Diagram

Based on Figure 6, this monitoring system uses a PIR sensor and Motion Detection feature on an IP camera as input. The input signal was processed by an Ethernet shield connected to an Arduino microcontroller, and the output signal was sent to the output block. The output block consists of an LED and notifications using an Android application.

3.2 Whole System Circuit

The input, process, and output blocks were integrated into the entire chain of this monitoring system, as shown in Figure 7.

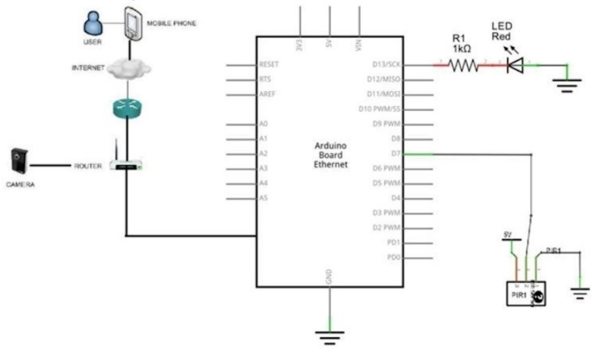

Figure 7. Entire System Circuit

Based on Figure 7, this figure illustrates the setup of an Arduino-based surveillance system integrated with a PIR sensor and Ethernet shield. The system allows users to monitor a connected camera remotely through the internet using their mobile phone. The Arduino board is connected to the internet via a router, and the PIR sensor is used to detect motion. When motion is detected, the sensor triggers the Arduino, which activates the connected red LED as a visual indicator. The circuit includes a 1kΩ resistor to limit current flow to the LED, ensuring proper operation. The mobile phone, through internet access, communicates with the system, allowing for real-time monitoring and interaction with the camera.

3.3 Software Design

The program begins with the initialization. Next, the program reads the PIR sensor data. If the PIR sensor detects movement, it means that the PIR sensor provides an input signal to the microcontroller for processing, and the microcontroller provides logic 1 (high) to the LED. The LED lights up, and the microcontroller sends information to the Android application via the router. The notification that will appear in the android application is the sentence 'Motion Detected, lets check http://surveillancesys.info/belindaadp/'. At the same time When the PIR sensor detects movement, the IP camera also detects movement, captures images, and sends them to the user's web. If the PIR sensor does not detect movement, the microcontroller will provide logic 0 (low) to the LED so that the LED does not light up, and the program will reread the condition of the PIR sensor.

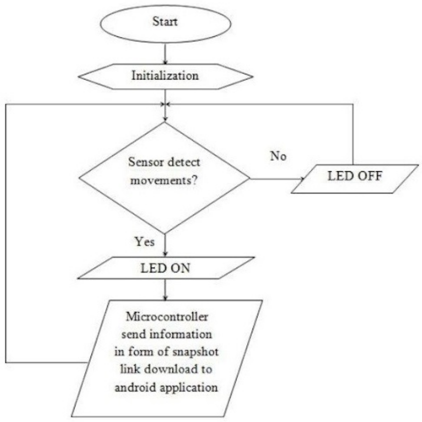

Figure 8. Program Flowchart

Based on Figure 8, the flowchart outlines the process of detecting motion and responding with an action. Initially, the system undergoes initialization. The PIR sensor then checks for any **detected movement**. If no movement is detected, the system turns the LED off and waits for further motion. If movement is detected, the system activates the LED to signal detection. Additionally, the microcontroller sends a snapshot link to an Android application, allowing the user to view the image related to the detected movement. This flow ensures the system responds appropriately to motion detection and communicates effectively with the user's mobile device.

4. Results and Discussion

4.1 Result

4.1.1 Input Block Test Results

The input block was tested by testing the PIR sensor and IP camera. The PIR sensor was tested by passing an object in front of the sensor. Testing the IP camera involved adjusting the camera settings so that it functioned as intended.

4.1.2 Test Results on PIR Sensor

Testing was performed by passing or moving an object in front of the PIR sensor. This test determines the maximum distance from which an object can be detected by the sensor. The object used in This test involves both living and non-living objects. The living object was a human, and the nonliving object was a chair. The tested circuit is shown in Figure 9.

