Unit 19 - The Internet of Things
Design and Build an IoT Prototype

Contents

Introduction

Following my investigation into IoT technology, I have been tasked by the manager to design and develop a prototype system to address issues with the company's current alarm systems. The proposed solution will involve using an Arduino Uno R3 to detect movement within a room via a PIR sensor, another compatible sensor, or a camera. Once movement is detected, the system will automatically send an email alert to a pre-configured email address and generate a log file that can be reviewed later.

Project Plan

To ensure the IoT prototype project is completed in a structured and timely manner, a detailed project plan that I have developed using Monday.com. The plan outlines each major task required by the brief, along with their subtasks, descriptions, and progress tracking. This structured approach helps manage time effectively, monitor progress, and make improvements based on the feedback at each stage of development.

Live Project Plan Access:
To view the most up-to-date version of this project plan, including real-time progress updates, here is the following link: https://view.monday.com/1905404805-121f9be3390f1cb23ed9138dcd9425ec?r=euc1

Task 1: Prototype System Design

What do I need to do?:
I have to plan out a smart alarm system and think about all the different bits it will have (sensor, Arduino, how it connects), how they'll talk to each other, any rules or standards it might follow, and how I'll keep it secure.

Research & Planning:

For the research and planning stage of my smart alarm system, I explored existing IoT-based alarm technologies. This helped me identify the essential components needed to build a functional prototype using Arduino-compatible hardware.

For this project, I will be using the **ELEGOO UNO R3 kit**, which includes an Arduino Uno-compatible board and a variety of sensors and components, and the system goal is to detect motion using a PIR motion sensor or compatible camera, if available. When motion is detected, the Arduino will send a signal over the USB to the computer that is running a script which would send an email to alert and log the event for future review.

Technologies Considered:

The technologies and components that I will be using for building:
The Arduino UNO R3 (Elegoo Uno R3) will be the main controller which would be detecting motion and sending a simple serial signal to the computer.

The PIR Motion will detect the infrared motion in the environment and send a high signal to the Arduino.

The computer running a program that connects to the Arduino over the USB listens for the Serial messages from the Arduino and when it detects a motion signal it would trigger an action that would send an email and log the event.

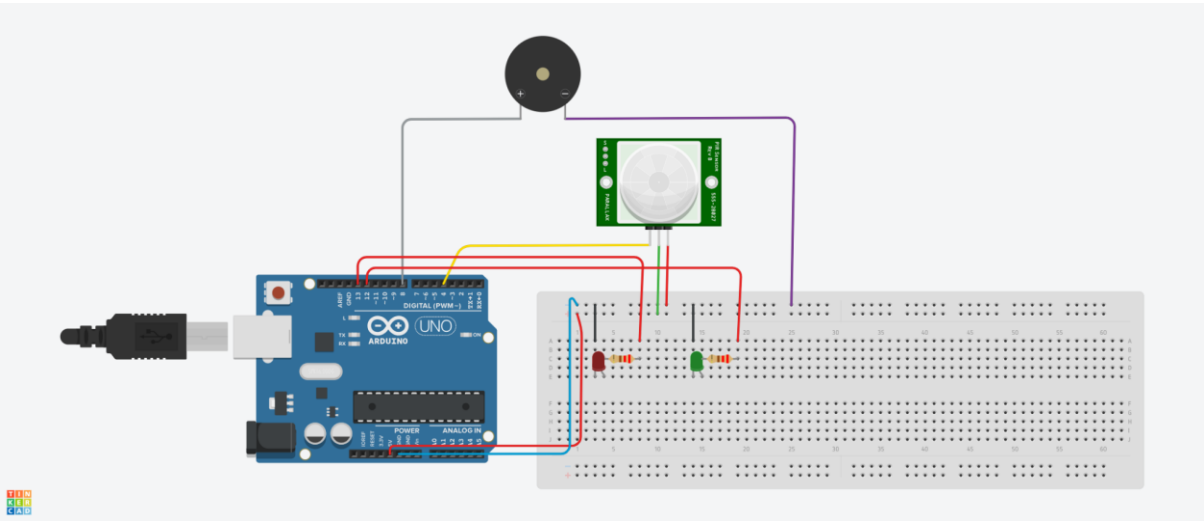
The email sending with C++ that would be using SMTP client libraries like libcurl or C++ Rest SDK to send the email when motion is detected.

The local Event Logging that the computer would store a timestamp log file for all detected motion events for audit and reviews.

The buzzer and the LEDs would be connected to the Arduino to provide a local sound and the light alert on motion detection. The red LED will light up when motion is detected and the green LED will stay on when no motion is detected, and this would give a clear and immediate visual indication of the system's status at all times.

