NON INVASIVE MONITORING SYSTEM FOR SLEEPING SUBJECTS

Submitted in partial fulfillment of the Degree of

Bachelor of Technology

May – 2015

Enrollment No.	-	111002
Name of Student	-	Pranjali Khattri
Name of supervisor	-	Dr. T. S. Lamba

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY,

WAKNAGHAT

CERTIFICATE

This is to certify that project report entitled "NON INVASIVE MONITORING SYSTEM FOR SLEEPING SUBJECTS", submitted by Pranjali Khattri (111002) in partial fulfillment for the award of degree of Bachelor of Technology in Electronics and Communication Engineering to Jaypee University of Information Technology, Waknaghat, Solan has been carried out under my supervision.

This work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma.

DATE:

Supervisor's Name: Dr. T. S. Lamba

Designation: Dean (Academic and Research)

ACKNOWLEDGEMENT

It is indeed a matter of great privilege and honor for me to express my sincere and deepest acknowledgements towards all those who have contributed either directly or indirectly towards the completion of my project.

On the very outset of the report , I would like to extend our sincere and heartfelt obligation towards Prof. T.S. Lamba who helped me in this endeavor. Without his active guidance, help, cooperation and encouragement, I would not have made headway in the project. I am grateful to him for his even willingness to give me valuable advice and direction whenever I approached him with a problem. I am thankful to him for providing immense guidance for this project. He has motivated me throughout the entire duration of my work and this could not have been possible without his blessed guidance and unmatched wealth of knowledge.

I would also thank Mr. Mohan for immense cooperation and motivation by providing all the necessary information for developing a good system.

Lastly, I would like to thank the almighty and my parents for their moral support and my friends with whom I shared my day to day experience and received lots of suggestions that improved my quality of work

TABLE OF CONTENTS

(I)

SUMMARY

This report focuses on non invasive methods for monitoring a person in sleep. The main focus is on two age groups; infants and elderly people where a system is required to monitor their body activities and set the alarm in case of any malfunctioning. The motive for using non invasive technique was as it doesn't harm human body in any way and is used to identify malfunction by not penetrating the skin.

Sleep monitoring is very important for elderly people as inadequate and irregular sleep are often related to serious diseases such as depression and diabetes. In many cases, it is necessary to monitor the body positions and movements made while sleeping because of their relationships to particular diseases (i.e., sleep apnea and restless legs syndrome). Analyzing movements during sleep also helps in determining sleep quality and irregular sleeping patterns.

The first module uses Arduino Mega 2560 and is interfaced with pulse counter, we get the set of values which is studied and interpreted and used to issue the alarm. The alarm is issued if the pulse rate goes too hign or too low.

The next module is measurement of body temperature using sensor LM35. It is again interfaced with the microcontroller to issue the alarm if the temperatures goes beyond the safe limits.

The other module is motion detection using Matlab. A video object is created using webcam and then it is processed to detect motion. If the motion is detected frequently, it implies that the person is having disrupted sleep due to some reason so again the alarm is issued.

These three modules combined form a monitoring system to measure motion, temperature and heart rate non invasively.

__________________	__________________
Signature of Student	Signature of Supervisor
Name	Name
Date	Date

(II)

LIST OF FIGURES

INTRODUCTION

New technology offers more and more possibilities to measure variables on the human body and mind in a non-invasive way as a basis for health management. By using miniaturized sensors, several variables can be measured such as heart rate, skin temperature, movements etc. Several sensing techniques (image analyses, sound analyses, etc) allow to measure other variables (such as posture, movements, facial expression) and sound production. Also at the other end of the scale, new technologies (e.g. remote sensing technology) in combination with smart algorithms offer possibilities for monitoring human health. By applying this technology several easy measurable variables can be monitored continuously in a fully automated way.

Heart rate is, among the many vital signs (respiration rate, blood oxygen saturation, arterial blood pressure, etc), one of the most commonly measured and monitored. Whatever will be the sensing principle or the monitoring method used, data referred to the heart rate can be considered the primary vital sign information which is needed on a patient approach in both emergency and clinical situations. Heart rate data are used to measure anomalous rate or irregular pulse rate (arrhythmias) or heart block. The post-processing of the data can be used to verify trends or single events, providing precious elements to the patient diagnosis. Heart-rate variability (HRV) can be performed on recorded data in order to have an objective measure of eventual cardiac abnormalities. Other possible use of the heart rate data are related to the analysis of the circadian rhythm (sleep), temperature regulation, cardiac sympathetic nervous activity and synchronization with respiration rate.

Monitoring the movements of the human during sleep can potentially give us a good estimation of aspects of the bodily as well as mental state of a human. When such data are combined, either with the knowledge of a sleep pathologist or with a special automated diagnosis system, they could prove quite useful towards the diagnosis of various types of sleep disorders such as parasomnias, insomnia, and dyspnea. Furthermore, such data could also be useful towards diagnosis of various medical conditions, and towards quantitative evaluation of the effects of drug therapy that is administered to a patient who is suffering from poor sleep quality, an important indication of which is the duration and patterns of various sleep stages.

The temperature of both the brain and the body fall during NREM sleep. The longer the NREM-sleep episode, the more the temperature falls. By contrast, brain temperature increases during REM sleep.

The control of body and brain temperature is closely tied to sleep regulation. Human beings are endotherms (able to thermoregulate) , that is, maintain their body temperature. Body temperature is regulated through a balance of heat absorption, production and loss. Human temperature must be maintained within a fairly small range, up or down from the resting temperature of 98.6. Temperatures above 104.9 degrees Fahrenheit or below 92.3 degrees generally cause injury or death.

SLEEP AND ITS ANALYSIS

Sleep is a behavioral state that is a natural part of every individual's life. We spend about one-third of our lives asleep. Nonetheless, people generally know little about the importance of this essential activity. Sleep is not just something to fill time when a person is inactive. Sleep is a required activity, not an option. It is important for normal motor and cognitive function.

Sleep is divided into two broad types: rapid eye movement (REM sleep) and non-rapid eye movement (NREM or non-REM sleep). Each type has a distinct set of physiological and neurological features associated with it. Usually sleepers pass through five stages: 1, 2, 3, 4 and REM (rapid eye movement) sleep. These stages progress cyclically from 1 through REM then begin again with stage 1. A complete sleep cycle takes an average of 90 to 110 minutes. The first sleep cycles each night have relatively short REM sleeps and long periods of deep sleep but later in the night, REM periods lengthen and deep sleep time decreases.

Stage 1, light sleep where you drift in and out of sleep and can be awakened easily. In this stage, the eyes move slowly and muscle activity slows. During this stage, many people experience sudden muscle contractions preceded by a sensation of falling.

Stage 2, eye movement stops and brain waves become slower with only an occasional burst of rapid brain waves.

Stage 3, extremely slow brain waves called delta waves are interspersed with smaller, faster waves.

Stage 4, the brain produces delta waves almost exclusively.

In the REM period, breathing becomes more rapid, irregular and shallow, eyes jerk rapidly and limb muscles are temporarily paralyzed. Brain waves during this stage increase to levels experienced when a person is awake. Also, heart rate increases, blood pressure rises, males develop erections and the body loses some of the ability to regulate its temperature. This is the time when most dreams occur.

Sleep plays an important role in quality of life, and is an important factor in staying healthy, active, and energetic. Having inadequate and irregular sleeping patterns has a serious impact on our health, and can lead to many serious diseases like cardiovascular disease, diabetes, depression, and obesity. Besides the amount of sleep, it is also necessary to have sound sleep. Despite sleeping for a sufficient amount of time, people can still feel fatigued and cannot concentrate during the day. This may be caused by interrupted sleep, such as having frequent periods of restlessness during sleep. Sleep monitoring systems are important to recognize sleeping disorders as early as possible for diagnosis and prompt treatment of disease. They can provide healthcare providers with quantitative data about irregularity in sleeping periods and durations. They can also provide detailed sleeping profiles that depict periods of restlessness and interruptions such as bed exits and entries due to visiting the bathroom. Moreover, it enables monitoring effectiveness of treatments to sleep-related diseases. Many studies are focused on finding correlations between body positions during sleep to various breathing problems (e.g., sleep apnea). So, if a sleep monitoring system can provide fine grained information about body positions during sleep, it would help such studies. To date, there are very few low-cost, unobtrusive sleep monitoring systems. Among the existing ones, the most accurate and reliable are polysomnography (e.g., electroencephalogram, electrooculogram, electromyogram) devices. There are drawbacks to using them, since they need to be worn, and require professional monitoring and thus are expensive to use.

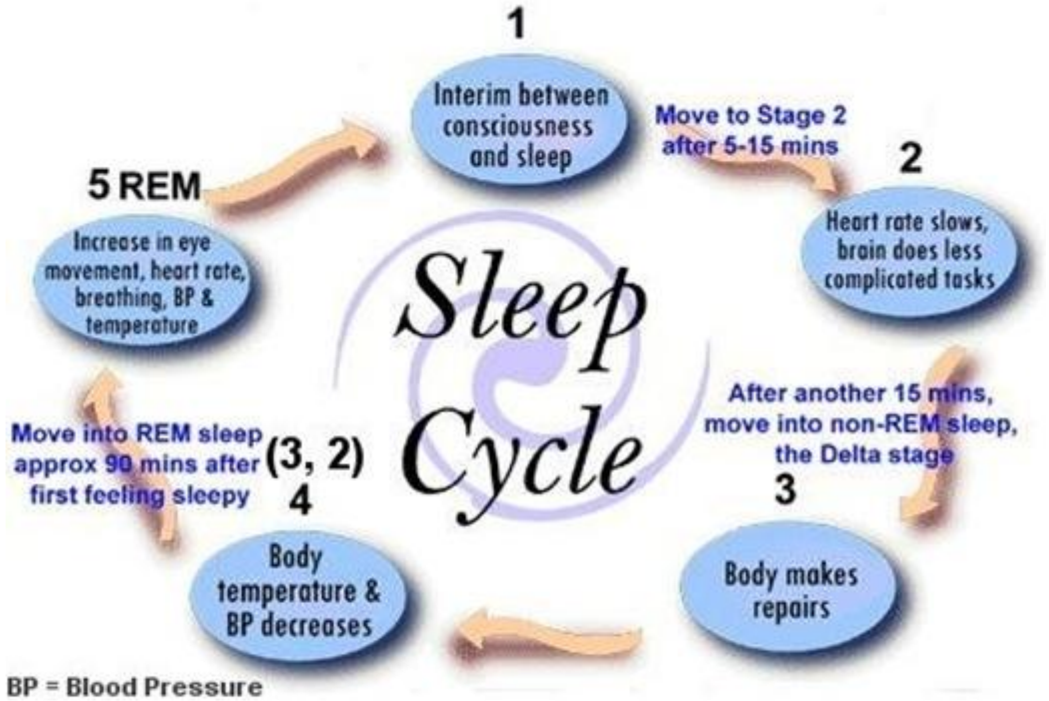

1
Interim between consciousness and sleep
Move to Stage 2 after 5-15 mins
2
Heart rate slows, brain does less complicated tasks
After another 15 mins, move into non-REM sleep, the Delta stage
3
Body makes repairs
4
Body temperature & BP decreases
(3, 2)
Move into REM sleep approx 90 mins after first feeling sleepy
5 REM
Increase in eye movement, heart rate, breathing, BP & temperature
Sleep Cycle
BP = Blood Pressure

Figure 1

TECHNICAL DETAILS

BODY MOTION DECTECTION

COMPUTER VISION

Many processes as image compression, restoration, enhancement and analysis are used in dealing with the digital image. Digitizing an analog video signal is to convert the signal to be compatible with computer media to store and restore the signal, this is done by sampling and quantization. The value of the analog signal at each instant is converted pixels, the smallest element of an image. After the process of digitizing is completed, we have 2D of data constructed of pixels, which contain the color information in each position of the image.

I(x, y) = the brightness of image at the point (x, y). Where x= row and y= column.

Each colored image is stored in three dimensional array MxNx3, number 3 comes from creating three layers of the basics colors, Red, Green and Blue. These layers are composed to represent the true color image. Each colored pixel composed of three values (red, green, blue), these values are having the ranged between 0-255.

MOTION DETECTION

This section describes four methods for extracting moving targets from real time video stream. The first approach is Background Subtraction which is the simplest but not without disadvantages that the second approach Temporal Differencing handle it. Then, a direct improvement has been added to temporal differencing to present a new algorithm called Fast Pixel Selection. Finally, an intelligent approach has been presented here that is based on Optic Flow method.