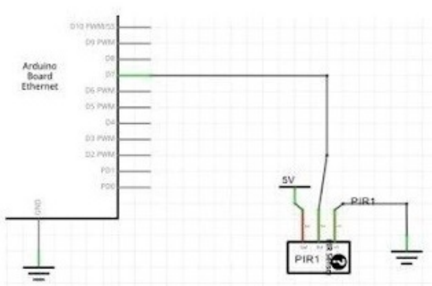

Figure 9. PIR Sensor Circuit

Based on Figure 9, this schematic diagram shows the basic wiring of the PIR motion sensor connected to the Arduino board. The PIR sensor is connected to the Arduino using the digital input pin D2. The sensor is powered by a 5V supply from the Arduino board, with the ground pin (GND) connected to the system's ground. When the PIR sensor detects motion, it sends a signal to the Arduino, which can then trigger further actions, such as activating the LED or sending data to an external system. The simplicity of this circuit ensures that motion detection can be easily integrated into various projects using the Arduino platform.

Table 2. PIR Sensor Test Results

Distance (cm)	Object	Indicator	Description	PIR Vout (V)
30	Human	LED ON	Detected	3.3
	Chair	LED ON	Detected	3.3
60	Human	LED ON	Detected	3.3
	Chair	LED ON	Detected	3.3
120	Human	LED ON	Detected	3.3
	Chair	LED ON	Detected	3.3
240	Human	LED ON	Detected	3.3
	Chair	LED ON	Detected	3.3
480	Human	LED ON	Detected	3.3
	Chair	LED ON	Detected	3.3
500	Human	LED OFF	Not Detected	0
	Chair	LED OFF	Not Detected	0

Based on the test results of the PIR sensor in Table 2 by passing an object in front of the sensor, it was able to detect object movement from a distance of 30 to 480 cm with an indication of the LED turning on. Meanwhile, at a distance of 500 cm and above, the PIR sensor cannot detect object movement; therefore, the LED does not light up. Based on the PIR HC-SR501 datasheet, the output voltage produced by the PIR sensor when the sensor detects an object is 3.3 V.

4.1.3 Test Results on IP Camera

The IP camera was tested by adjusting the IP camera settings. The settings were adjusted for motion detection so that the IP camera could detect movement and capture images.

Figure 10. IP Camera Web Login

As seen in Figure 10, to set the IP camera settings, the first step is to log in to the IP camera web by typing 192.168.0.101 in Internet Explorer then enters the username and password, which are printed on the back of the IP camera.

After logging in to the IP camera web, the homepage of the IP camera web appears, and there are several menus. The homepage of the IP camera is shown in Figure 11.

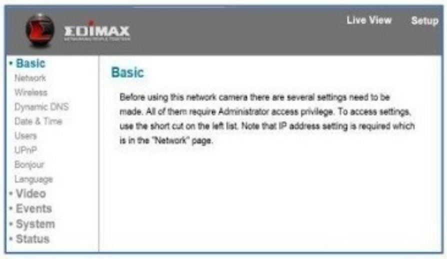

Figure 11. IP Camera Web Home

Based on Figure 11, the IP camera web homepage, there are several menus: Basic, Video, Events, System, and Status. Because this security system uses an IP camera as a motion detector, the Events menu was used. Within the Events menu, there is a submenu called Motion Detection. Motion Detection is a feature of IP cameras that captures images when movement occurs in front of the camera.

Figure 12. Settings in the Motion Detection Sub-Menu

Figure 12 shows the settings for the Motion Detection submenu. The first setting for the Motion Detection feature was to enable Motion Detection. The capture interval was set to 3 s. The next setting for sending snapshots can be selected to send via email or FTP, or just one or the other option. The next submenu in the Event menu to be configured is the File Transfer Protocol (FTP). This setting is intended for sending snapshots captured by an IP camera to the user's website.