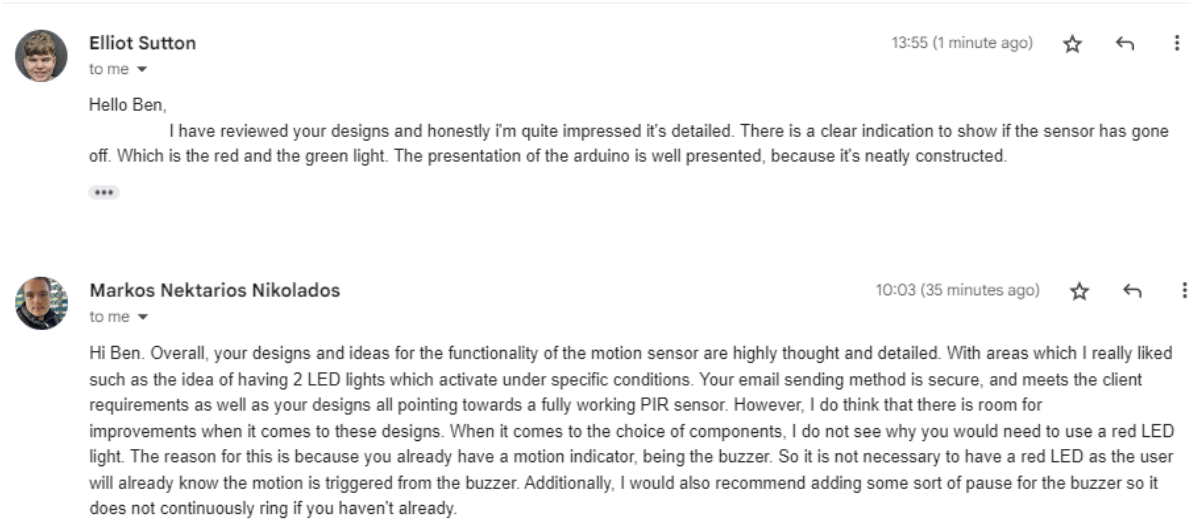
Why this Approach:
The reason why I have gone for this approach is because it makes the system clear and easy to understand. When motion is detected, the red LED light and the buzzer give an instant alert, which makes it obvious that something is happening. At the same time, the system will send an email to notify the user that motion has been detected. When there is no motion, the green LED light will stay on to show that the area is secure and the system is monitoring. This visual, audio, and email feedback makes it simple for the user to know what's going on without needing to look at the computer or the logs.

System Architecture Design:



Task 2: Design Review and Optimisation

To improve my motion sensor project, I showed my plan to two people and asked for their feedback.



Elliot Sutton 13:55 (1 minute ago)
to me

Hello Ben,

I have reviewed your designs and honestly i'm quite impressed it's detailed. There is a clear indication to show if the sensor has gone off. Which is the red and the green light. The presentation of the arduino is well presented, because it's neatly constructed.

Markos Nektarios Nikolados 10:03 (35 minutes ago)
to me

Hi Ben. Overall, your designs and ideas for the functionality of the motion sensor are highly thought and detailed. With areas which I really liked such as the idea of having 2 LED lights which activate under specific conditions. Your email sending method is secure, and meets the client requirements as well as your designs all pointing towards a fully working PIR sensor. However, I do think that there is room for improvements when it comes to these designs. When it comes to the choice of components, I do not see why you would need to use a red LED light. The reason for this is because you already have a motion indicator, being the buzzer. So it is not necessary to have a red LED as the user will already know the motion is triggered from the buzzer. Additionally, I would also recommend adding some sort of pause for the buzzer so it does not continuously ring if you haven't already.

What did Person 1 said:

He liked my ideas especially how the LEDs work based on different conditions and he thought my email system was secure and that the PIR sensor part was done well but he also suggested I didn't need the red LED because the buzzer already lets the user know when motion is detected and he also recommended adding a short delay to the buzzer so it doesn't go off continuously which it could get annoying.

What did Person 2 said:

He has said that he was impressed with how detailed my design was and he liked how the red and green LEDs made it clear if the sensor was on or off and he also said that the way I presented the Arduino was really neat and easy to understand.

What I changed:

I have decided to add a delay to the buzzer and like person 1 has said it only goes off for a short time and doesn't keep making noise.
The other one is that I kept both LEDs because person 2 liked how they clearly showed the sensor status and even though Person 1 thought the red one wasn't needed and I felt it still helped with making the system clearer.