Background subtraction

The main steps of detecting the target system are:

1. Frames differencing.
2. Detecting the moving target.

Frames differencing: The idea is to take the absolute difference between a reference stored background and a new frame which is grabbed from the camera. If no target is entered, the result pixel is expected to be zero value (difference result), else, if a target exists, the resulting pixels will give a value different than zero.

$D = | I_{K\text{-}1} – I_K |$

K: The frame number, I: frames, D: difference result.

- **Detecting the moving target**

After taking the difference, the number of nonzero value pixels will specify the size of the target; if these pixels met the requirement of the system an action will be taken to alarm, or lights, etc.

If the number of the nonzero pixels did not approach the minimum number of the required pixels to decide, another frame is taken and goes on to the difference process again.

- **Disadvantages**

This is a fast algorithm and compatible with real time video analysis, but, many problems appears in using the still background, these can be summarized as follows:

1. Sensitivity to lights, where any lighting condition will cause the system to recognize it as a moving target.
2. Sensitivity to clutter, and image texture, the same thing will be done by the system in error decision.

Temporal Differencing

The temporal differencing is the next algorithm which will solve the still background problems. The basic principle of this method depends on extracting (two-frames or three-frames), and taking the first frame as background and the subtraction is applied between it and between the next two frames. So we have an updating background.

- **Motion Analysis**

Motion analysis using temporal difference is simply done by taking a consecutive video frames and determines the absolute difference. A threshold value is then chosen to determine the change.

- **Disadvantages**

Time consuming of temporal method cause that the method to be inapplicable with security system. Where the time of differencing two frames and taken the decision and possibility of using a third frame takes a lot of time. So the search goes on to find a better algorithm to enhance the working in still camera motion detection.

Fast Pixel Selection

If the frames were of big dimensions in a normal speed of measuring the difference between the consecutive frames to detect and there is a target in the zone, it will take a lot of time that may permit the target to pass the secure area without taking its frame because the system was busy in computing the difference in the previous frames. So the need for more a faster computing algorithm is required in order to recover the missing frames during that time.

The approach here depends on: the number of the white pixels, which are the output of the difference process where they decide if there is a target or not. So instead on taking the difference of all the pixels in the consecutive frames, it would be faster to take a selection of the pixels and reduce the decision number of pixels.

Since no target takes the size of one pixel or even ten pixels the pixel differencing can be spanned, such that the pixels in each row can be taken after ignoring a number of them and the same thing can be done with the columns.

The pixel selection can be made by taking not the adjacent rows or columns of the pixels in the frames this will speed the process of differencing to four times the normal differencing.

Optical Flow

This method based on the differential methods and the region-based matching methods [17]. The differential methods effectively track intensity gradients. The advantage of this is that flow vectors can be determined based on the information (measured by intensity gradient).

With reliable flow vectors for every pixel in an object it becomes possible to track individual pixels from frame to frame. This capability can be employed to cluster of pixels into “body parts” for model-based motion analysis. It also means that an object’s rigidity can be determined by calculating residual flow that is the motion of “body parts” relative to the target gross motion .

Detection of bodies in a video stream is performed by real-time human motion analysis. This is basically a process of background subtraction using a dynamic updating background model. First, each frame is smoothed with a (3X3) Gaussian filter to remove video noise (as lights effect or tree vibration). Then the differencing method is applied. After this, non-moving pixels are updated using the Gaussian filter to reflect changes in the scene (such as lighting).

The idea is to use edge gradients as a measure of information and, at every pixel position, grow the support region until there is enough edge gradient information to justify matching. Furthermore, flow needs to be computed only for pixels contained within the moving object. Consequently, this particular implementation is only valid for video streams derived from stationary cameras, or streams which have been stabilized.

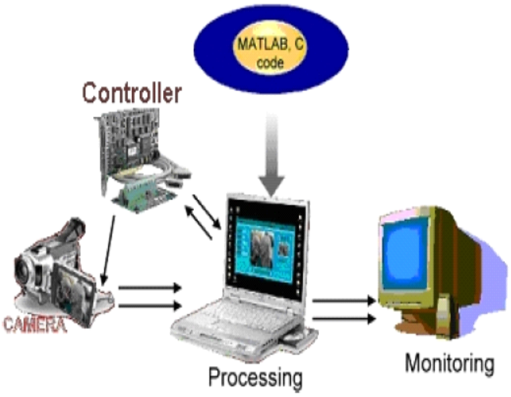


Figure 2

HEART RATE MEASUREMENT

Heart Rate, determined by the number of times heart beats in a minute, is substantial, as it gives information on the overall physical condition of the body. In case of patients suffering from diseases, continuous or routine measurement of heart rate is very important. Analysis of heart rate would help maintain health, diagnose and detect coronary diseases.Heart rate is commonly measured by doctors by feeling and counting the pulse of the pulsating arterial blood in suitable parts of the body such as wrist and neck.

While the heart is beating, it is actually pumping blood throughout the body, and that makes the blood volume inside the finger artery to change too. This fluctuation of blood can be detected through an optical sensing mechanism placed around the fingertip. The signal can be amplified further for the microcontroller to count the rate of fluctuation, which is actually the heart rate.

The sensor unit consists of an infrared light-emitting-diode (IR LED) and a photo diode, placed side by side, and the fingertip is placed over the sensor assembly, as shown below. The IR LED transmits an infrared light into the fingertip, a part of which is reflected back from the blood inside the finger arteries. The photo diode senses the portion of the light that is reflected back. The intensity of reflected light depends upon the blood volume inside the fingertip. So, every time the heart beats the amount of reflected infrared light changes, which can be detected by the photo diode. With a high gain amplifier, this little alteration in the amplitude of the reflected light can be converted into a pulse.

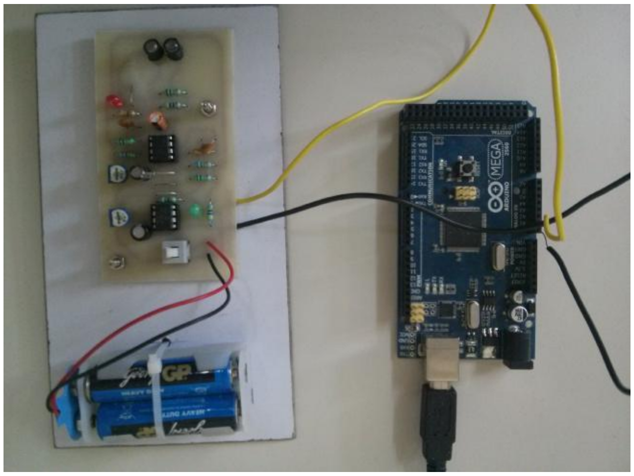


Figure 3

TEMPERATURE MEASUREMENT

A Digital Thermometer can be easily constructed using a PIC Microcontroller and LM35 Temperature Sensor. LM35 series is a low cost and precision Integrated Circuit Temperature Sensor whose output voltage is proportional to Centigrade temperature scale. Thus LM35 has an advantage over other temperature sensors calibrated in Kelvin as the users don't require subtraction of large constant voltage to obtain the required Centigrade temperature. It doesn't requires any external calibration. It is produced by National Semiconductor and can operate over a -55 °C to 150 °C temperature range. Its output is linearly proportional to Centigrade Temperature Scale and it output changes by 10 mV per °C.

The LM35 Temperature Sensor has Zero offset voltage, which means that the Output = 0V, at 0 °C. Thus for the maximum temperature value (150 °C), the maximum output voltage of the sensor would be 150 * 10 mV = 1.5V.

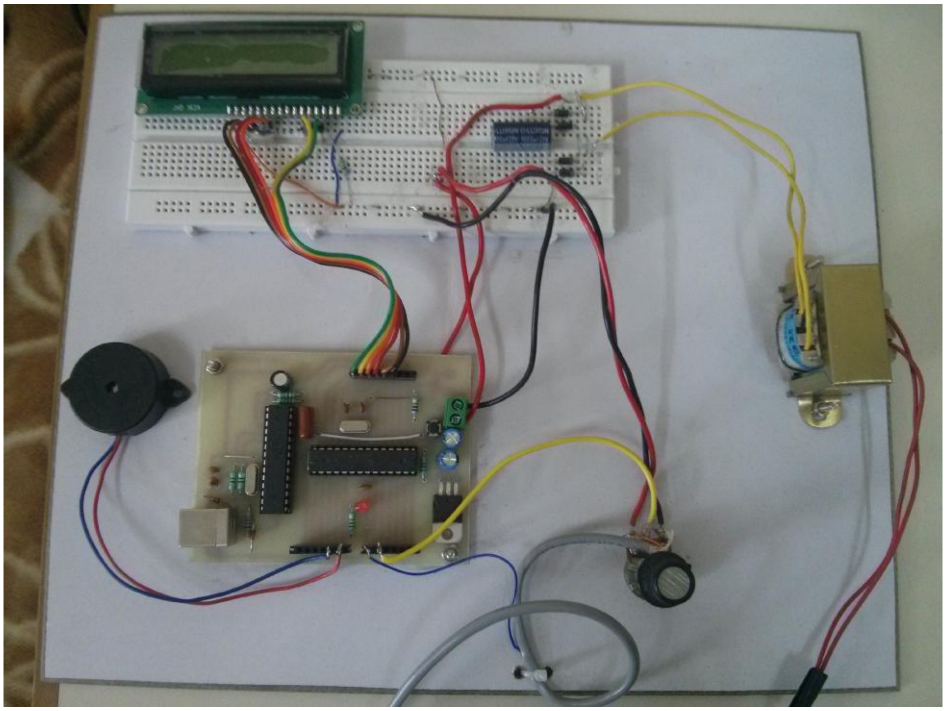


Figure 4

HARDWARE DETAILS

LM35 (temperature sensor)

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in ˚ Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±1⁄4˚C at room temperature and ±3⁄4˚C over a full −55 to +150˚C temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only 60 µA from its supply, it has very low self-heating, less than 0.1˚C in still air. The LM35 is rated to operate over a −55˚ to +150˚C temperature range, while the LM35C is rated for a −40˚ to +110˚C range (−10˚ with improved accuracy). The LM35 series is available packaged in hermetic TO-46 transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface mount small outline package and a plastic TO-220 package.

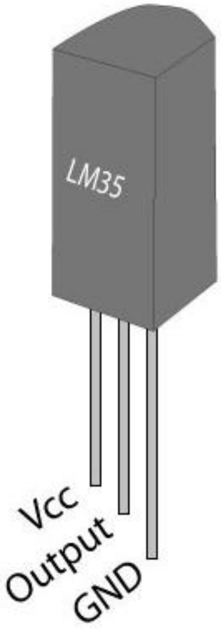


Figure 5

ARDUINO 2560

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 (datasheet). It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

The Arduino Mega can be powered via the USB connection or with an external power supply. The power source is selected automatically.

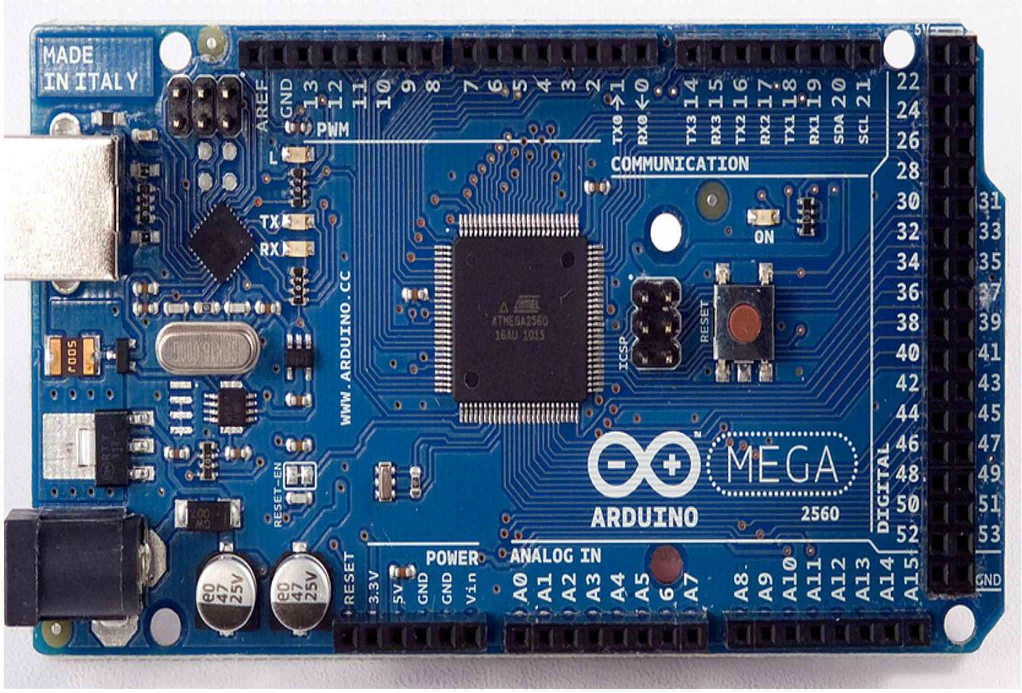


Figure 6

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library).