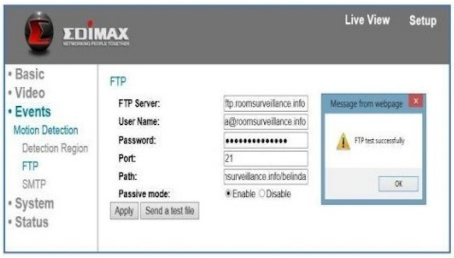

Figure 13. FTP Settings

The first step to enable FTP is to enter the FTP Server information. The FTP Server is the FTP address on the user's website so that snapshots can be sent correctly. Then, enter the username and password set in the Cpanel on the user's website. The port in the FTP settings was set to port 21. Next, enter the path or directory to save the sent snapshot. For passive Mode, the same port is set. The final step was to click Apply and Send a test file to verify that the settings were correct or not. Figure 13 shows that the FTP test was successful, meaning that the FTP server entered in the FTP settings was correct and that the user's website was ready to receive snapshots sent by the IP camera.

4.1.4 Process Bloock test Results

The process block testing was performed on the Arduino microcontroller. Testing is carried out by the voltage on the microcontroller port when the system is active can be calculated as follows:

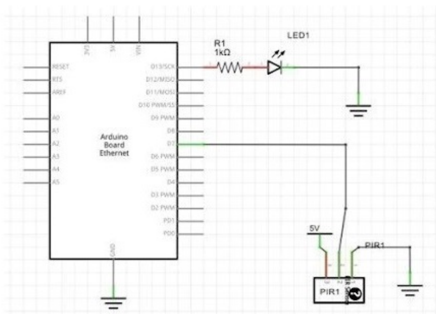

Figure 14. Process Block Series

As seen in Figure 14, the components tested in the process block are the PIR sensor and LED. The output voltages of both components were measured using a multimeter.

Table 3. Block Process Testing Results

No	Component	Port	Voltage (Volt)	Description
1	PIR Sensor	D7	3.3	Detects object motion
2	LED	D13	4.91	Lights up when the sensor detects object motion

Based on Table 3 of the process block test in Table 4.2, the output voltage produced by the PIR sensor when detecting motion is 3.3 V, and the output voltage produced by the LED when it was lit was 4.91 V.

4.1.5 Output Block Test Results

The output block test consisted of two parts: the LED and the Android application. The output block was tested by calculating the voltage across the LED when it was illuminated owing to the sensor detecting movement. In addition, testing was also carried out on the Android application by setting the application to receive notifications from the microcontroller when the PIR sensor detects movement, and this test shows the display of the Android application when there is a notification.

4.1.6 Test Results on LED

The LEDs were tested by measuring the output voltage (V) of the LED when it was lit. The circuit under test is illustrated in Figure 15.

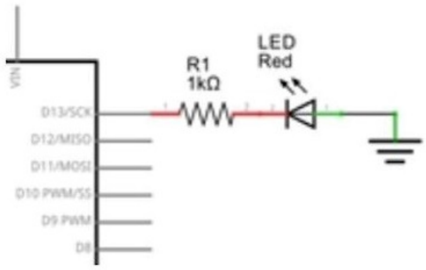

Figure 15. LED Circuit

Based on Figure 15, a LED was used as an indicator of the PIR sensor. When the PIR sensor detects movement, the LED lights up. To test the LED, the output voltage was measured using a multimeter.

The LED test results can be seen in Table 4.

Table 4. LED Testing Results

No	PIR Sensor	LED Condition	Voltage (Volt)
1	0	Off	0
2	1	On	4.91

Explanation:
0 = Does not detect object movement
1 = Detects object movement

Based on the LED test results on the Table 4, the output voltage produced by the LED when it is lit is 4.91 V.

4.1.7 Test Results on Android Applications

In the Android application, testing was performed by generating a key number that was entered into the Arduino program. This allows the application to connect to the microcontroller and display notifications when the PIR sensor detects movement. The test results for the Android application are shown in Figure 16.