Why I made these changes:

The delay makes the buzzer less annoying and more useful and keeping both LEDs means people can see what is going on even without sound and I tried to balance both pieces of feedback to make my project better and easier to use.

Task 3: Justification of Design Decisions

When I designed my alarm system I thought carefully about how to make it simple, clear and effective and I made the decisions based on how useful each part would be and how the system would work in real life.

PIR Sensor:
I chose to use a PIR sensor because it is a reliable way to detect motion and it's perfect for an alarm system because it can sense when someone moves in front of it and it doesn't need to touch anything, and it works automatically.

Buzzer:
I have added a buzzer to act as a loud warning when motion is detected, and I wanted the user to be immediately alerted if someone enters the area and after getting the feedback I added a pause/delay to the buzzer so it wouldn't keep going off nonstop and making it more user friendly.

Led Light:
For the led light I used two LED lights which are one red and one green to show the system status and the green led shows when everything is fine and the sensor is not triggered and the red led turns on when motion is detected and even though someone suggested removing the red LED and I decided to keep it because it gives a clear visual sign of danger and especially if the buzzer can't be heard.

Email Notification:
For the email notification I have included an email feature to alert the user even if they are not nearby and it is a helpful way for someone to know something's wrong when they are not at home and this would make the alarm more useful and secure.

Neat Wiring and Clear Layout:
For neat wiring and clear layout, I have kept the wiring and layout neat so it's easy to understand and fix if something goes wrong and it also helps if someone else wants to build or upgrade the system.

Task 4: Prototype Development and Testing

For the task I built a working version of my smart alarm to test if it works the way I planned.

Hardware Setup:

The hardware I used the following components for my prototype:
Arduino Uno R3 is the main controller that runs the code.
PIR motion sensor is to detect the movement.
Buzzer is to make a sound when motion is detected.
The red and green LEDs that show whether motion is detected or not.

How it works:

When no movement is detected the green LED light would stay on and if the PIR sensor detects motion the following thing would go off
The green LED light would turn off.
The red LED light and the buzzer would turn on.
The system would send an email alert to notify the user, and a pause/delay is added so that the buzzer doesn't keep going off.

Code:

For the code, I wrote it using the Arduino IDE and the code checks the PIR motion sensor and controls the LEDs and buzzer based on whether motion is detected or not.

The green led stays on when there is no movement detected and when the PIR sensor detects the movement the red led, and the buzzer turns on and it also uses the serial monitor to see messages like "Motion Detected" or "Motion ended!" which would help me test and debug the system while it was connected to the computer.

The Arduino is connected to the computer using a USB cable, which also powers the system, and the serial monitor in the Arduino IDE shows real-time messages that let me confirm the system is working correctly.

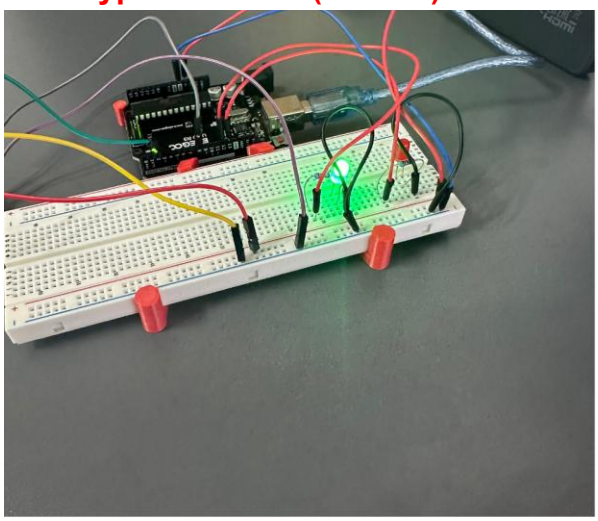
For the future, I could connect the Arduino to an email system using a computer-based program or external module. This would send an alert when motion is detected. Right now, I focused on testing the LED and buzzer response.

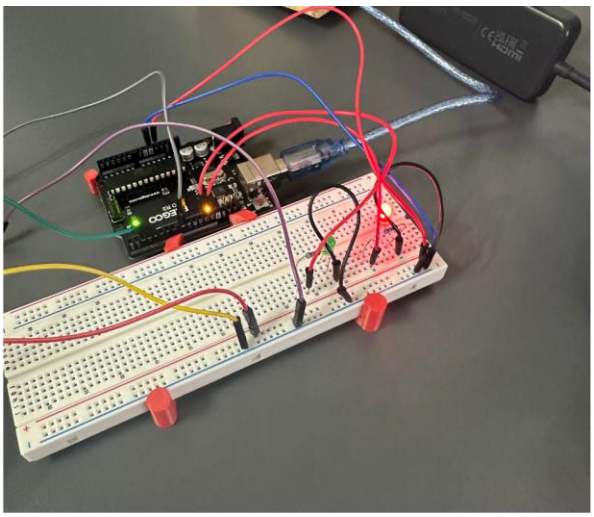
Prototype Testing Table:

Test No	What I Tested	Expected Result	Actual Result	Pass or Fail
1	No Motion in front of sensor	The green led stays on and the red led, and buzzer are on serial message.	The green led stays on and the red led and buzzed stays off and no serial message is shown	Pass
2	Motion detected in front of sensor	The red led and buzzer turns on and the green led turns off serial shows “Motion detected!”	The exactly as expected the red led and buzzer activated, and the green led off and the message shows	Pass
3	Stop moving after triggering the sensor	The buzzer and red led turns off and the green led turns back on with the serial show “Motion ended!”	For all the components reset correctly when Motion is ended and appears in the serial monitor	Pass
4	Serial monitor output when triggered	It should display the correct timestamp and message when the motion is detected or stops	The time stamped message appeared clearly: “Motion Detected!” or “Motion Ended!”	Pass
5	Power through USB and device connection	The Arduino powers on and runs the code properly when connected to the computer	The powered correctly and the code ran automatically the LEDs and buzzer worked as expected	Pass
6	Walk in and out of sensor rang repeatedly	The LEDs and the buzzer should switch on and off each time motion is detected or ends	The system responded well every time when it switched correctly between green and red LEDs	Pass
7	Delay between detection and buzzer deactivation	The buzzer turns off shortly after motion has stopped	The delay worked fine, and the buzzer stopped within the expected time.	Pass
8	PIR sensor range	The sensor	The motion	Pass

	and angle	should detect the motion from front but not from the side or far distance	detected reliably from the front which ignored motion from the outside range.	

Prototype Evidence (Photos)





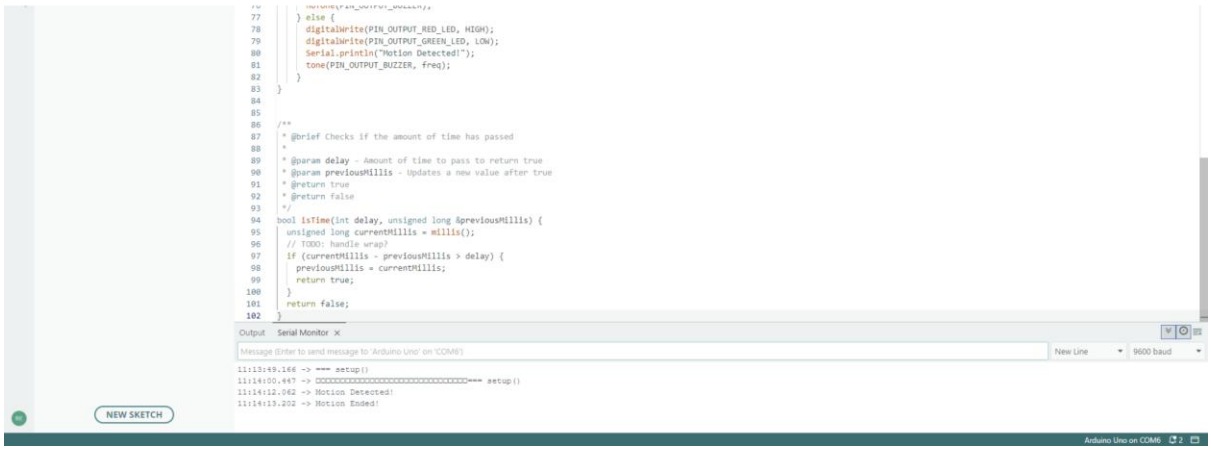


Image 1: System in idle mode
The green LED is lit, indicating that no motion is currently detected. The red LED and buzzer are off. This demonstrates that the system is in an active monitoring state and ready to detect motion.

Image 2: Motion Detected - Alert Active
When motion is detected by the PIR sensor, the red LED turns on and the green LED switches off. The buzzer is triggered, and the system sends a signal via the serial connection to the PC to initiate an email alert and log the event. This confirms the system's response to motion is functioning as intended.

Image 3: Code output in serial monitor
The serial monitor displays real-time feedback from the Arduino, showing messages such as “Motion Detected!” and “Motion Ended!”. This confirms the system correctly logs detection events and reflects the programmed behaviour.

Task 5: Prototype Optimisation Based on Feedback

Feedback I Received:

I showed my working alarm to two classmates and my teacher and one person said that the buzzer was too loud, and the teacher suggested adding a delay or timer to reduce how long it stays on and the another person said it would be helpful if the system could display how many times it had been triggered, to make it easier to keep track of activity.