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega16U2(ATmega 8U2 on the revision 1 and revision 2 boards) on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2/ATmega16U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

ATMEGA 8A-PU

The Atmel AVR ATmega8A is a low-power CMOS 8-bit microcontroller based on the AVR RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega8A achieves throughputs approaching 1 MIPS per MHz, allowing the system designer to optimize power consumption versus processing speed.

The Atmel AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega8A provides the following features: 8K bytes of In-System Programmable Flash with Read-While- Write capabilities, 512 bytes of EEPROM, 1K byte of SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible Timer/Counters with compare modes, internal and external interrupts, a serial programmable USART, a byte oriented Two-wire Serial Interface, a 6-channel ADC (eight channels in TQFP and QFN/MLF packages) with 10-bit accuracy, a programmable Watchdog Timer with Internal Oscillator, an SPI serial port, and five software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next Interrupt or Hardware Reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except asynchronous timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low-power consumption.

The device is manufactured using Atmel’s high density non-volatile memory technology. The Flash Program memory can be reprogrammed In-System through an SPI serial interface, by a conventional non-volatile memory programmer, or by an On-chip boot program running on the AVR core. The boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash Section will continue to run while the Application Flash Section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-

Programmable Flash on a monolithic chip, the Atmel ATmega8A is a powerful microcontroller that provides a highly-flexible and cost-effective solution to many embedded control applications.

The Atmel AVR ATmega8A is supported with a full suite of program and system development tools, including C compilers, macro assemblers, program simulators and evaluation kits.

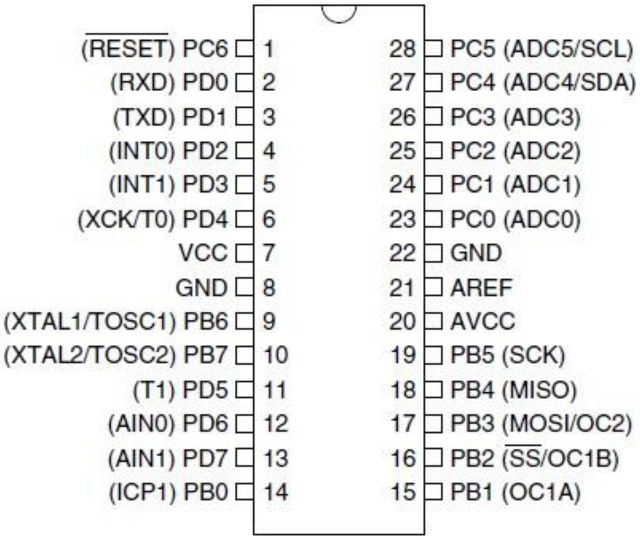


Figure 7

SOFTWARE DETAILS

MATLAB R2008a

MATLAB (matrix laboratory) is a multi-paradigm numerical computing environment and fourth-generation programming language. Developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java,Fortran and Python.

Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the MuPAD symbolic engine, allowing access to symbolic computing capabilities. An additional package, Simulink, adds graphical multi-domain simulation and Model-Based Design for dynamic and embedded systems.

MATLAB is a high-level language and interactive environment for numerical computation, visualization, and programming. Using MATLAB, we can analyze data, develop algorithms, and create models and applications. The language, tools, and built-in math functions enables us to explore multiple approaches and reach a solution faster than with spreadsheets or traditional programming languages, such as C/C++ or Java. We can use MATLAB for a range of applications, including signal processing and communications, image and video processing, control systems, test and measurement, computational finance, and computational biology. More than a million engineers and scientists in industry and academia use MATLAB, the language of technical computing.

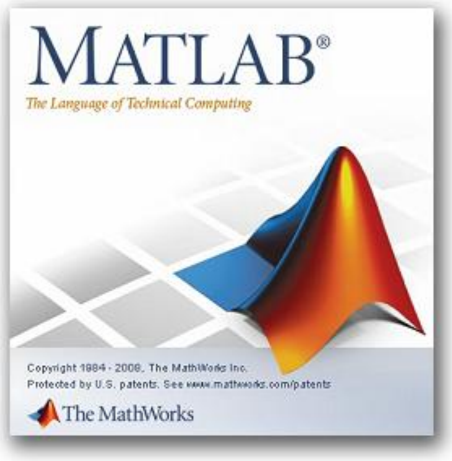

MATLAB®
The Language of Technical Computing
Copyright 1984 - 2008, The MathWorks Inc.
Protected by U.S. patents. See www.mathworks.com/patents
The MathWorks

Figure 8

ARDUINO 1.5.8

The Arduino integrated development environment (IDE) is a cross-platform application written in Java, and derives from the IDE for the Processing programming language and the Wiring projects. It is designed to introduce programming to artists and other newcomers unfamiliar with software development. It includes a code editor with features such as syntax highlighting, brace matching, and automatic indentation, and is also capable of compiling and uploading programs to the board with a single click. A program or code written for Arduino is called a "sketch".

Arduino programs are written in C or C++. The Arduino IDE comes with a software library called "Wiring" from the original Wiring project, which makes many common input/output operations much easier. The users need only to define two functions to make an executable cyclic executive program:

- setup(): a function run once at the start of a program that can initialize settings
- loop(): a function called repeatedly until the board powers off

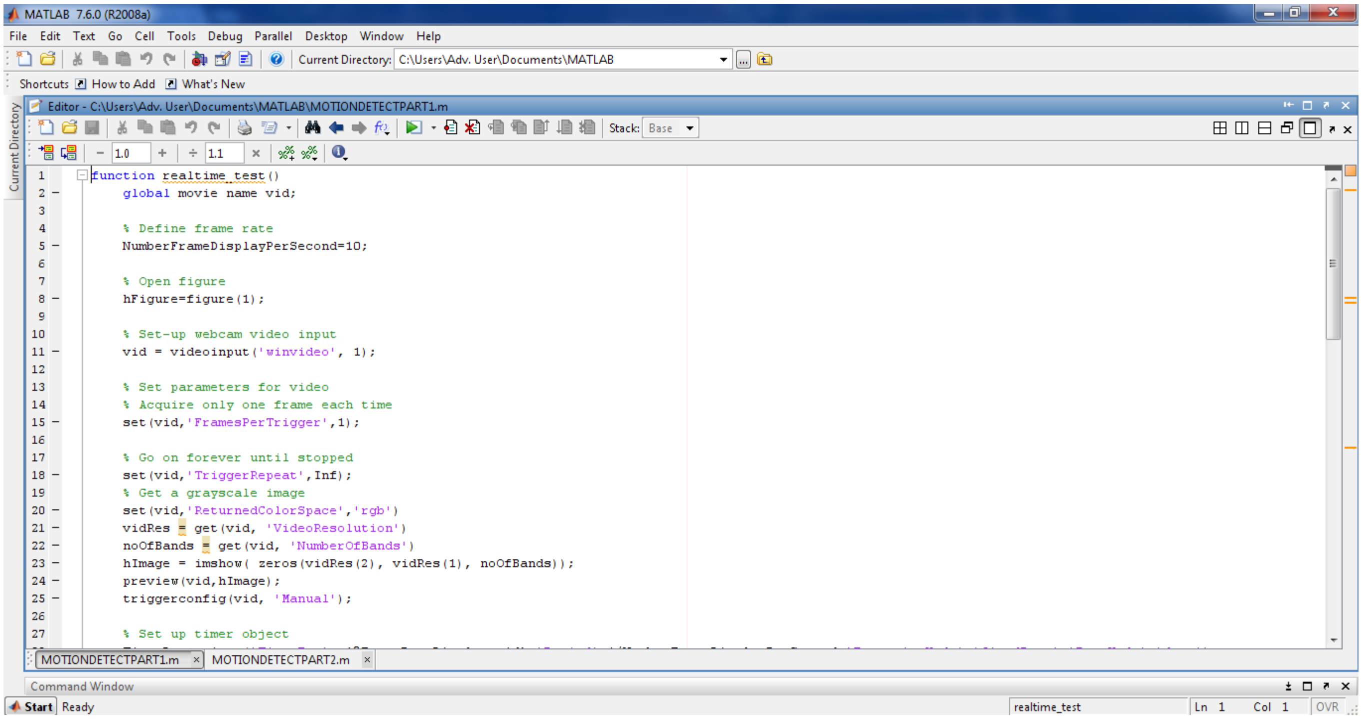
The Arduino IDE uses the GNU toolchain and AVR Libc to compile programs, and uses avrdude to upload programs to the board.

The open-source Arduino Software(IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

