RESEARCH PAPER | Engineering | Volume : 5 | Issue : 9 | September 2015 | ISSN - 2249-555X

Sound Based Home Automation

KEYWORDS: Home Automation System, MATLAB, Arduino microcontroller, Acoustic signal processing

Nisha Patel	Laisha Prakash
Student at VIT, Vellore	Student at VIT, Vellore

ABSTRACT *The growing need for automation in our day to day lives in an easier way has made us design a prototype for controlling home devices with the mere use of sounds than voice signals. Aimed at providing a simple and easy to use technology to the elderly, the processing is faster and also cost effective. Implemented using MATLAB interfaced with a microcontroller for processing signals, the prototype has been designed to accept sound signals based on their frequency and peaks, and can be modified to accept sound of any other type to control any electrical appliance in homes, hospitals, or offices. With the increasing need to automate the mundane, this paper presents an effective solution to achieving a device for home automation.*

INTRODUCTION

In this project we have tried to implement an intelligent and self-capable system for home automation based on sound recognition and processing using peak detection technique in MATLAB.

The MATLAB program distinguishes a given characteristic signal from other sounds by detecting defined peaks only and these peaks are considered in an interval above a predefined frequency. It converts all the other signals to zero and considers the highest peak above the threshold.

The number of peaks thus detected is sent to microcontroller interfaced via a serial port communication established between the Human Machine Interface and Microcontroller. Based on the number of signals, the following actions are taken in our prototype 1-to switch on light, 2-to switch on fan and 3-to turn either the fan or the light or both off. The functionality can be easily modified according to ones requirement. The complete working is discussed in details in the paper.

HARDWARE DESIGN:

The microcontroller selected for this prototype is Arduino UNO, for it can be customized as per requirement. *It* offers a variety of digital and analog inputs, *SPI* and serial interface and digital and *PWM* outputs so a variety of input signals can be used. Arduino Uno is not expensive so it provides a cost-effective solution and is easier to program. [1]

We used the serial port interface, the USB to connect this microcontroller to the computer to enable communication between the microphone, MATLAB program and itself during the process of exchange of signals. Using Arduino support package [2] available, we interfaced arduino microcontroller with MATLAB. A server program running on the board, listens to commands arriving via USB from MATLAB, executes the commands, gives the necessary output, and stops communication [3]. In Universal Serial Bus, data transmission in the bus occurs in a serial form. Bytes of data are broken up and sent along the bus one bit at a time, with the least significant bit first.

For Processing the input received on the Dell Inspiron internal microphone, the MATLAB uses signal processing techniques. After processing it sends the output to Arduino via USB.

L293D- This motor driver circuit can be used to control two motors, fan prototype is connected on one side of the IC. The datasheet is obtained [5] and a motor for fan is taken of a rating less than 5Volt. The LEDs (0.7 V dc) are connected in parallel along with the resistors to prevent the LEDs from heating damage due to excessive current drawn. The arduino Microcontroller presents output in the form of lights and fan. Resistors of 10k Ohm have been used to limit the current drawn.

SOFTWARE AND IT'S WORKING:

By default, the Adafruit\MotorShieldV2 library is not included in the server code on the board. Since we have used an [6] arduino Uno board, an arduino object and include the Adafruit\MotorShieldV2 library to the hardware. This library was used as it can handle two stepper motors a better option for our prototype. It's a cheap and easy way to start controlling actuators via MATLAB.

To establish serial communication between [7] microcontroller and MATLAB, the microcontroller must be connected to the serial port using FOPEN() function. The block input accepts vector or scalar uint8 data. [10]When the serial port object is constructed, the object's Status property is closed. Once the object is connected to the serial port with the FOPEN() [8] [9] function, the Status property is configured to open.

Audiorecorder() function was used to create 8000 Hz, 8-bit and single channel audiorecorder object since our recording device is a micro-phone inbuilt in a laptop. The recorded sound is stored in a variable whose absolute value was calculated. Using the distribution of absolute value of normal random variable, the contrast between high amplitude peaks and low amplitude peaks was increased. A lot of high frequency data can be seen around a sharp peak point. Next, to eliminate low amplitude peaks, all of them were equated to noise. Our aim was to count the number of claps, number of sharp peak points.

Initially, the sample was divided by average value of all surrounding samples. This accentuated the peaks and flattened the noise to some extent.

To find peaks in a data, a function findpeaks() is inbuilt in MATLAB library. A minimum peak height was included, since we needed the number of peaks above a particular frequency. Also, by specifying a minimum peak height the processing time is reduced. Next, to eliminate the high frequency redundant data, all the peaks not separated by a minimum peak distance were eliminated. This parameter ignored the smaller peaks within a close proximity of a larger peak. After an extensive trial and error, an MPH of 0.3 and MPD of 2500 worked best for our prototype of claps. The number of peaks was finally counted and the output was serially communicated to micro controller using USB. A loop can be added in the MATLAB program to repeat the process intermittently.

A small C script was written and run on the Arduino to watch the incoming serial data, react to instructions fired in from MATLAB, send the direction and instruction (number of steps to peaks) signals onward to each of the motors.