Changes I Made:

Based on this feedback, I made two main improvements:
I added a delay to make the buzzer only sound for 1 second when motion is detected, so it's less disruptive.

I added a counter in the code that increases every time motion is detected and shows the count in the serial monitor.

Testing the Improved Version:

Test No.	What I changed	Expected Result	Actual Result	Pass or Fail
1	Reduced buzzer time to 1 second	The buzzer sounds for 1 second then stops.	The buzzer turned off after 1 second as expected	Pass
2	Added motion detection counter	The serial monitor shows "Motion detected!".	The counter would increase and show each detection clearly	Pass
3	Repeated motion in sensor range	The LEDs and the buzzer work correctly and the counter keeps going	The LEDs worked fine and the counter increased each time.	Pass

Evidence of Improvements:

With in my video you can see that the alarm system works properly and when the sensor detects motion the red LED turns on and the buzzer makes a sound which I changed the buzzer so that it only stays on for 1 second because someone has said it was too loud and stayed on for too long and now it's turns off quickly which is better and you can also see in the serial monitor that it shows how many times motion was detected and I added that so i can keep track of how often it's goes off and these changes were based on the feedback i got and the video shows that they are now working the way I wanted.

Task 6: Evaluation of the Optimised IoT System

For my final version of the smart alarm system meet what the client asked for It uses a PIR sensor connected to an Arduino Uno R3 to detect movement and when motion is detected it turns on the red led and buzzer and sends a signal to the computer which can then send an email alert and log the event. The green LED stays on when everything is normal and I tested the system several times and it always reacted as expected and this shows the system works well and solves the original problem.

Using Feedback to Improve It:

After showing the design to other I made some changes based on their advice and one person said that the buzzer was too loud so I added a delay to make it turn off after 1 second and the other suggested I keep track of how often the sensor was triggered so I added a counter in the serial monitor and these changes made the system easier to use and more useful.

Risk and Benefits:

Benefits	Risks
It uses low cost and easy to find components	Might go off by mistake from pets or other movement
Gives clear alerts with lights and sound	Needs a computer to be on for the email to work
Can be expanded in future e.g., with an camera or an app	No password or encryption on the data
Sends email alerts and keeps a log for later checks	Doesn't work if the power or the USB connection is lost.

Security, Legal and environmental Issues:

Security:

The system sends data over a USB serial connection which isn't protected and if someone else used the computer it could see or change the messages and in future I could use a secure Wi-Fi board (like ESP32) and send data safely and I'd also use secure email login.

Legal:

For even though the system doesn't take photos it still logs the motion data and in some cases like in a shared space or business that could be personal data and if it is used in real life it would need to follow the data privacy rules like GDPR.

Sustainability:

The system is powered through an USB and uses very little electricity and all the parts can be reused in other projects and no batteries are wasted which makes it more environmentally friendly.

My Time Management and Responsibility:
I used a planning tool called Monday.com to keep track of what I needed to do and when and I split the project into smaller steps like designing, testing and making improvements I worked on fixing problems by myself like adjusting the buzzer code and adding the counter and I also made sure that my wires and code were neat and easy to understand and this helped me to stay organised and finish the project properly.

References

Arduino, 2024. Arduino IDE - Integrated Development Environment. [online] Available at: https://www.arduino.cc/en/software [Accessed 20 May 2025].

ELEGOO, 2024. UNO Project Super Starter Kit. [online] Available at: https://www.amazon.co.uk/ELEGOO-Project-Tutorial-Controller-Projects/dp/B01D8KOZF4 [Accessed 20 May 2025].

Autodesk, 2024. U19A2v1 Motion Sensor Simulation – Tinkercad Circuit. [online] Available at: https://www.tinkercad.com/things/7DtlKlEJNEx-u19a2v1 [Accessed 20 May 2025].

Instructables, 2024. PIR Motion Sensor With Arduino in Tinkercad. [online] Available at: https://www.instructables.com/PIR-Motion-Sensor-With-Arduino-in-Tinkercad/ [Accessed 20 May 2025].

monday.com, 2024. Work OS by monday.com. [online] Available at: https://www.monday.com [Accessed 20 May 2025].

GDPR.eu, 2024. General Data Protection Regulation (GDPR) Compliance Guidelines. [online] Available at: https://gdpr.eu [Accessed 20 May 2025].

UK Government, 2018. Data Protection Act 2018. [online] Available at: https://www.legislation.gov.uk/ukpga/2018/12/contents/enacted [Accessed 20 May 2025].