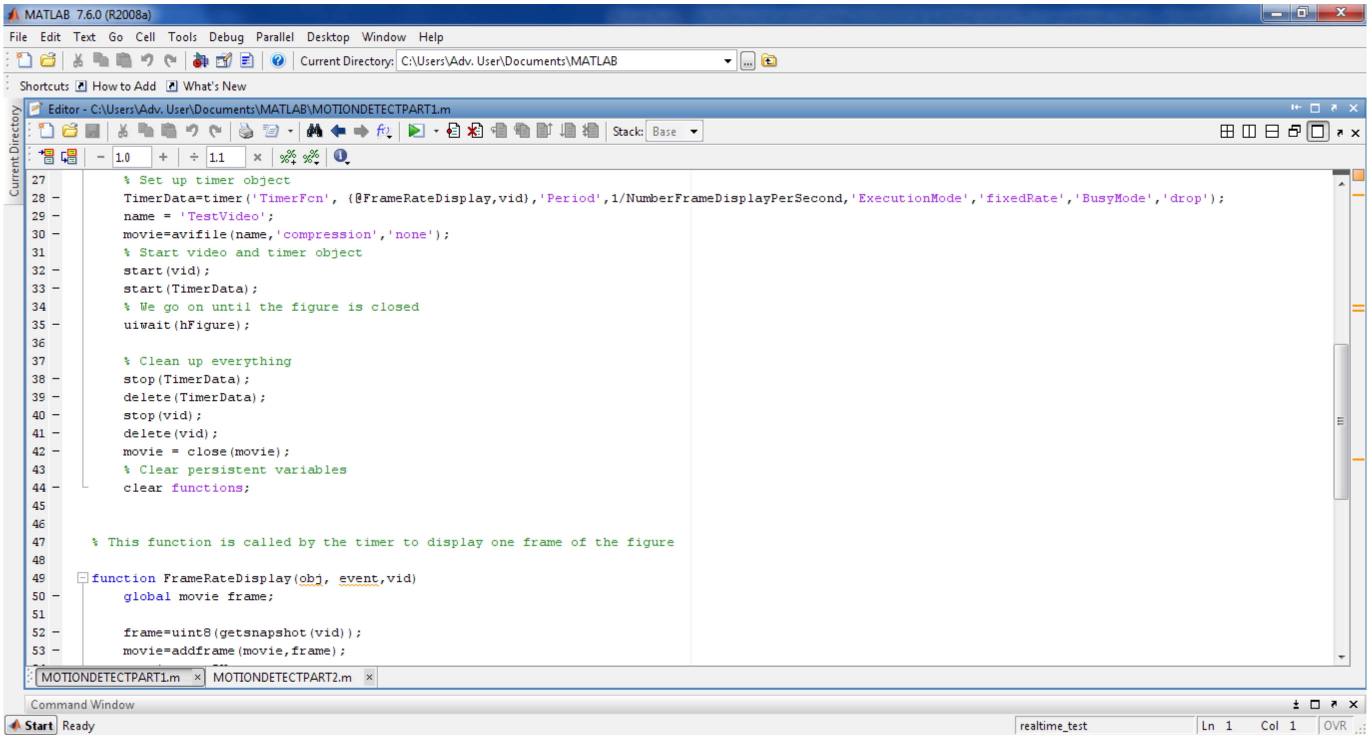
Figure 9

CODE

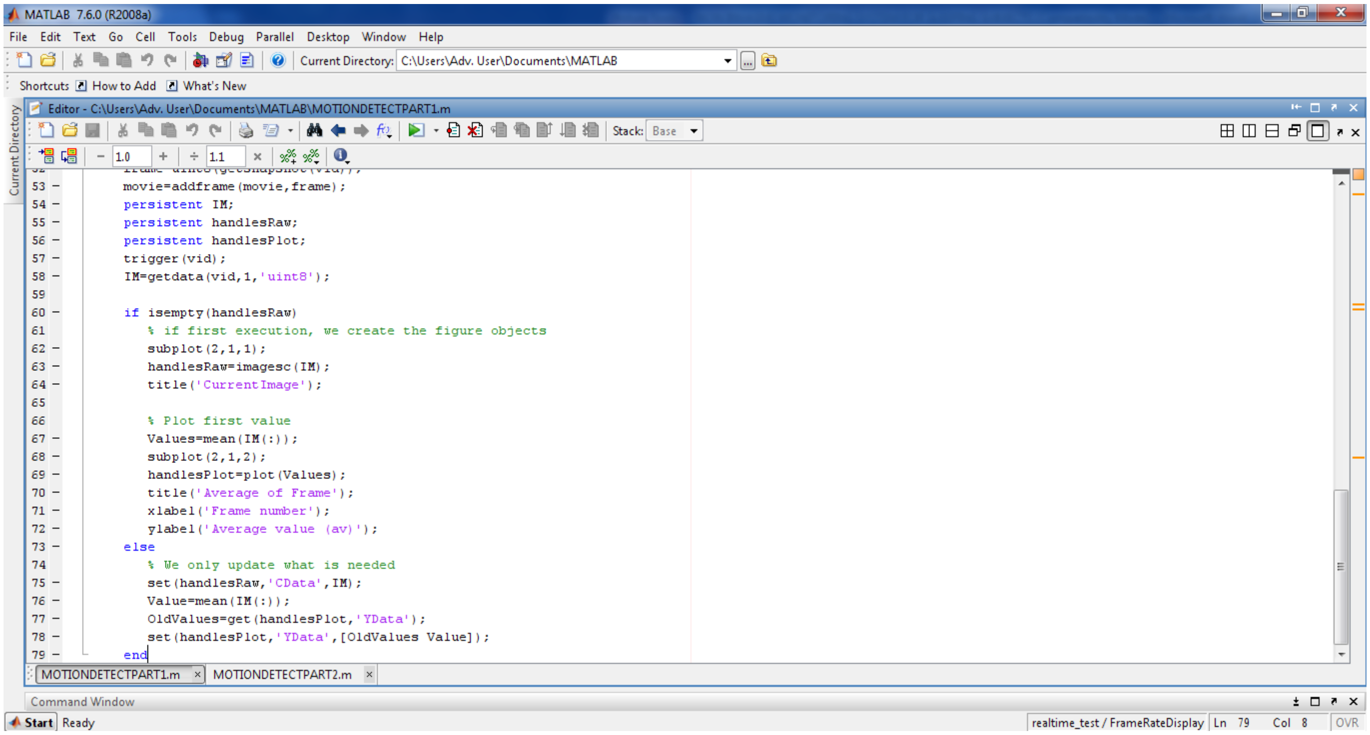
(i) MOTION DETECTION:


MATLAB 7.6.0 (R2008a)
File Edit Text Go Cell Tools Debug Parallel Desktop Window Help
Current Directory: C:\Users\Adv. User\Documents\MATLAB
Shortcuts How to Add What's New
Editor - C:\Users\Adv. User\Documents\MATLAB\MOTIONDETECTPART1.m
Stack: Base
MOTIONDETECTPART1.m
MOTIONDETECTPART2.m
Command Window
Start Ready
realtime_test Ln 1 Col 1 OVR

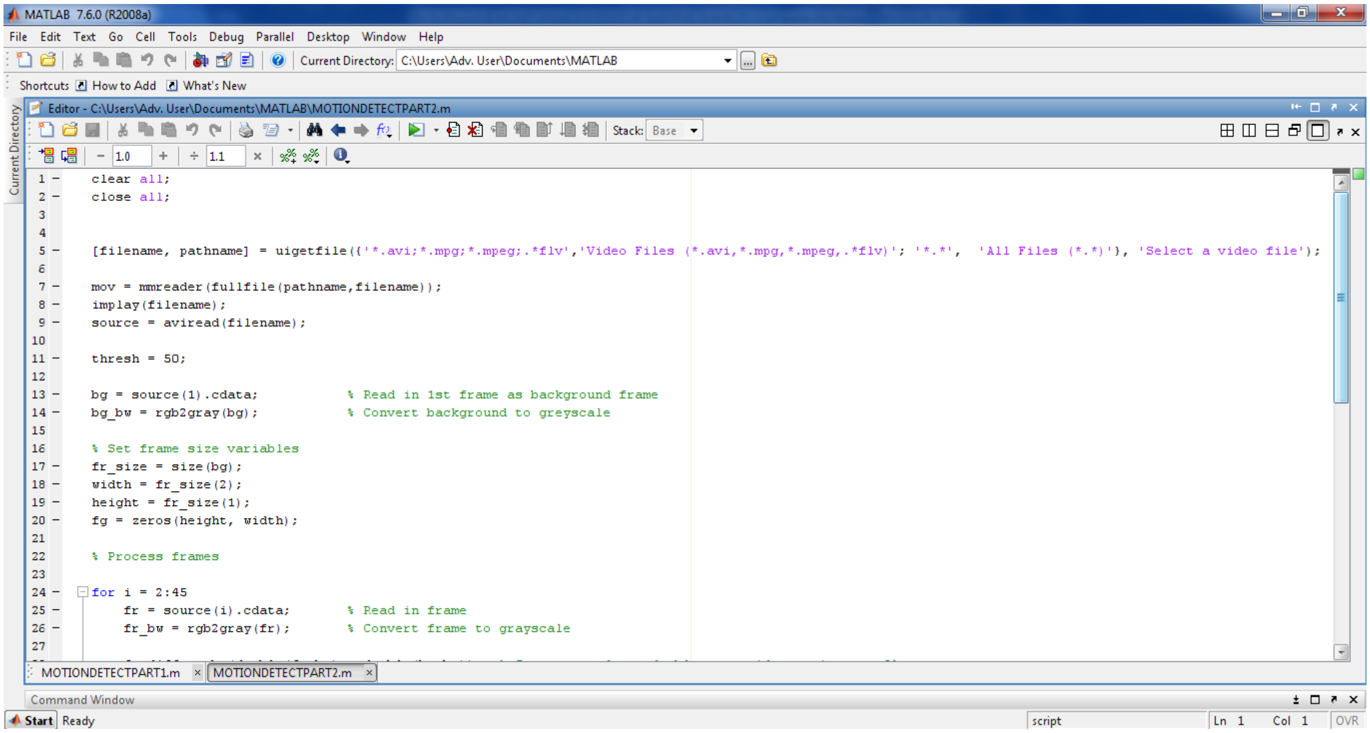
```
function realtime_test()
    global movie name vid;

    % Define frame rate
    NumberFrameDisplayPerSecond=10;

    % Open figure
    hFigure=figure(1);

    % Set-up webcam video input
    vid = videoinput('winvideo', 1);

    % Set parameters for video
    % Acquire only one frame each time
    set(vid,'FramesPerTrigger',1);

    % Go on forever until stopped
    set(vid,'TriggerRepeat',Inf);
    % Get a grayscale image
    set(vid,'ReturnedColorSpace','rgb')
    vidRes = get(vid, 'VideoResolution')
    noOfBands = get(vid, 'NumberOfBands')
    hImage = imshow( zeros(vidRes(2), vidRes(1), noOfBands));
    preview(vid,hImage);
    triggerconfig(vid, 'Manual');

    % Set up timer object
```


MATLAB 7.6.0 (R2008a)
File Edit Text Go Cell Tools Debug Parallel Desktop Window Help
Current Directory: C:\Users\Adv. User\Documents\MATLAB
Shortcuts How to Add What's New
Editor - C:\Users\Adv. User\Documents\MATLAB\MOTIONDETECTPART1.m
Stack: Base
MOTIONDETECTPART1.m
MOTIONDETECTPART2.m
Command Window
Start Ready
realtime_test Ln 1 Col 1 OVR

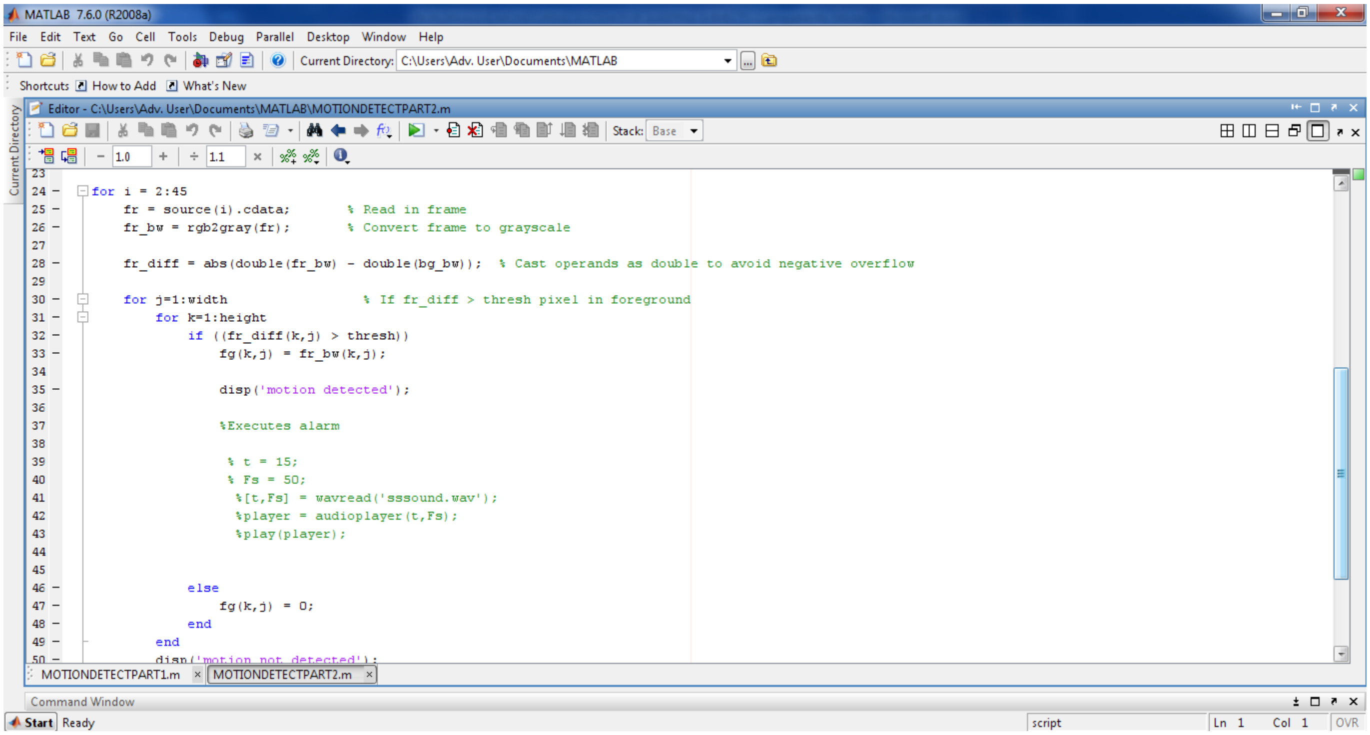
```
    % Set up timer object
    TimerData=timer('TimerFcn', {@FrameRateDisplay,vid},'Period',1/NumberFrameDisplayPerSecond,'ExecutionMode','fixedRate','BusyMode','drop');
    name = 'TestVideo';
    movie=avifile(name,'compression','none');
    % Start video and timer object
    start(vid);
    start(TimerData);
    % We go on until the figure is closed
    uiwait(hFigure);

    % Clean up everything
    stop(TimerData);
    delete(TimerData);
    stop(vid);
    delete(vid);
    movie = close(movie);
    % Clear persistent variables
    clear functions;


% This function is called by the timer to display one frame of the figure

function FrameRateDisplay(obj, event,vid)
    global movie frame;

    frame=uint8(getsnapshot(vid));
    movie=addframe(movie,frame);
```


