Term Project Documentation

GROUP-5

WEB CAM SECURITY USING BEAGLE BONE BLACK

TEAM MEMBERS

Mahaveerasaiteja Veluduti - G01354854

Vivekananda Reddy Sappidi - G01380849

INDEX

Abstract

Security and surveillance systems play a pivotal role in safeguarding homes and businesses against unauthorized access and potential threats. In this paper, we present a novel security door camera system implemented on the Beagle Bone Black platform, a low-cost, open-source embedded computer. Our system leverages the power and versatility of the Beagle Bone Black to create a robust and cost-effective solution for monitoring and securing entry points. We explore the hardware and software components of the system, including camera integration, real- time image processing, and remote access capabilities. Additionally, we discuss the system's performance and security features, demonstrating its potential as a reliable tool for enhancing security and providing peace of mind to users.

Introduction

Security issues have expanded tremendously in recent years, necessitating the development of novel solutions that combine price, functionality, and ease of use. Doorbell cameras, often known as video doorbells or security door cameras, are becoming increasingly popular as an essential component of modern security systems. These gadgets offer real-time surveillance and remote access, letting homeowners and businesses to effortlessly monitor their entry points.\cite{b1}

The Beagle Bone Black, a single-board computer created by the BeagleBoard.org community, has proven to be a viable platform for a wide range of embedded applications. Because of its small size, low cost, and extensive networking choices, it is an excellent contender for designing security systems. We offer a full overview of our security door camera system built on the Beagle Bone Black platform in this article. Organizational editing before formatting. We use a simple switch i.e., doorbell, we will connect a vibration detector whenever a person knocks on the door. So, if the bell is pushed the camera opens and takes a picture and sends an email notification to the owner. It uses a Webcam as the Security camera. This Webcam is directly connected to the Beagle Bone Black, we power the Webcam through USB. The project is used for the purpose when the owner of the house is out, and any person comes and knocks, they can view through there and phone and even if they are at home they can check before opening the door. The selection of the project involving a web cam security doorbell using Beagle Bone Black was driven by several:

1)Relativity: Security and Surveillance systems have become increasingly vital in today's world due to rising concerns about safety and property. Beagle Bone Black is an open-source Linux – based platform. This project gain hands on experience in embedded systems, real-time image processing, networking, and security protocols, contributing to our technical skill development.

In the subsequent sections of this paper, we provide a detailed account of our system's architecture, hardware components, software design, and performance evaluation, show- casing its effectiveness as a security tool in today's dynamic and evolving security landscape.

1. Components & OS Requirements:

OS Requirements:

OS: **debian@BeagleBone:~$ uname -a**

Linux BeagleBone 5.10.168-ti-rt72 #1bullseye SMP PREEMPT Sat Sep 30 03:37:21 UTC 2023 armv7l GNU/Linux

Components Required:

Beagle Bone Black, Bread Board, Wires, Resistor 10K, Switch, Web Camera(Logitech). The cam can be powered through a USB. Connect the webcam to the Beagle Bone Black board using a USB port. Connect the switch to the Beagle Bone Black board using GPIO pins.

2. Implementation

Connect BeagleBoneBlack to Internet on Windows OS Via USB Port

We need share the internet network of our computer with beagle bone black . once the network is shared we need to set the default gateway by using below command.

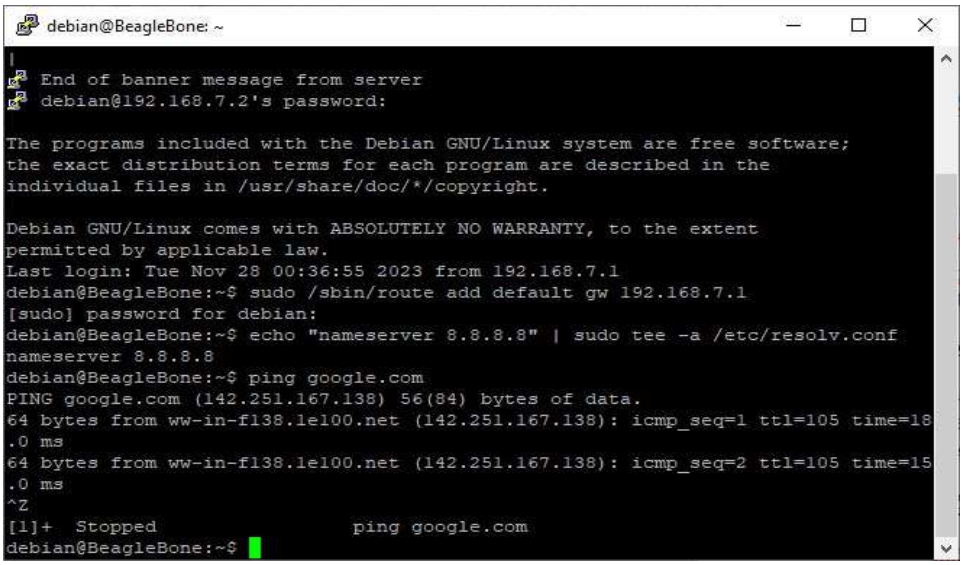
Below link helps to setup up the internet on the beagle bone black.