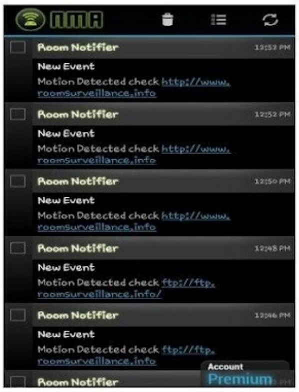

Figure 16. Android Application

Based on Figure 16, several notifications appeared that read 'Motion Detected check ftp:// ftp.roomsurveillance.info'. These notifications appear in the Android app every time the PIR sensor detects movement. Once the notification appears, users can click the link in the Android app to download the snapshot sent by IP camera.

4.1.8 Results of System Implementation

Testing on the input, process, and output blocks has been carried out, the next step is to implement all tests in the form of images starting from when the system is not yet active until the application receives a notification from the microcontroller and downloads an image of room conditions.

Figure 17. System Before Activation

The implementation of the system before it becomes active is shown in Figure 17. In the figure there are all the components used in this monitoring system, including IP cameras, PIR sensors, Arduino microcontrollers and Ethernet shields, LEDs resistors, adapters, switch hubs, and cable rolls.

To activate the system, the microcontroller adapter, switch hub, and IP camera were connected to the cable roll provided in the system. Once the system was active, the LEDs on the IP camera, adapter, and switch hub lit up (Adam et al., 2019; Zabala et al., 2017).

Figure 18. System After Activation

Figure 18 shows that the monitoring system was active, as indicated by the LEDs on the cable roll, switch hub, adapter, microcontroller, and IP camera. Once the system was active, the PIR sensor was tested by moving the participant in front of it. A notification then appears on the Android app. Simultaneously, the IP camera detects motion and sends snapshots to the website. The Android app screenshot is shown in Figure 20.

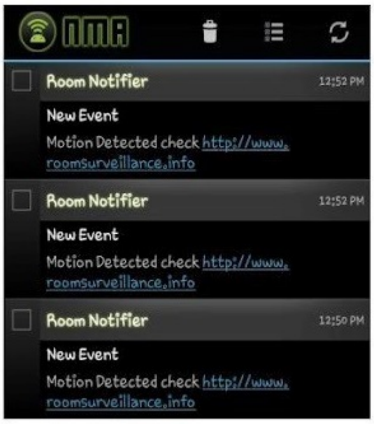

Figure 19. Android Application Display

On the Android app, the notification says, ' Motion Detected check ftp://ftp. roomsurveillance. info. An FTP link was used to view and download the images captured by the IP camera. To view snapshots sent by the IP camera, click the FTP link in the Android application. When the link is clicked, all images captured and sent by the IP camera appear.

Figure 20. User Website View

As shown in Figure 20, the snapshot links in the user's website directory are sorted by capture time. Therefore, the most recent image sent by the IP camera was at the bottom of the list. Clicking the image link displays the image captured by the IP camera and is ready for download.

Figure 21. Snapshot on User Website

The snapshot results captured and sent by the IP camera to the user's website are shown in Figure 21. Submitted snapshots in the user's website directory are sorted by time taken; therefore, to see the most recent snapshot, you will need to scroll to the bottom of the directory. Snapshots in the directory can be downloaded or viewed.

5. Conclusions

Based on the results of the device testing, the following conclusions were drawn: this system is designed to increase security in rooms containing valuables by using an IP camera as the main tool and a passive infrared (PIR) sensor as a motion detector. Users can receive information about room security in real time via the application android. The maximum distance at which the PIR sensor can detect objects is 480 cm. In this monitoring system, the IP camera sent the results of motion monitoring as snapshots. The results are then sent to the user's website. This monitoring system cannot use a cellular router because its data transfer speed is slow.

Acknowledgements

The authors would like to express their sincere gratitude to Dian Nusantara University for providing the necessary resources and support during this research. Special thanks to the contributors of the IP camera, PIR sensor, and Arduino components, whose contributions were invaluable in the successful completion of this project. The authors also appreciate the helpful feedback from the reviewers and the assistance from the technical team at the university.