MATLAB 7.6.0 (R2008a)
File Edit Text Go Cell Tools Debug Parallel Desktop Window Help
Current Directory: C:\Users\Adv. User\Documents\MATLAB
Shortcuts How to Add What's New
Editor - C:\Users\Adv. User\Documents\MATLAB\MOTIONDETECTPART1.m
Stack: Base
MOTIONDETECTPART1.m
MOTIONDETECTPART2.m
Command Window
Start Ready
realtime_test / FrameRateDisplay Ln 79 Col 8 OVR

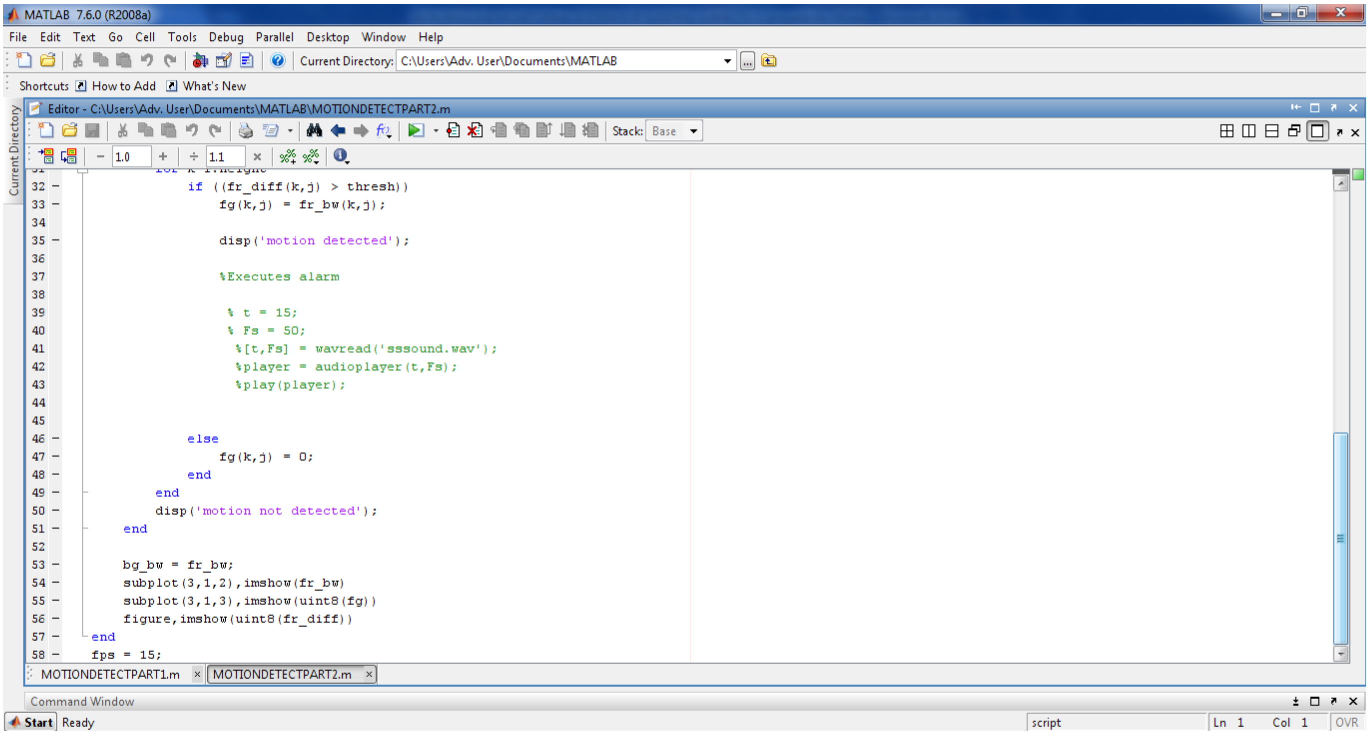
```
    frame=uint8(getsnapshot(vid));
    movie=addframe(movie,frame);
    persistent IM;
    persistent handlesRaw;
    persistent handlesPlot;
    trigger(vid);
    IM=getdata(vid,1,'uint8');

    if isempty(handlesRaw)
       % if first execution, we create the figure objects
       subplot(2,1,1);
       handlesRaw=imagesc(IM);
       title('CurrentImage');

       % Plot first value
       Values=mean(IM(:));
       subplot(2,1,2);
       handlesPlot=plot(Values);
       title('Average of Frame');
       xlabel('Frame number');
       ylabel('Average value (av)');
    else
       % We only update what is needed
       set(handlesRaw,'CData',IM);
       Value=mean(IM(:));
       OldValues=get(handlesPlot,'YData');
       set(handlesPlot,'YData',[OldValues Value]);
    end
```


MATLAB 7.6.0 (R2008a)
File Edit Text Go Cell Tools Debug Parallel Desktop Window Help
Current Directory: C:\Users\Adv. User\Documents\MATLAB
Shortcuts How to Add What's New
Editor - C:\Users\Adv. User\Documents\MATLAB\MOTIONDETECTPART2.m
Stack: Base
MOTIONDETECTPART1.m
MOTIONDETECTPART2.m
Command Window
Start Ready
script Ln 1 Col 1 OVR

```
clear all;
close all;


[filename, pathname] = uigetfile({'*.avi;*.mpg;*.mpeg;.*flv','Video Files (*.avi,*.mpg,*.mpeg,.*flv)'; '*.*',  'All Files (*.*)'}, 'Select a video file');

mov = mmreader(fullfile(pathname,filename));
implay(filename);
source = aviread(filename);

thresh = 50;

bg = source(1).cdata;           % Read in 1st frame as background frame
bg_bw = rgb2gray(bg);           % Convert background to greyscale

% Set frame size variables
fr_size = size(bg);
width = fr_size(2);
height = fr_size(1);
fg = zeros(height, width);

% Process frames

for i = 2:45
    fr = source(i).cdata;       % Read in frame
    fr_bw = rgb2gray(fr);       % Convert frame to grayscale

```

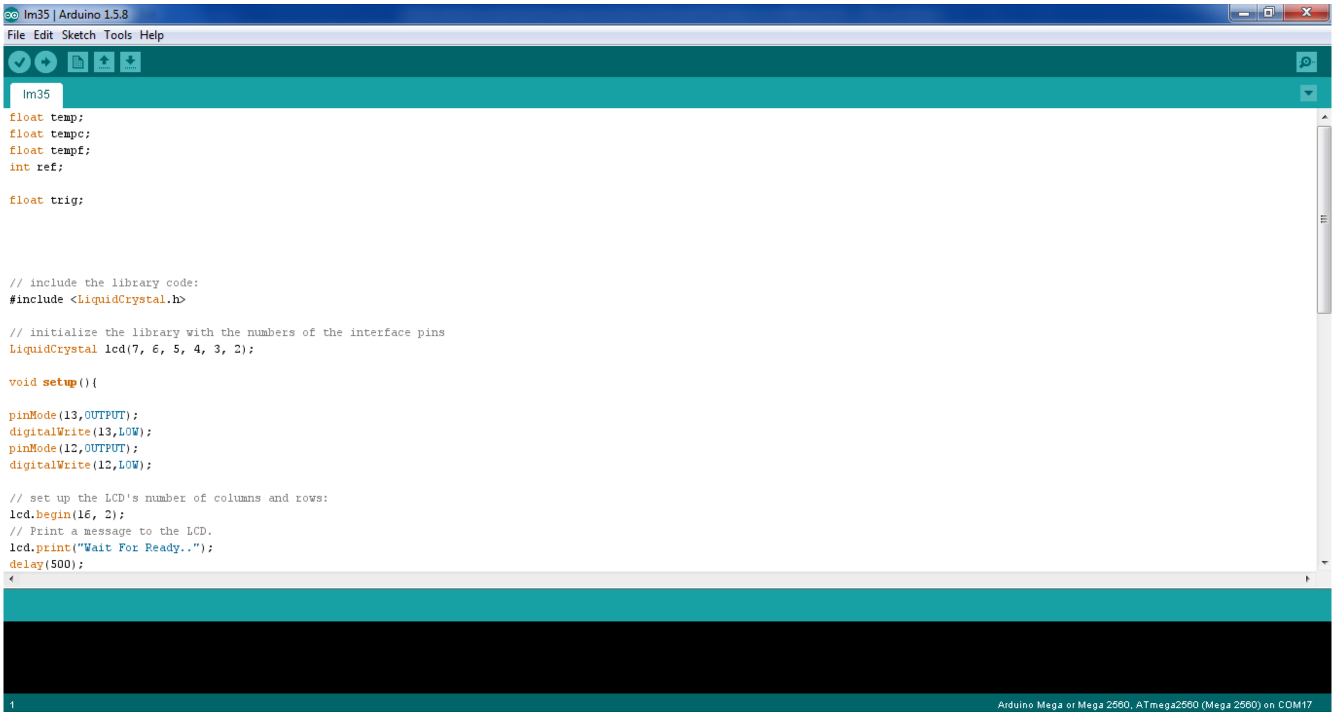

MATLAB 7.6.0 (R2008a)
File Edit Text Go Cell Tools Debug Parallel Desktop Window Help
Current Directory: C:\Users\Adv. User\Documents\MATLAB
Shortcuts How to Add What's New
Editor - C:\Users\Adv. User\Documents\MATLAB\MOTIONDETECTPART2.m
Stack: Base
1.0
1.1
MOTIONDETECTPART1.m
MOTIONDETECTPART2.m
Command Window
Start Ready
script
Ln 1 Col 1 OVR

```
for i = 2:45
    fr = source(i).cdata;       % Read in frame
    fr_bw = rgb2gray(fr);       % Convert frame to grayscale

    fr_diff = abs(double(fr_bw) - double(bg_bw));  % Cast operands as double to avoid negative overflow

    for j=1:width                 % If fr_diff > thresh pixel in foreground
        for k=1:height
            if ((fr_diff(k,j) > thresh))
                fg(k,j) = fr_bw(k,j);

                disp('motion detected');

                %Executes alarm

                 % t = 15;
                 % Fs = 50;
                  %[t,Fs] = wavread('sssound.wav');
                  %player = audioplayer(t,Fs);
                  %play(player);


            else
                fg(k,j) = 0;
            end
        end
        disp('motion not detected');
```


MATLAB 7.6.0 (R2008a)
File Edit Text Go Cell Tools Debug Parallel Desktop Window Help
Current Directory: C:\Users\Adv. User\Documents\MATLAB
Shortcuts How to Add What's New
Editor - C:\Users\Adv. User\Documents\MATLAB\MOTIONDETECTPART2.m
Stack: Base
1.0
1.1
MOTIONDETECTPART1.m
MOTIONDETECTPART2.m
Command Window
Start Ready
script
Ln 1 Col 1 OVR