https://www.digikey.com/en/maker/blogs/how-to-connect-a-beaglebone-black-to-the-internet-using-usb

- **debian@BeagleBone:~$ ping 8.8.8.8**

ping: connect: Network is unreachable

- **debian@BeagleBone:~$ sudo /sbin/route add default gw 192.168.7.1**
- **debian@BeagleBone:~$ echo “nameserver 8.8.8.8” | sudo tee -a /etc/resolv.conf**



Now test the network is connected to beagle bone or not we need to ping 8.8.8.8 because we set DNS as 8.8.8.8 and we can test by ping google.com

Then we need install the dependencies for our program
before that we need to update the beagle bone
Below are the coomands

$sudo apt-get update

$sudo apt-get upgrade

A. Library required:

fswebcam: This is the software used to capture images from the webcam. Install it with:
$ sudo apt-get install fswebcam

To use the mpack utility for attaching the image to an email, you'll need to install the mpack package on your BeagleBone Black. Additionally we need the ssmtp package for sending emails. Here are the commands to install these packages:

$ sudo apt-get install mpack

$ sudo apt-get install ssmtp

B. Configuring the ssmtp file:

Configuring ssmtp involves setting up the /etc/ssmtp/ssmtp.conf file with the appropriate parameters for your email provider.

The configuration file for ssmtp is typically located at /etc/ssmtp/ssmtp.conf. Open the configuration file with a text editor:

$ sudo nano /etc/ssmtp/ssmtp.conf

We need add this content in that file

root=your_email@example.com
mailhub=smtp.your-email-provider.com:587
hostname=your-hostname
AuthUser=your_email@example.com
AuthPass=your_email_password
UseSTARTTLS=YES
FromLineOverride=YES

- root: The default sender email address.
- mailhub: The address and port of your SMTP server. Replace your-email-provider.com and 587 with the appropriate values.
- hostname: The hostname of your machine.
- AuthUser: Your email address for authentication.
- AuthPass: Your email app password . we need to create email app password according to your email provider
- UseSTARTTLS: Enable TLS encryption.
- FromLineOverride: Use the From address specified in the email, not the system default.

My ssmtp.conf file look like this

```
#
# Config file for sSMTP sendmail
#
# The person who gets all mail for userids < 1000
# Make this empty to disable rewriting.
root=mahaveerasaiteja@outlook.com
# The place where the mail goes. The actual machine name is required no
# MX records are consulted. Commonly mailhosts are named mail.domain.com
mailhub=smtp.office365.com:587
# Where will the mail seem to come from?
rewriteDomain=outlook.com
# The full hostname
hostname=BeagleBone.localdomain
# Are users allowed to set their own From: address?
# YES - Allow the user to specify their own From: address
# NO - Use the system generated From: address
FromLineOverride=YES
# Use STARTTLS for secure connection
UseSTARTTLS=YES
UseTLS=YES
# Gmail authentication details
AuthUser=mahaveerasaiteja@outlook.com
AuthPass=vojxfypzurapwcqx
```

$ sudo chmod 640 /etc/ssmtp/ssmtp.conf

The above command sets the file permissions to be readable and writable only by the owner and readable by the group.
Now, we should be able to use ssmtp to send emails using the configuration you've set up.
Now we can run the our code using gcc compile

