piMask

Capture in my Pi (PiCapture)

Functional Specification

Version 0.3

Sylvestor George
sylvestor.george88@gmail.com

Version History

Version	Changes
Version 0.1	Functionality Specification for PiCapture feature
Version 0.2	Storing videos in Cloud server instead of locally on Pi storage
Version 0.3	Using Wi-Fi enabled camera instead of USB camera

Introduction

This document entails the functionality of using a low-cost HD camera for surveillance purposes that records a video for a certain amount of set time. The video is recorded based on the motion sensor that is enabled on the camera. The recorded video is fed to a Raspberry Pi device from where it is further transferred to a cloud server where it is stored.

References

1. Adafruit Learning System. (n.d.). Retrieved July 31, 2015, from https://learn.adafruit.com/category/learn-raspberry-pi

2. Turn your Pi into a low-cost HD surveillance cam - Raspberry Pi. (2013, October 14). Retrieved July 31, 2015, from https://www.raspberrypi.org/blog/turn-your-pi-into-a-low-cost-hd-surveillance-cam

3. Buenger, C. (n.d.). Raspberry Pi as low-cost HD surveillance camera. Retrieved July 31, 2015, from http://www.codeproject.com/Articles/665518/Raspberry-Pi-as-low-cost-HD-surveillance-camera

4. How to make a DIY home alarm system with a raspberry pi and a webcam. (2013, September 15). Retrieved July 31, 2015, from https://medium.com/@Cvrsor/how-to-make-a-diy-home-alarm-system-with-a-raspberry-pi-and-a-webcam-2d5a2d61da3d

Requirements

The basic requirements for implementing this functionality are:

- HD Camera (wireless)
- Raspberry Pi Model B
- Power Supply for Pi
- Cloud Server
- Internet

Functional Overview

The implementation of this functionality can be described in multiple steps:

Step 1: Deploying the HD camera for surveillance that captures video based on any motion detection. A wireless camera is used which is connected to the home network.

Step 2: Setting up the Raspberry Pi as a server, which receives the captured videos from the surveillance camera and transfers it to the remote server.

Step 3: Setup a cloud server that receives and stores the videos recorded by the Raspberry Pi.

Configuration/ External Interfaces

External Interfaces

- Configure HD surveillance camera

Dependencies

- HD camera that can be configured with Raspberry Pi. Not all cameras are configurable with Raspberry Pi

Debug

Logging

The server running on Raspberry Pi will be configured with a logger that logs all the activities that are performed by the server. The logs will entail information such as:

- Status of the server
- Time of the video recorded
- Time taken for the video to be transferred to the cloud server
- If the file transfer was successful or not

These logs will be useful in identifying bugs and fixing issues at the developing stage. Once the application is deployed these logs will be useful for the users to navigate through the history of the server.

Implementation

The implementation of this functionality has been briefly described in 3 different steps above. These steps can be further divided into sub-tasks to achieve the deployment of the functionality.

Step 1 – Deploying HD surveillance camera
Deploying a HD surveillance camera wifi connection. The camera will be connected to the home network. The camera transfers the video files to the Raspberry Pi over the network when recorded. This step does not require any additional coding and can be deployed with some simple steps.

Step 2 – Raspberry Pi Setup

This task is further divided into following sub-tasks:

i) **Install Raspbian OS on Raspberry Pi**
For enabling motion detection feature for the surveillance camera we will be installing a Linux based OS. Raspbian OS seems to be the optimum choice as it's a Linux based operating system and also optimized for Raspberry Pi hardware.

ii) **Enable Secure SSH**
The next step is to enable ssh on the Pi for it to be accessible from any system via SSH. This can be easily achieved by enabling SSH in Raspbian OS. Once the SSH has been enabled, a fixed IP can be assigned to pi in order to make the connection even after the pi gets restarted.

iii) **Motion Sensor**
Raspbian provides motion installation that can be configured with the USB camera. Various actions can be defined which will be triggered whenever a movement is detected. The parameters are to be defined in the motion.conf file.

iv) **FTP video files to cloud**
Due to the limited access to the raspberry pi from outside network, we will be transferring the recorded video files to the cloud server, which could easily be accessed by users from any location with Internet access. For this implementation, we will be using Amazon S3 server. We will be creating a service that transfers the new video that is generated to the S3 server.

Step 3 – Setup Cloud Server
This task will involve creating Amazon S3 server and make it accessible to our pi. This server will receive files from the pi and store in order of day/time. This server will also be made open to the mobile application. Users would be able to access the stored video files and watch/remove selected videos instantly.

Testing

General Approach

Configuration Testing
This functionality includes multiple installations and configurations. For testing these configuration various methods are used that ensures that the device has been configured as desired.

i) Testing Camera Configuration
Once the camera has been physically attaché to the Raspberry Pi, the latest kernel needs to be installed by using the below commands

```
sudo apt-get update
sudo apt-get upgrade
```

Next step is to enable the camera support in the Raspbian configuration

```
sudo raspi-config
```

The installed camera can then be tested with below command, which displays five second preview from the camera and then takes a picture and saves it to a .jpg file.

```
raspistill -v -o test.jpg
```

Performance Testing

We will be running multiple tests on the Pi to evaluate its performance.

- Stress tests
- Paging/Swapping Tests
- WiFi Tests
- LAN Test
- Speed Test

Appendix

N/A