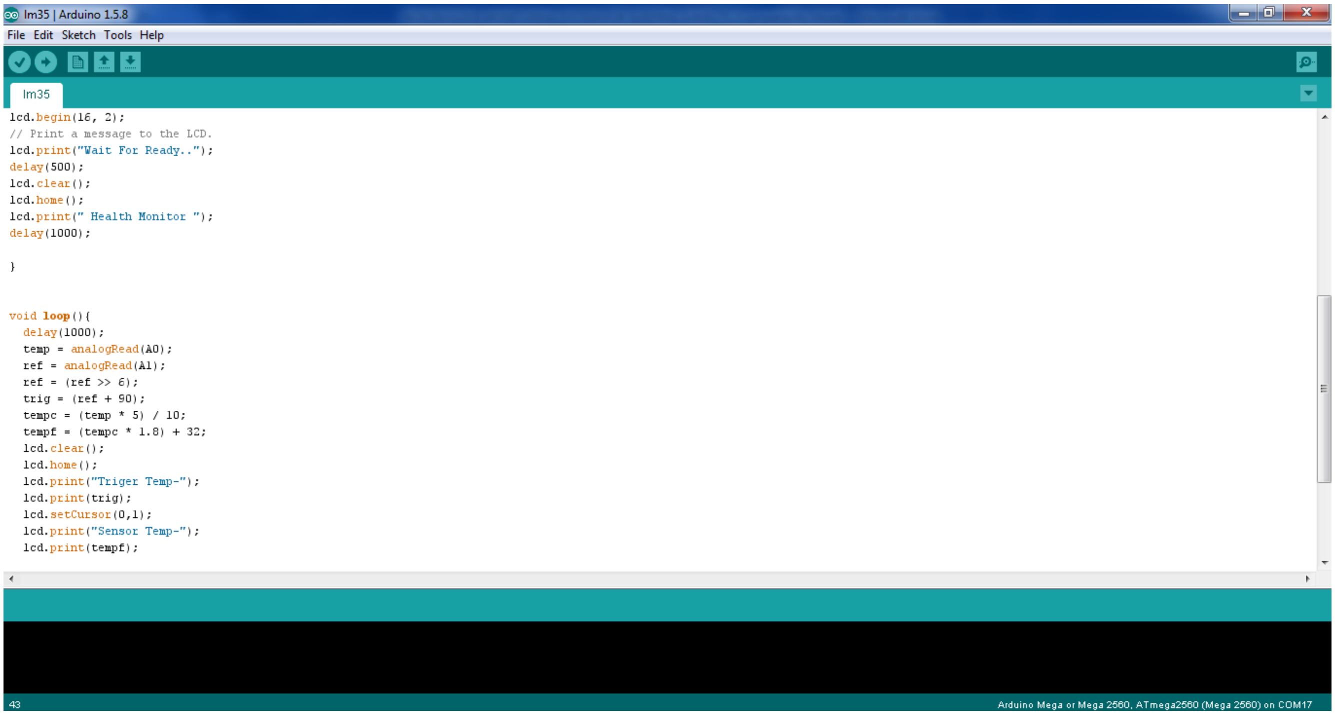
```
        for k=1:height
            if ((fr_diff(k,j) > thresh))
                fg(k,j) = fr_bw(k,j);

                disp('motion detected');

                %Executes alarm

                 % t = 15;
                 % Fs = 50;
                  %[t,Fs] = wavread('sssound.wav');
                  %player = audioplayer(t,Fs);
                  %play(player);


            else
                fg(k,j) = 0;
            end
        end
        disp('motion not detected');
    end

    bg_bw = fr_bw;
    subplot(3,1,2),imshow(fr_bw)
    subplot(3,1,3),imshow(uint8(fg))
    figure,imshow(uint8(fr_diff))
end
fps = 15;
```

(ii) TEMPERATURE MEASUREMENT:



```
float temp;
float tempc;
float tempf;
int ref;

float trig;




// include the library code:
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);

void setup(){

pinMode(13,OUTPUT);
digitalWrite(13,LOW);
pinMode(12,OUTPUT);
digitalWrite(12,LOW);

// set up the LCD's number of columns and rows:
lcd.begin(16, 2);
// Print a message to the LCD.
lcd.print("Wait For Ready..");
delay(500);
```

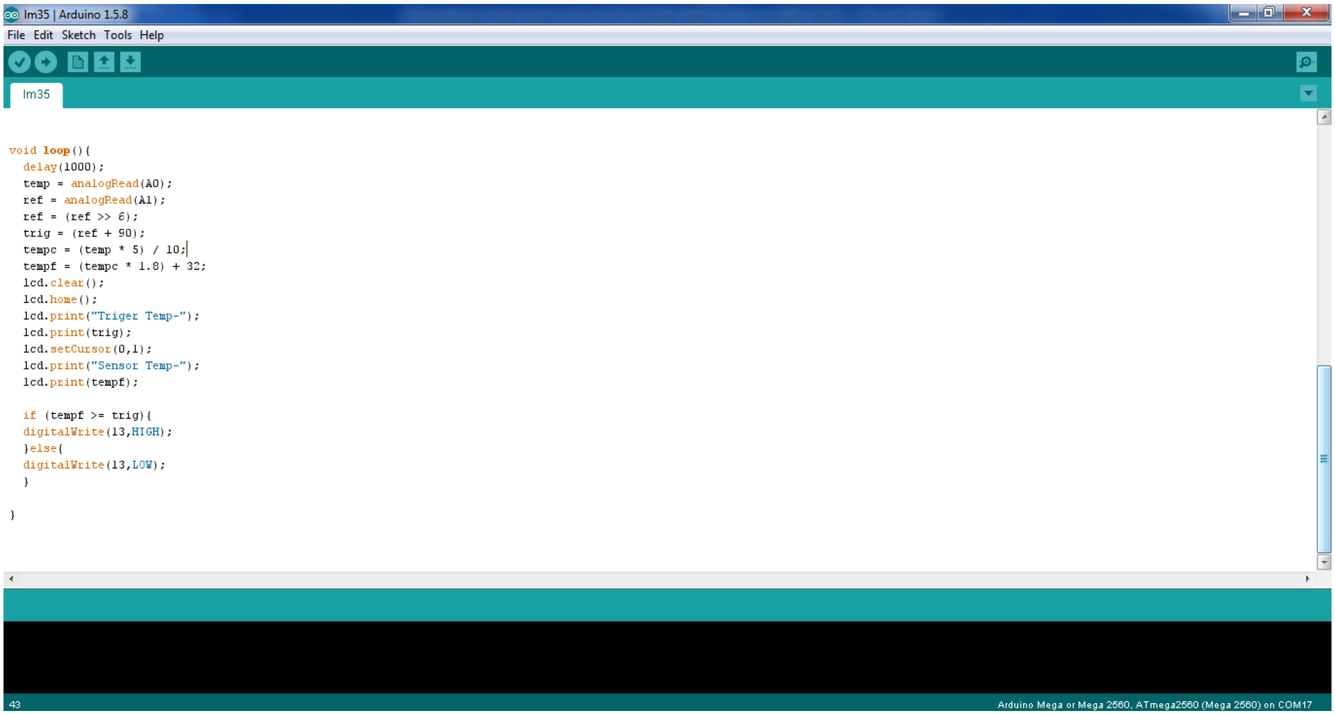
1
Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM17



lm35 | Arduino 1.5.8
File Edit Sketch Tools Help
lm35

```
lcd.begin(16, 2);
// Print a message to the LCD.
lcd.print("Wait For Ready..");
delay(500);
lcd.clear();
lcd.home();
lcd.print(" Health Monitor ");
delay(1000);

}


void loop(){
  delay(1000);
  temp = analogRead(A0);
  ref = analogRead(A1);
  ref = (ref >> 6);
  trig = (ref + 90);
  tempc = (temp * 5) / 10;
  tempf = (tempc * 1.8) + 32;
  lcd.clear();
  lcd.home();
  lcd.print("Triger Temp-");
  lcd.print(trig);
  lcd.setCursor(0,1);
  lcd.print("Sensor Temp-");
  lcd.print(tempf);
```

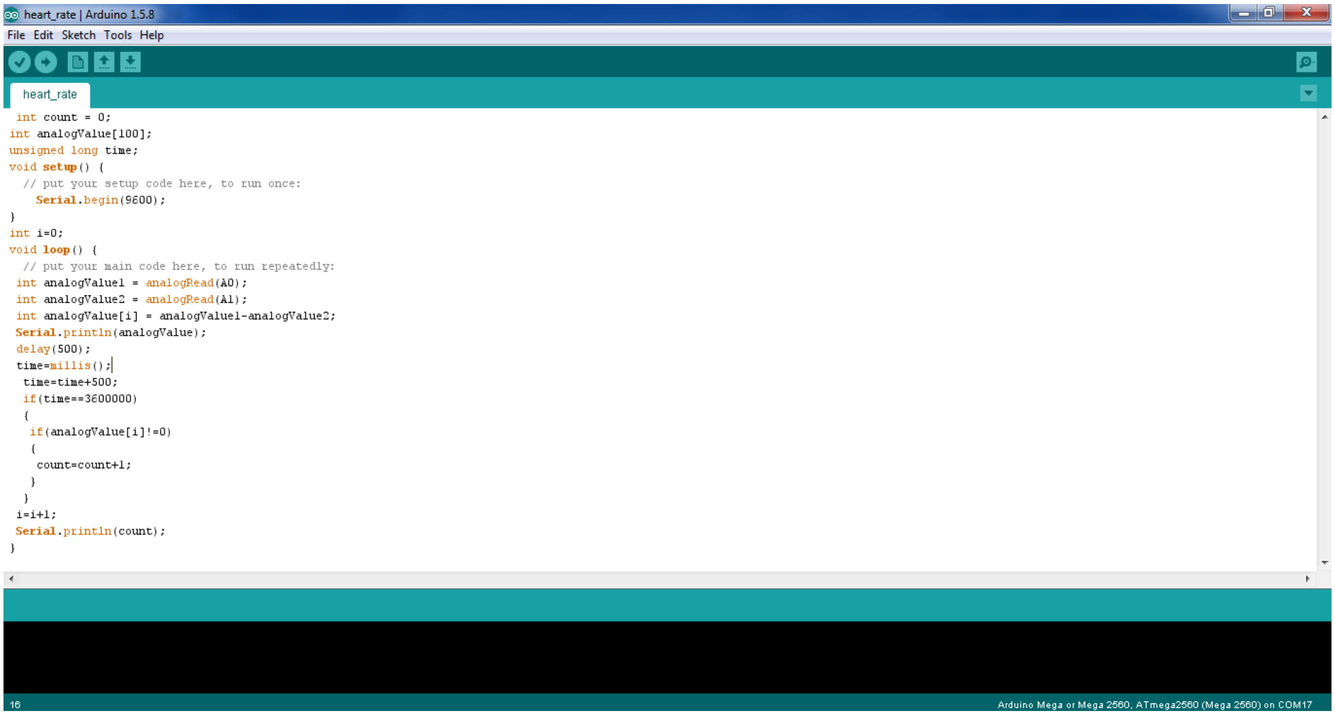
43
Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM17



```
void loop(){
  delay(1000);
  temp = analogRead(A0);
  ref = analogRead(A1);
  ref = (ref >> 6);
  trig = (ref + 90);
  tempc = (temp * 5) / 10;
  tempf = (tempc * 1.8) + 32;
  lcd.clear();
  lcd.home();
  lcd.print("Triger Temp-");
  lcd.print(trig);
  lcd.setCursor(0,1);
  lcd.print("Sensor Temp-");
  lcd.print(tempf);

  if (tempf >= trig){
  digitalWrite(13,HIGH);
  }else{
  digitalWrite(13,LOW);
  }

}
```

43   Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM17

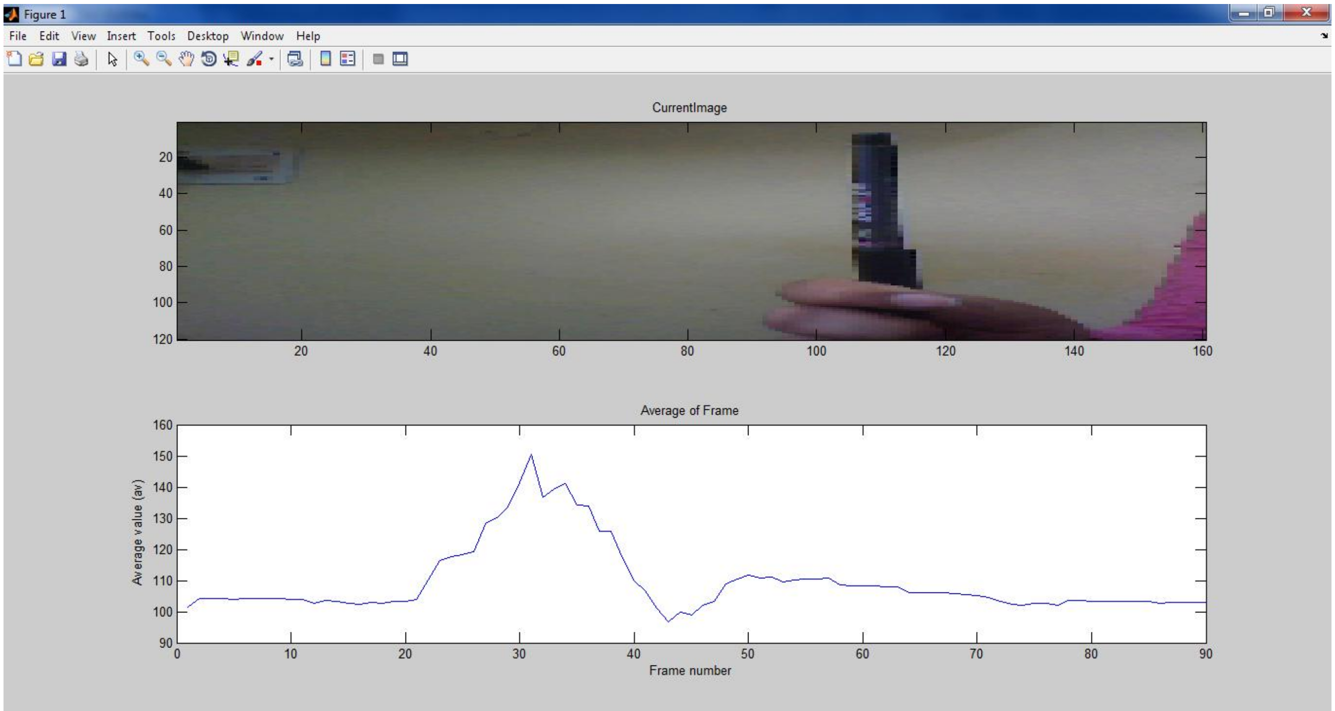
(iii)HEART RATE MEASUREMENT:

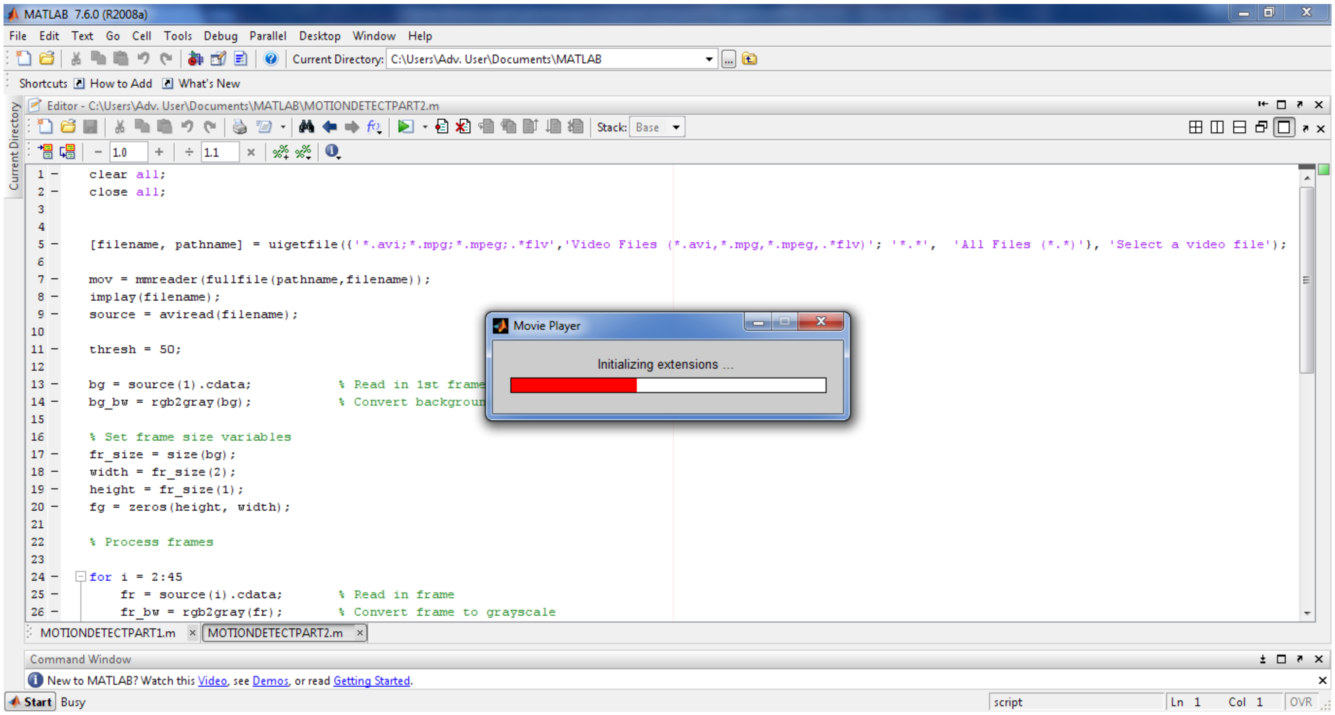

heart_rate | Arduino 1.5.8
File Edit Sketch Tools Help
heart_rate
16
Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM17

```
 int count = 0;
int analogValue[100];
unsigned long time;
void setup() {
  // put your setup code here, to run once:
    Serial.begin(9600);
}
int i=0;
void loop() {
  // put your main code here, to run repeatedly:
 int analogValue1 = analogRead(A0);
 int analogValue2 = analogRead(A1);
 int analogValue[i] = analogValue1-analogValue2;
 Serial.println(analogValue);
 delay(500);
 time=millis();
  time=time+500;
  if(time==3600000)
  {
   if(analogValue[i]!=0)
   {
    count=count+1;
   }
  }
 i=i+1;
 Serial.println(count);
}
```

RESULTS

(i) BODY MOTION DETECTION




MATLAB 7.6.0 (R2008a)
File Edit Text Go Cell Tools Debug Parallel Desktop Window Help
Current Directory: C:\Users\Adv. User\Documents\MATLAB
Shortcuts How to Add What's New
Editor - C:\Users\Adv. User\Documents\MATLAB\MOTIONDETECTPART2.m
Stack: Base
1.0
1.1
Current Directory
MOTIONDETECTPART1.m
MOTIONDETECTPART2.m
Command Window
Start
script
Ln 1
Col 1
OVR

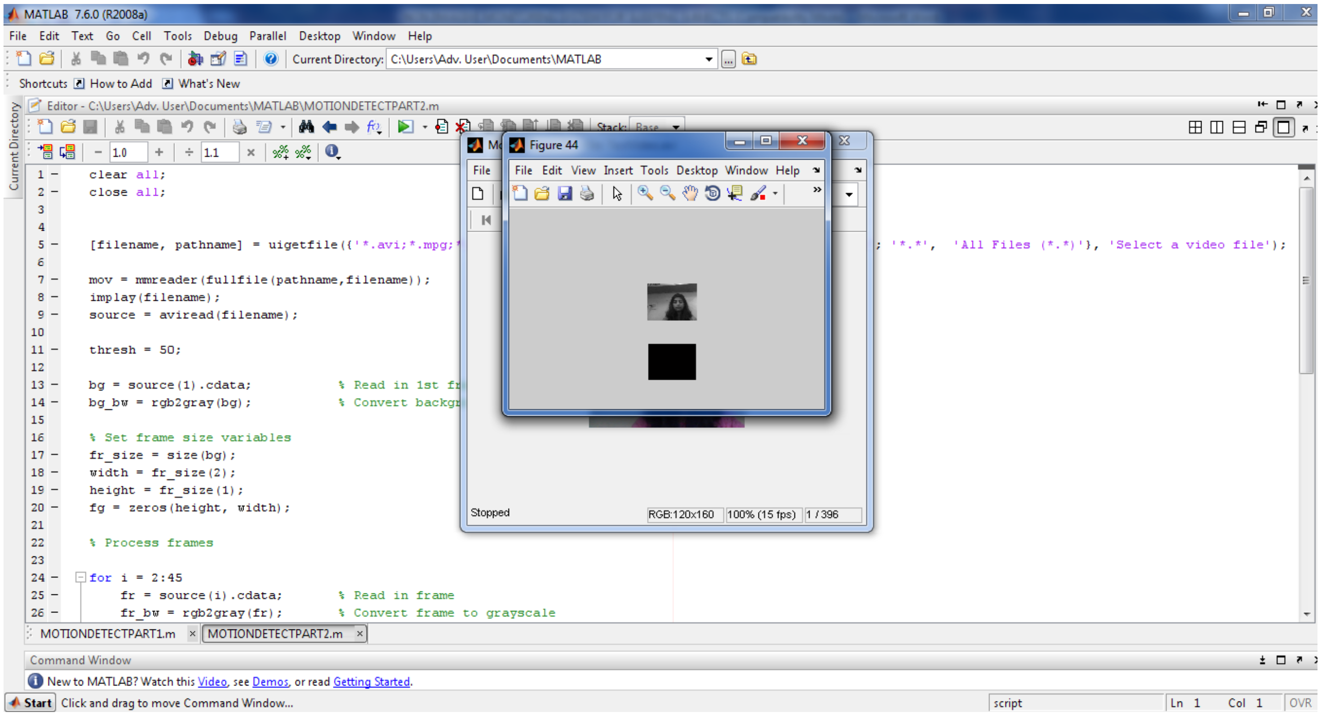
```
clear all;
close all;


[filename, pathname] = uigetfile({'*.avi;*.mpg;*.mpeg;.*flv','Video Files (*.avi,*.mpg,*.mpeg,.*flv)'; '*.*',  'All Files (*.*)'}, 'Select a video file');

mov = mmreader(fullfile(pathname,filename));
implay(filename);
source = aviread(filename);

thresh = 50;

bg = source(1).cdata;            % Read in 1st frame
bg_bw = rgb2gray(bg);            % Convert backgroun

% Set frame size variables
fr_size = size(bg);
width = fr_size(2);
height = fr_size(1);
fg = zeros(height, width);

% Process frames

for i = 2:45
    fr = source(i).cdata;        % Read in frame
    fr_bw = rgb2gray(fr);        % Convert frame to grayscale
```

Movie Player

Initializing extensions ...

New to MATLAB? Watch this Video, see Demos, or read Getting Started.

Busy



MATLAB 7.6.0 (R2008a)
File Edit Text Go Cell Tools Debug Parallel Desktop Window Help
Current Directory: C:\Users\Adv. User\Documents\MATLAB
Shortcuts How to Add What's New
Current Directory
Editor - C:\Users\Adv. User\Documents\MATLAB\MOTIONDETECTPART2.m
Stack: Base
1.0
1.1
Figure 44
File Edit View Insert Tools Desktop Window Help
MOTIONDETECTPART1.m
MOTIONDETECTPART2.m
Command Window
Start
script
Ln 1
Col 1
OVR

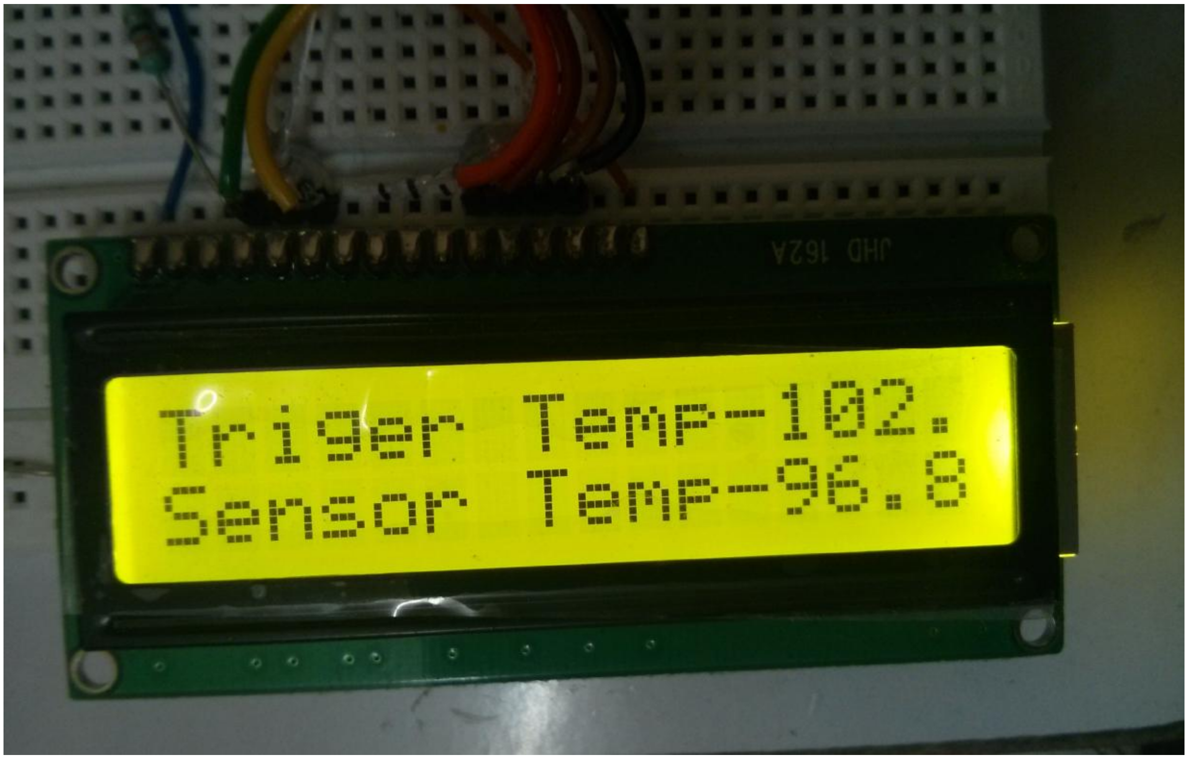
```
clear all;
close all;


[filename, pathname] = uigetfile({'*.avi;*.mpg;'                    ; '*.*',  'All Files (*.*)'}, 'Select a video file');

mov = mmreader(fullfile(pathname,filename));
implay(filename);
source = aviread(filename);

thresh = 50;

bg = source(1).cdata;            % Read in 1st fr
bg_bw = rgb2gray(bg);            % Convert backgr

% Set frame size variables
fr_size = size(bg);
width = fr_size(2);
height = fr_size(1);
fg = zeros(height, width);

% Process frames

for i = 2:45
    fr = source(i).cdata;        % Read in frame
    fr_bw = rgb2gray(fr);        % Convert frame to grayscale
```

```
Stopped    RGB:120x160    100% (15 fps)    1 / 396
```

