Watchman

Introduction:

There are many areas where security is of prime importance e.g. Bank locker security, Ammunition security, Jewelry security etc. The area where the valuables are kept must be secured. This project can provide the security to these areas. In this project camera will monitor the area where security is required. There should not be any movement in security area after the working hours. If movement is observed after working hours, there may be possibility of unauthorized activity. This activity is monitored by camera and security alert sms will be send to security guard and responsible authority as well alarm will be given to the security room. And all doors will get locked as well as recording will start.

Methodology:

Block Diagram:

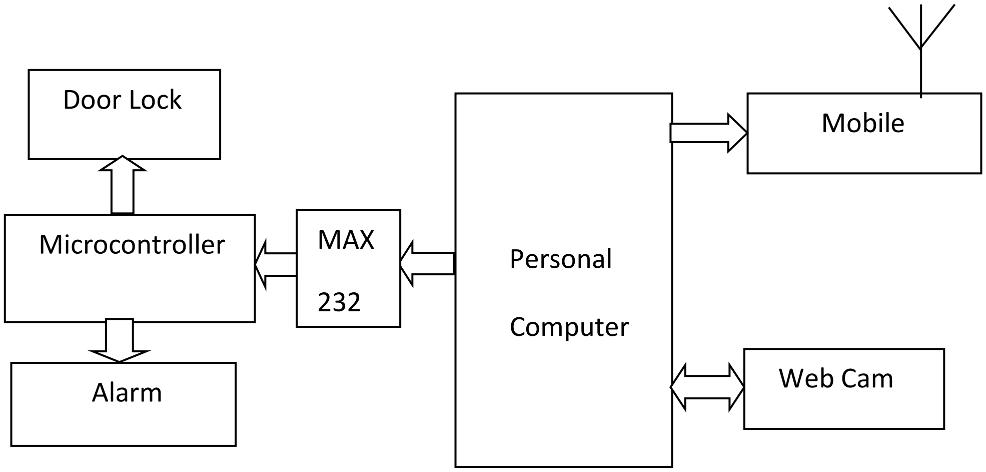

Fig. Block Diagram of Security Alert System

Figure shows block diagram of Security Alert System. It consists of mobile phone or GSM modem, Personal computer, web cam, Microcontroller, PC and Microcontroller interface, Alarm system, Door lock system.

1) Mobile Phone or GSM Modem:

   This is use to send alert sms when command is given by computer. Any mobile phone with inbuilt model can be used provided that modem should support standard AT commands [1]. AT commands are the commands for mobile phone.

e.g. If we give AT command to mobile, in response it gives OK massage. This indicate successful communication between mobile and PC.

AT+CMGR = 1 command is used to read the sms from the first memory location of the sim card.

In general these commands are used to communicate with mobile via PC. For detail of AT commands reference 1 should be referred.

2) Personal computer:

This is main controlling unit of the project. The program is written in MATLAB to control the activity. This will initialize the web cam and monitor the status of security area through web cam. If movement is occurs in the security area the sms will be send to authority. It will also give command to microcontroller to actuate alarm system.

MATLAB [2]:

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include:

- Math and computation
- Algorithm development
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including Graphical User Interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar noninteractive language such as C or Fortran.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects, which together represent the state-of-the-art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of application-specific solutions called toolboxes. Very important to most users of MATLAB, toolboxes allow you to *learn* and *apply* specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

3) Web cam:

Web cam is the main monitoring eye of the system. Image of security area will be taken by web cam. This image will be processed by image processing tool of matlab. According to result decision will be taken.

To initialize the web cam following procedure should be used.

Basic Image Acquisition Procedure

Overview

This section illustrates the basic steps required to create an image acquisition application by implementing a simple motion detection application. The application detects movement in a scene by performing a pixel-to-pixel comparison in pairs of incoming image frames. If nothing moves in the scene, pixel values remain the same in each frame. When something moves in the image, the application displays the pixels that have changed values.

The example highlights how you can use the Image Acquisition Toolbox software to create a working image acquisition application with only a few lines of code.

Note To run the sample code in this example, you must have an image acquisition device connected to your system. The device can be a professional grade image acquisition device, such as a frame grabber, or a generic Microsoft® Windows® image acquisition device, such as a webcam. The code can be used with various types of devices with only minor changes.

To use the Image Acquisition Toolbox software to acquire image data, you must perform the following basic steps.

Step	Description
Step 1:	Install and configure your image acquisition device
Step 2:	Retrieve information that uniquely identifies your image acquisition device to the Image Acquisition Toolbox software
Step 3:	Create a video input object
Step 4:	Preview the video stream (Optional)
Step 5:	Configure image acquisition object properties (Optional)
Step	Acquire image data

Step	Description
6:	
Step 7:	Clean up

Step 1: Install Your Image Acquisition Device

Follow the setup instructions that come with your image acquisition device. Setup typically involves:

- Installing the frame grabber board in your computer.
- Installing any software drivers required by the device. These are supplied by the device vendor.
- Connecting a camera to a connector on the frame grabber board.
- Verifying that the camera is working properly by running the application software that came with the camera and viewing a live video stream.