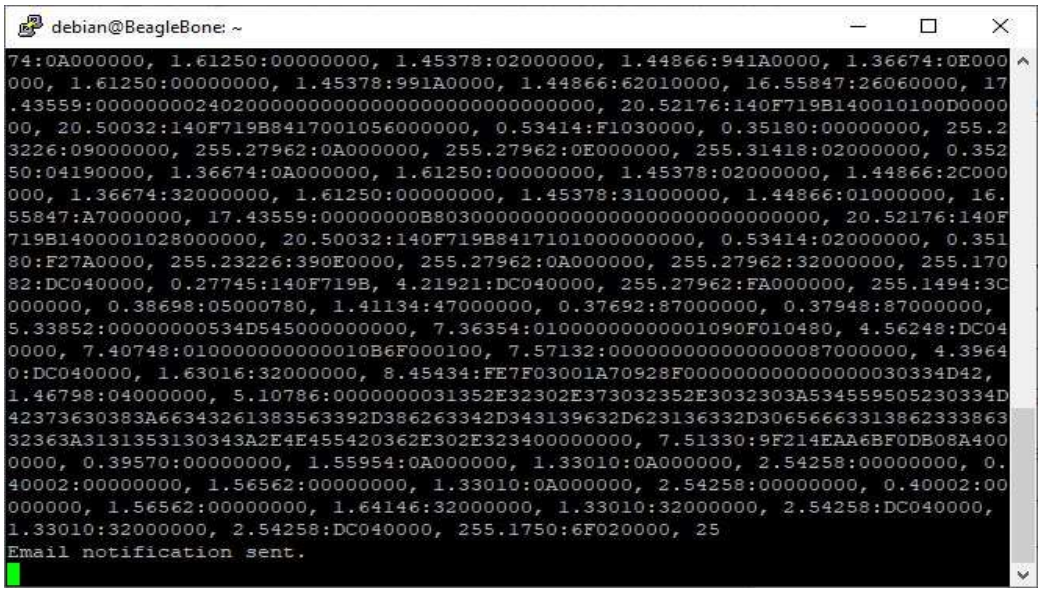
3. Results

In our code we used three thread
1.button thread which used to pressed the button
2. Capture thread which is used to capture the image from the webcam
3. Email thread this is used to send the email notification.
Now we should run our code by using the following commands

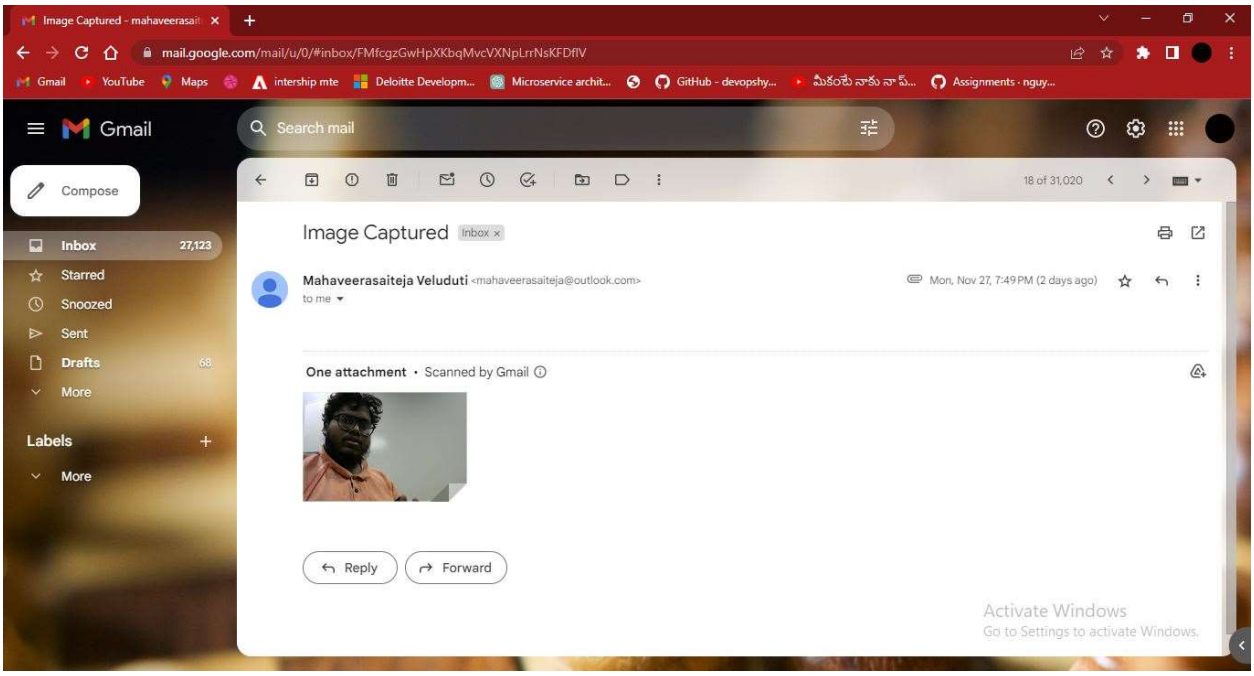
$ gcc -pthread project.c -o test1
$ sudo ./test1