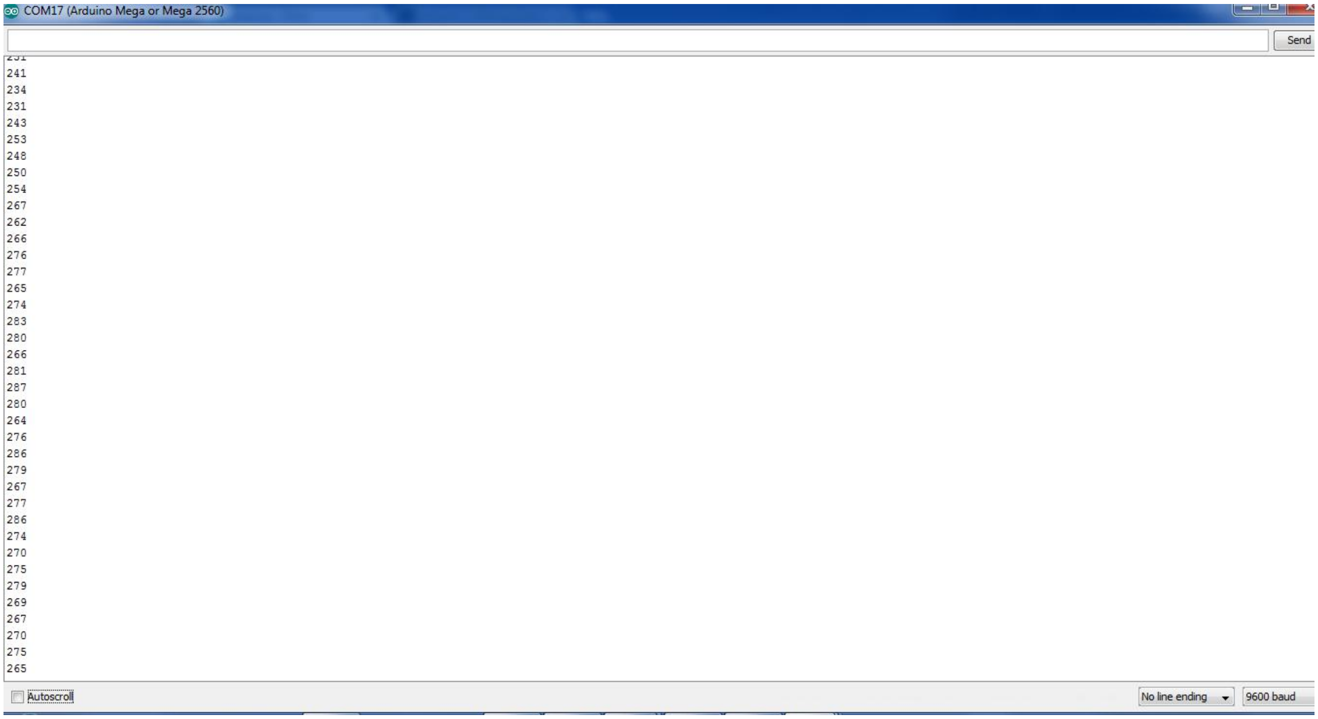
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

Click and drag to move Command Window...

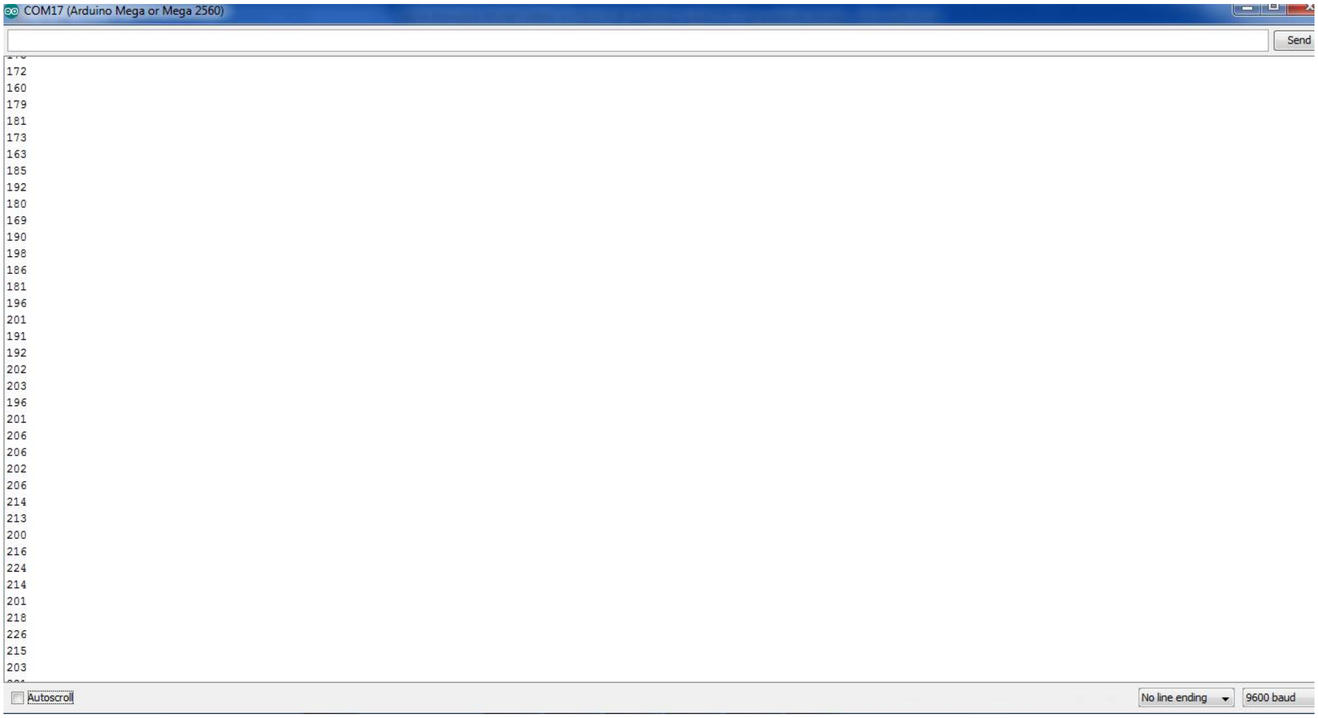
(ii)TEMPERATURE MEASUREMENT



(iii)HEART RATE MEASUREMENT



```
241
234
231
243
253
248
250
254
267
262
266
276
277
265
274
283
280
266
281
287
280
264
276
286
279
267
277
286
274
270
275
279
269
267
270
275
265
```



```
172
160
179
181
173
163
185
192
180
169
190
198
186
181
196
201
191
192
202
203
196
201
206
206
202
206
214
213
200
216
224
214
201
218
226
215
203
```

CONCLUSION

Sleep is an important part of our lives which affects many life factors such as memory, learning, metabolism and the immune system. Researchers have found correlations between sleep and several diseases such as Chronic Obstructive Pulmonary disease, Chronic Heart Failure, Alzheimer’s disease, etc. However, sleep data is mainly recorded and diagnosed in sleep labs or in hospitals for some critical cases with high costs.

Approximately a third of the population suffers from difficulty falling asleep, frequent waking, poor quality of sleep, and a variety of sleep-related breathing problems. For older adults, there are concerns that medical emergencies that occur during sleep could go unnoticed. Unfortunately, most current devices for monitoring sleep are uncomfortable and used primarily for making medical diagnoses. However, many health benefits could result from an unobtrusive way to monitor sleep in a home environment.

The monitoring of sleep patterns is of major importance for various reasons such as the detection and treatment of sleep disorders, the assessment of the effect of different medical conditions or medications on the sleep quality, and the assessment of mortality risks associated with sleeping patterns in adults and children. Sleep monitoring by itself is a difficult problem due to both privacy and technical considerations.

Sleep has profound effects on the physical and mental well-being of an individual. The National Institutes of Health (NIH) Sleep Disorder Research Plan gives particular emphasis to non-invasive sleep monitoring methods. Older adults experience sleep fragmentation due to sleep disorders. Unobtrusive non-contact monitoring can be the only realistic solution for long term home-based sleep monitoring. The demand for a low-cost and non-invasive sleep monitoring system for in-home use is more than before due to an increasingly stressful life style. Cost and complexity of current sensor elements hinder the development of low-cost sleep monitoring devices for in-home use.

In this project, I worked on analysis of sleep patterns using non-invasive sensors, microcontroller and matlab software. Our experimental results on real user datasets show that the task of analyzing sleep patterns with the intent to detect symptoms related to sleep disorders can be successfully achieved.

Although the available dataset was relatively small, the classification accuracy results are promising and show that the proposed tools and methods could be used in the future for the detection of sleep disorders and other related diseases affecting sleep quality. To this end, further experimentation with bigger datasets and improved fusion methodology would be of high interest.

FUTURE WORK

While sleep research in a clinical setting has been going on for decades, it is only through the development of modern signal processing methods that long-term, non-intrusive monitoring of sleep has been made possible. This opens new possibilities for research, such as the so far little studied interaction of two sleepers sharing a bed and e.g. the development of their sleep patterns as the subjects age and their relationship matures.

In the future, I plan to apply the system to large-scale clinical tests and I believe that it will be possible to associate our findings with pathological cases as well as depression. The big challenge is the diagnosis of diseases by recognizing the sleep patterns, which may lead to more focused medical treatments. The more focused treatments are expected to enhance the quality of life for millions of patients suffering from sleep disorders.

REFERENCES

[1] Affanni, Jorge M., Claudio O. Cervino, and Hernan J. Aldana Marcos. 2001. Absence of penile erections during paradoxical sleep. Peculiar penile events during wakefulness and slow wave sleep in the armadillo. Journal of Sleep Research 10(3), 219–228.

[2] Corthout, J., S. Van Huffel, M. O. Mendez, A. M. Bianchi, T. Penzel, and S. Cerutti. 2008. Automatic screening of Obstructive Sleep Apnea from the ECG based on Empirical Mode Decomposition and Wavelet Analysis. In Proceedings of the 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, volume 2008, pages 3608–3611. Vancouver BC. ISBN 9781424418152.

[3] Connor, Jennie, Robyn Norton, Shanthi Ameratunga, Elizabeth Robinson, Ian Civil, Roger Dunn, John Bailey, and Rod Jackson. 2002. Driver sleepiness and risk of serious injury to car occupants: population based case control study. British Medical Journal 324(7346), 1125–1129.

[4] A. Yadollahi and Z.Mousavi , “Acoustic Obstructive sleep apnea detection” 31st Annual International Conference of the IEEE EMBS Minneapolis, Minnesota, USA, Sep 2009.

[5] Miller MA, Cappuccio FP. “Inflammation, sleep, obesity and cardiovascular disease”. Curr Vasc Pharmacol. 2007;5:93–102.

[6]American Academy of Sleep Medicine (2001) International classification of sleep disorders, revised: diagnostic and coding manual. American Academy of Sleep Medicine, Illinois.

[7] Sung J, Ponce C, Selman B, Saxena A (2012) Unstructured human activity detection from RGBD images. In: Proceedings of the International Conference on Robotics and Automation (ICRA). IEEE, pp 842–849.

[8] Susan R, Sonia AI (2010) Sleep disorders in the elderly. Indian J Med Res 131:302–310.

[9] http://www.arduino.cc/

[10] http://in.mathworks.com/

[11] http://www.sleepdex.org/stages.htm