Generic Windows image acquisition devices, such as webcams and digital video camcorders, typically do not require the installation of a frame grabber board. You connect these devices directly to your computer via a USB or FireWire port.

After installing and configuring your image acquisition hardware, start MATLAB on your computer by double-clicking the icon on your desktop. You do not need to perform any special configuration of MATLAB to perform image acquisition.

Step 2: Retrieve Hardware Information

In this step, you get several pieces of information that the toolbox needs to uniquely identify the image acquisition device you want to access. You use this information when you create an image acquisition object, described in Step 3: Create a Video Input Object.

The following table lists this information. You use the imaqhwinfo function to retrieve each item.

Device Information	Description
Adaptor name	An *adaptor* is the software that the toolbox uses to communicate with an image acquisition device via its device driver. The toolbox includes adaptors for certain vendors of image acquisition equipment and for particular classes of image acquisition devices. See Determining the Adaptor Name for more information.
Device ID	The *device ID* is a number that the adaptor assigns to uniquely identify each image acquisition device with which it can communicate. See Determining the Device ID for

Device Information	Description
	more information. **Note** Specifying the device ID is optional; the toolbox uses the first available device ID as the default.
Video format	The *video format* specifies the image resolution (width and height) and other aspects of the video stream. Image acquisition devices typically support multiple video formats. See Determining the Supported Video Formats for more information. **Note** Specifying the video format is optional; the toolbox uses one of the supported formats as the default.

Determining the Adaptor Name

To determine the name of the adaptor, enter the imaqhwinfo function at the MATLAB prompt without any arguments.

```
imaqhwinfo
ans =

    InstalledAdaptors: {'dcam'  'winvideo'}
        MATLABVersion: '7.4 (R2007a)'
          ToolboxName: 'Image Acquisition Toolbox'
       ToolboxVersion: '2.1 (R2007a)'
```

In the data returned by imaqhwinfo, the InstalledAdaptors field lists the adaptors that are available on your computer. In this example, imaqhwinfo found two adaptors available on the computer: 'dcam' and 'winvideo'. The listing on your computer might contain only one adaptor name. Select the adaptor name that provides access to your image acquisition device. For more information, see Determining the Device Adaptor Name.

Determining the Device ID

To find the device ID of a particular image acquisition device, enter the imaqhwinfo function at the MATLAB prompt, specifying the name of the adaptor as the only argument. (You found the adaptor name in the first call to imaqhwinfo, described in Determining the Adaptor Name.) In the data returned, the DeviceIDs field is a cell array containing the device IDs of all the devices accessible through the specified adaptor.

> **Note** This example uses the DCAM adaptor. You should substitute the name of the adaptor you would like to use.

```
info = imaqhwinfo('dcam')
info =

       AdaptorDllName: [1x77 char]
    AdaptorDllVersion: '2.1 (R2007a)'
          AdaptorName: 'dcam'
            DeviceIDs: {[1]}
           DeviceInfo: [1x1 struct]
```

Determining the Supported Video Formats

To determine which video formats an image acquisition device supports, look in the DeviceInfo field of the data returned by imaqhwinfo. The DeviceInfo field is a structure array where each structure provides information about a particular device. To view the device information for a particular device, you can use the device ID as a reference into the structure array. Alternatively, you can view the information for a particular device by calling the imaqhwinfo function, specifying the adaptor name and device ID as arguments.

To get the list of the video formats supported by a device, look at SupportedFormats field in the device information structure. The SupportedFormats field is a cell array of strings where each string is the name of a video format supported by the device. For more information, see Determining Supported Video Formats.

```
dev_info = imaqhwinfo('dcam',1)

dev_info =

          DefaultFormat: 'F7_Y8_1024x768'
    DeviceFileSupported: 0
             DeviceName: 'XCD-X700  1.05'
               DeviceID: 1
      ObjectConstructor: 'videoinput('dcam', 1)'
       SupportedFormats: {'F7_Y8_1024x768'  'Y8_1024x768'
```

Step 3: Create a Video Input Object

In this step you create the video input object that the toolbox uses to represent the connection between MATLAB and an image acquisition device. Using the properties of a video input object, you can control many aspects of the image acquisition process. For more information about image acquisition objects, see Connecting to Hardware.

To create a video input object, use the videoinput function at the MATLAB prompt. The DeviceInfo structure returned by the imaqhwinfo function contains the default videoinput function syntax for a device in the ObjectConstructor field. For more information the device information structure, see Determining the Supported Video Formats.

The following example creates a video input object for the DCAM adaptor. Substitute the adaptor name of the image acquisition device available on your system.

```
vid = videoinput('dcam',1,'Y8_1024x768')
```

The videoinput function accepts three arguments: the adaptor name, device ID, and video format. You retrieved this information in step 2. The adaptor name is the only required argument; the videoinput function can use defaults for the device ID and video format. To determine the default video format, look at the DefaultFormat field in the device information structure. See Determining the Supported Video Formats for more information.