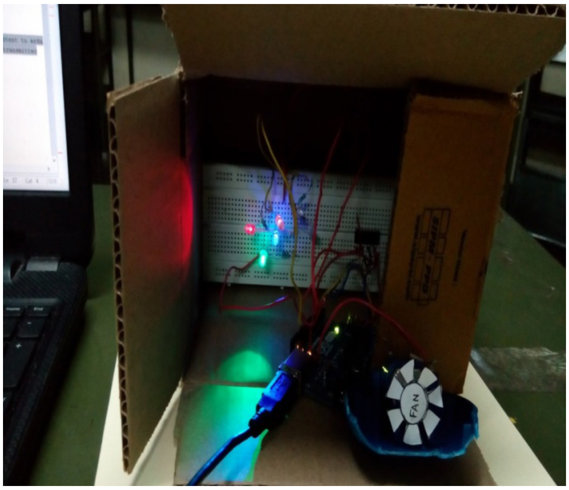
The serial communicated is finally ended using fclose() and all the instrument objects and device group objects were deleted. To avoid error, the asynchronous serial transmission between Rx and Tx in process with the micro controller connected to the serial port is stopped.

Input	Output	
No. of peaks detected	Appliance A	Appliance B
1	1	0
2	0	1
3	0	0

RESULTS:

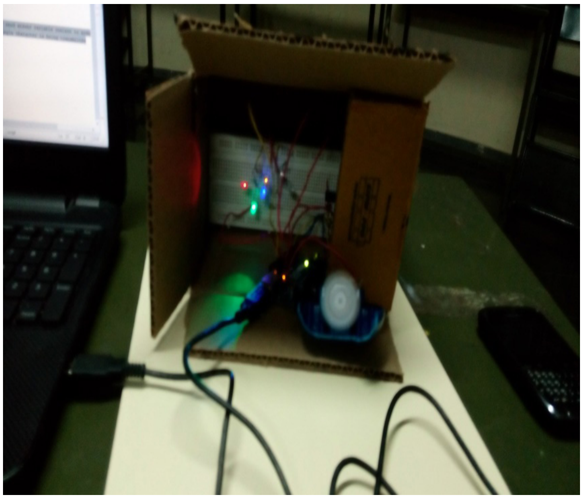
When number of peaks = 1

No. of peaks	1
Bits per Sample	8
Buffer Length	0.1000
Current Sample	1
DeviceID	-1
Number Of Buffers	10
Number of Channels	1
Running	'off'
Sample Rate	8000
Timer Period	0.0500
Total Samples	0
Peaks	0.4842



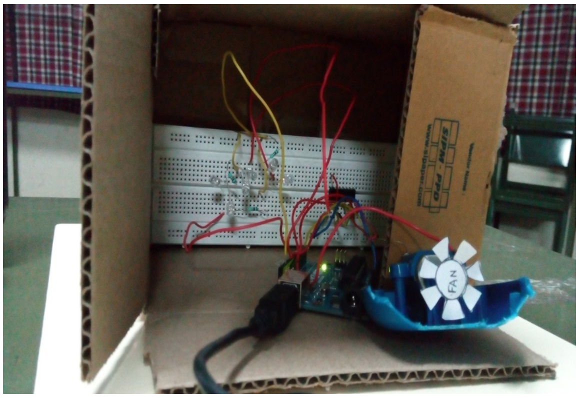
When number of peaks = 2

No. of peaks	2
Bits per Sample	8
Buffer Length	0.1000
Current Sample	1
DeviceID	-1
Number Of Buffers	10
Number of Channels	1
Running	'off'
Sample Rate	8000
Timer Period	0.0500
Total Samples	0
Peaks	0.3879, 0.7882



When number of peaks = 3

No. of peaks	3
Bits per Sample	8
Buffer Length	0.1000
Current Sample	1
DeviceID	-1
Number Of Buffers	10
Number of Channels	1
Running	'off'
Sample Rate	8000
Timer Period	0.0500
Total Samples	0
Peaks	0.5705, 0.7882, 0.6823



APPLICATIONS

- The above prototype can be implemented easily in most homes.
- It is aimed at providing a more convenient solution to the elderly , who cannot walk up to the switchboard to switch on the fan and light will find it helpful where understanding the complex functioning is eliminated by use of simpler signals such as a knock or a clap signal.
- It can also be used by hospitals to implement a cost effective method of alarming the doctors when their patients are unable to make the call, or when they immediately need the ventilator or such other cares.
- It can even be used to save energy resources by the use of proximity sensor, which when added can control the switching action based on the number of people in the room

FUTURE PROSPECTS:

- If this system is connected through internet, it can be used for security purposes such as for initiating alarms in emergencies like hostages where cell phones are inaccessible.
- In cases where patient's life is crucial to time, this system can also be used extensively by doctors when they are unavailable to switch on healthcare instruments like ventilators till the time they reach the patient.

REFERENCE [1] http://www.varesano.net/blog/Fabio | [2]http://in.mathworks.com/hardware-support/arduino-matlab.html | [3]http://in.mathworks.com/matlabcentral/fileexchange/32374-legacy-matlab-and-simulink-support-for-arduino | [4]http://www.digitaldandt.org/index.php?option=com_content&view=article&id=248:l293d-motor-driver&catid=51 | [5]http://www.digitaldandt.org/index.php?option=com_content&view=article&id=248:l293d-motor-driver&catid=51 | [6]http://in.mathworks.com/help/supportpkg/arduino/ug/pin-mapping-on-arduino-blocks.html | [7]http://in.mathworks.com/help/supportpkg/arduino/ug/use-serial-communications-with-arduino-hardware.html | [8]http://www.mathworks.com/matlabcentral/fileexchange/32374-legacy-matlab-and-simulink-support-for-arduino | [9]http://in.mathworks.com/help/supportpkg/arduinoio/examples/control-motors-using-adafruit-motor-shield-v2.html | [10]http://principialabs.com/arduino-python-4-axis-servo-control/ |