Author Contributions

BA conceptualized the research, designed the system, and wrote the manuscript. SI contributed to the development and testing of the system, provided data analysis, revised the manuscript, assisted in the research methodology and provided valuable feedback during the manuscript writing process.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this study. This research was conducted independently, and no financial or personal relationships influenced the results or interpretation of the findings.

References

Abdulaali Jasim, Y., Othman Alsaaigh, M., Miqdad Flaih, T., & Ghanem Saeed, M. (2020). On announcement for university whiteboard using mobile application. *Computer Science Research and Its Development Journal*, *12*(1), 63–78.

Adam, G. K., Kontaxis, P. A., Doulos, L. T., Madias, E. N. D., Bouroussis, C. A., & Topalis, F. V. (2019). Embedded microcontroller with a ccd camera as a digital lighting control system. *Electronics*, *8*(1), 33. https://doi.org/10.3390/electronics8010033

Ashby, M. P. (2017). The value of cctv surveillance cameras as an investigative tool: An empirical analysis. *European Journal on Criminal Policy and Research*, *23*(3), 441–459. https://doi.org/10.1007/s10610-017-9341-6

Besteiro, R., Rodríguez, M. R., Fernández, M. D., Ortega, J. A., & Velo, R. (2018). Agreement between passive infrared detector measurements and human observations of animal activity. *Livestock Science*, *214*, 219–224. https://doi.org/10.1016/j.livsci.2018.06.008

Bhuyan, M. H., & Hasan, M. (2020). Design and simulation of heartbeat measurement system using arduino microcontroller in proteus. *International Journal of Biomedical and Biological Engineering*, *14*(10), 350–357.

Chaudry, A. M. (2020). Using arduino uno microcontroller to create interest in physics. *The Physics Teacher*, *58*(6), 418–421. https://doi.org/10.1119/10.0001841

Furter, J. S., & Hauser, P. C. (2019). Interactive control of purpose built analytical instruments with forth on microcontrollers – a tutorial. *Analytica Chimica Acta*, *1058*, 18–28. https://doi.org/10.1016/j.aca.2018.10.071

Hussain, M., Zaidan, A. A., Zidan, B. B., Iqbal, S., Ahmed, M. M., Albahri, O. S., & Albahri, A. S. (2018). Conceptual framework for the security of mobile health applications on android platform. *Telematics and Informatics*, *35*(5), 1335–1354. https://doi.org/10.1016/j.tele.2018.03.005

Jang, Y., Kim, D., Park, J., & Kim, D. (2018). Conditional effects of open-street closed-circuit television (cctv) on crime: A case from korea. *International Journal of Law, Crime and Justice*, *53*, 9–24. https://doi.org/10.1016/j.ijlcj.2018.02.001

Kyeremeh, F., & Peprah, F. (2017). Design and construction of an arduino microcontroller-based egg incubator. *International Journal of Computer Applications*, *168*(1), 15–23. https://doi.org/10.5120/ijca2017914261

Lahfaoui, B., Zouggar, S., Mohammed, B., & Elhafyani, M. L. (2017). Real time study of p&o mppt control for small wind pmsg turbine systems using arduino microcontroller. *Energy Procedia*, *111*, 1000–1009. https://doi.org/10.1016/j.egypro.2017.03.263

Martinez-Santos, J. C., Acevedo-Patino, O., & Contreras-Ortiz, S. H. (2017). Influence of arduino on the development of advanced microcontrollers courses. *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, *12*(4), 208–217. https://doi.org/10.1109/RITA.2017.2776444

Maw, W. W., Myint, C. C., & Lwin, S. S. (2019). College information system using android. *International Journal of Trend in Scientific Research and Development (IJTSRD)*, *3*(3), 806–809. https://doi.org/10.31142/ijtsrd22774

Mukhopadhyay, B., Srirangarajan, S., & Kar, S. (2018). Modeling the analog response of passive infrared sensor. *Sensors and Actuators A: Physical*, *279*, 65–74. https://doi.org/10.1016/j.sna.2018.05.002