Instead of specifying the video format, you can optionally specify the name of a device configuration file, also known as a camera file. Device configuration files are typically supplied by frame grabber vendors. These files contain all the required configuration settings to use a particular camera with the device. See Using Device Configuration Files (Camera Files) for more information.

Viewing the Video Input Object Summary

To view a summary of the video input object you just created, enter the variable name (vid) at the MATLAB command prompt. The summary information displayed shows many of the characteristics of the object, such as the number of frames that will be captured with each trigger, the trigger type, and the current state of the object. You can use video input object properties to control many of these characteristics. See Step 5: Configure Object Properties (Optional) for more information.

```
vid

Summary of Video Input Object Using 'XCD-X700  1.05'.

   Acquisition Source(s):  input1 is available.

  Acquisition Parameters:  'input1' is the current selected source.
                           10 frames per trigger using the selected source.
                           'Y8_1024x768' video data to be logged upon START.
                           Grabbing first of every 1 frame(s).
                           Log data to 'memory' on trigger.

Trigger Parameters:  1 'immediate' trigger(s) on START.
        Status:  Waiting for START.
              0 frames acquired since starting.
        0 frames available for GETDATA.
```

Step 4: Preview the Video Stream (Optional)

After you create the video input object, MATLAB is able to access the image acquisition device and is ready to acquire data. However, before you begin, you might want to see a preview of the video stream to make sure that the image is satisfactory. For example, you might want to change the position of the camera, change the lighting, correct the focus, or make some other change to your image acquisition setup.

> **Note** This step is optional at this point in the procedure because you can preview a video stream at any time after you create a video input object.

To preview the video stream in this example, enter the preview function at the MATLAB prompt, specifying the video input object created in step 3 as an argument.

```
preview(vid)
```

The preview function opens a Video Preview figure window on your screen containing the live video stream. To stop the stream of live video, you can call the stoppreview function. To restart the preview stream, call preview again on the same video input object.

While a preview window is open, the video input object sets the value of the Previewing property to 'on'. If you change characteristics of the image by setting image acquisition object properties, the image displayed in the preview window reflects the change.

4) Interface between PC and Microcontroller [3]

   MAX 232 is used as interface between PC and Microcontroller as the signal slandered of PC and microcontrollers are different.

   MAX 232[3].

The MAX232 is a dual driver/receiver that includes a capacitive voltage generator to supply EIA-232 voltage levels from a single 5-V supply. Each receiver converts EIA-232 inputs to 5-V TTL/CMOS levels. These receivers have a typical threshold of 1.3 V and a typical hysteresis of 0.5 V, and can accept ±30-V inputs. Each driver converts TTL/CMOS input levels into EIA-232 levels. The driver, receiver, and voltage-generator functions are available as cells in the Texas Instruments LinASIC□ library

5) Microcontroller:

   Command will be given to microcontroller through PC to actuate alarm system and door lock
   Here AT89C2051 Microcontroller will be used.
   AT89C2051[4]:

The AT89C2051 is a low-voltage, high-performance CMOS 8-bit microcomputer with 2K Bytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high density nonvolatile memory technology and is compatible with the industry standard MCS-51™ instruction set. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C2051 is a powerful microcomputer which provides a highly flexible and cost effective solution to many embedded control applications. The AT89C2051 provides the following standard features: 2K Bytes of Flash, 128 bytes of RAM, 15 I/O lines, two 16-bit timer/counters, a five vector two-level interrupt architecture, a full duplex serial port, a precision analog comparator, on-chip oscillator and clock circuitry. In addition, the AT89C2051 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port and interrupt system to continue functioning. The Power Down Mode saves the RAM contents but freezes the oscillator disabling all other chip functions until the next hardware reset.

6) Alarm:

Alarm is use to alert security guard that somebody is there in the security area.

7) Door lock:

When movement is observed all doors get locked by the system.

Referances:

[1] AT Command set for Nokia GSM Product : Copyright 2000. Nokia Mobile Phones.

[2] http://cimss.ssec.wisc.edu/wxwise/class/aos340/spr00/whatismatlab.htm

[3] MAX232, MAX232I DUAL EIA-232 DRIVERS/RECEIVERS: TEXAS INSTRUMENTS: SLLS047I – FEBRUARY 1989 – REVISED OCTOBER 2002.

[4] Datasheet : ATMEL : 8-Bit Microcontroller with 2K Bytes Flash

Biliography:

Digital Image processing: Rafael C. González, Richard Eugene Woods

The 8051 microcontroller and embedded system: Muhammad Ali Mazidi,Rolin D.McKinlay

www.mathswork.com

http://www.cs.utah.edu/~croberts/courses/cs7966/index.html

http://appnote.avrportal.com/appnotes/Image-processing/An-Introduction-to-Digital-Image-Processing-with-Matlab-Notes-for-SCM2511-Image-Processing.pdf