The Above commands will run our program.

The above screenshot show that the command is executed and the when we press the button we can button pressed and capturing image displayed



The above screenshot shows that the email notification sent to email which was mentioned in program



The above screenshot shows the email notification captured by the webcam using the beagle bone black received to the mail mentioned the program.

4. TEST DATA

To test your program we have divided our code into two part
First , task is to caputer the image and save it as image.jpg ,Secondly the saved image.jpg will be sent to email as notification using mpack and ssmtp library .

Task 1. Task is to caputer the image and save it as image.jpg

To do the task 1 we used fswebcam library : This is the software used to capture images from the webcam. And after capturing the image we gave specific path to save the image that is **“/home/debian/image.jpg"**

Task 2 : The saved image.jpg will be sent to email as notification.

To do the above task we need to set the /etc/ssmtp/ssmtp.conf
using $ sudo nano /etc/ssmtp/ssmtp.conf we can configure the file according to our email provider .

To secure the configuration file, restrict its permissions:

Using the following commands we can set
$sudo chmod 640 /etc/ssmtp/ssmtp.conf
$ sudo chown root:mail /etc/ssmtp/ssmtp.conf

Now we can test the configuration by sending a test email by the following command
$ echo "Subject: Test Email" | sudo ssmtp your_email@example.com

This should send an email with a subject line to your specified email address.

5. Conclusion:

Creating a security using a BeagleBone Black and a webcam involves integrating hardware and software components to monitor and respond to events at a door. Below is a sample conclusion for a project like this:

In conclusion, the development of a Web Cam Security using the BeagleBone Black has resulted in a robust and efficient solution for enhancing the security of a door or entrance. The project successfully integrates a webcam with the BeagleBone Black, leveraging its GPIO capabilities and the flexibility of the Linux environment.

The key achievements of this project include:

[1] Hardware Integration
Successful integration of a webcam with the BeagleBone Black, enabling real-time video capture and processing.

[2] GPIO and Button Integration;
Implementation of GPIO-based button control for triggering the doorbell system. This provides a simple and effective user interface.

[3] Image Capture and Processing:
Utilization of the `fswebcam` tool to capture images when the doorbell is pressed. The captured images are stored locally for further analysis or reference.

[4] Email Notification:
Implementation of an email notification system using the `ssmtp` tool, allowing users to receive alerts with attached images when the doorbell is pressed.

[5] Security and Automation:
Integration of security features, such as capturing images upon doorbell press, enhances the overall security of the monitored area.

The email notification system adds an automation layer, ensuring that users receive timely alerts, even when not physically present at the monitored location.

[6] Flexibility and Customization:
The use of a Linux environment on the BeagleBone Black provides flexibility and room for customization. Users can extend the functionality of the doorbell system by incorporating additional features or integrating it with other smart home devices.

[7] Potential for Expansion:
The project lays the groundwork for future expansions, such as video streaming, cloud integration, or integration with other home automation platforms.

While the current implementation fulfills the basic requirements of a security doorbell, there is ample room for future enhancements and customization based on specific user needs and evolving technologies.

In summary, the Web Cam Security project demonstrates the versatility of the BeagleBone Black in creating efficient and customized solutions for home security applications. The combination of hardware interfacing, software development, and automation opens the door to a wide range of possibilities for users seeking to enhance the security and convenience of their homes.