Muthusenthil, B., & Kim, H. S. (2018). Cctv surveillance system, attacks and design goals. *International Journal of Electrical and Computer Engineering*, *8*(4), 2072. https://doi.org/10.11591/ijece.v8i4.pp2072-2082

Nayyar, A. (2016). An encyclopedia coverage of compiler's, programmer's & simulator's for 8051, pic, avr, arm, arduino embedded technologies. *International Journal of Reconfigurable and Embedded Systems*, *5*(1), 18–42. https://doi.org/10.11591/ijres.v5.i1.pp18-42

Patil, N., Algikar, B., Dange, S., Shejal, R., & Digge, R. (2017). Smart ticket checker system in railway. *JournalNX*, *3*(4), 42–44.

Piza, E. L., Welsh, B. C., Farrington, D. P., & Thomas, A. L. (2019). Cctv surveillance for crime prevention: A 40-year systematic review with meta-analysis. *Criminology & Public Policy*, *18*(1), 135–159. https://doi.org/10.1111/1745-9133.12419

Ray, P. P. (2017). A survey on visual programming languages in internet of things. *Scientific Programming*, *2017*(1), 1231430. https://doi.org/10.1155/2017/1231430

Riadi, I., Fadlil, A., & Fauzan, A. (2018). Evidence gathering and identification of line messenger on android device. *International Journal of Computer Science and Information Security (IJCSIS)*, *16*(5), 201–205.

Tao, H., Bhuiyan, M. Z. A., Rahman, M. A., Wang, G., Wang, T., Ahmed, M. M., & Li, J. (2019). Economic perspective analysis of protecting big data security and privacy. *Future Generation Computer Systems*, *98*, 660–671. https://doi.org/10.1016/j.future.2019.03.042

Thakare, M. C. H., Tote, M. M. A., & Pandey, M. A. P. (2016). To design database application for android operating system. *Global Journal of Engineering Science and Research Management*, *3*(4), 7–13.

Tt, O., & Oo, A. (2018). Design and implementation of arduino microcontroller based automatic lighting control with i2c lcd display. *Journal of Electrical & Electronic Systems*, *7*(2). https://doi.org/10.4172/2332-0796.1000258

Welbourne, D. J., Claridge, A. W., Paull, D. J., & Lambert, A. (2016). How do passive infrared triggered camera traps operate and why does it matter? breaking down common misconceptions. *Remote Sensing in Ecology and Conservation*, *2*(2), 77–83. https://doi.org/10.1002/rse2.20

Welsh, B. C., Piza, E. L., Thomas, A. L., & Farrington, D. P. (2020). Private security and closed-circuit television (cctv) surveillance: A systematic review of function and performance. *Journal of Contemporary Criminal Justice*, *36*(1), 56–69. https://doi.org/10.1177/1043986219890192

Wu, L., & Wang, Y. (2018). A low-power electric-mechanical driving approach for true occupancy detection using a shuttered passive infrared sensor. *IEEE Sensors Journal*, *19*(1), 47–57. https://doi.org/10.1109/JSEN.2018.2875659

Wu, L., Wang, Y., & Liu, H. (2018). Occupancy detection and localization by monitoring nonlinear energy flow of a shuttered passive infrared sensor. *IEEE Sensors Journal*, *18*(21), 8656–8666. https://doi.org/10.1109/JSEN.2018.2869555

Wu, Y. L., Tao, Y. H., & Chang, C. J. (2017). A comparative review on privacy concerns and safety demands of closed-circuit television among taiwan, japan, and the united kingdom. *Journal of Information and Optimization Sciences*, *38*(1), 173–196. https://doi.org/10.1080/02522667.2016.1220076

Yang, D., Xu, B., Rao, K., & Sheng, W. (2018). Passive infrared (pir)-based indoor position tracking for smart homes using accessibility maps and a-star algorithm. *Sensors*, *18*(2), 332. https://doi.org/10.3390/s18020332

Zabala, P., Abas, M. C., & Cerna, P. (2017). Development of programmable relay switch using microcontroller. *American Journal of Remote Sensing*, *5*(5), 529–551. https://doi.org/10.11648/j.ajrs.20170505.11