# VALUE ADDED COURSES CONDUCTED BY THE INSTITUTION

| Course | Year | No. of Time Offered | Duration | No. Students Attended |
|---|---|---|---|---|
| CRT Programme | 22-05-2023 to 27.05.2023 | 2 | 1 Week | 280 |
| CRT Programme | 29-05-2023 to 03.06.2023 | 2 | 1 Week | 190 |
| CRT Programme | 23.11.2021 to 26.11.2021 | 1 | 1 Week | 320 |
| Python Programming | 12.12.2022 to 18.12.2022 | 1 | 2 Week | 160 |
| Java Programming | 21.03.2022 to 26.03.2022 | 1 | 1 Week | 240 |
| Implementation Of Image Processing Concepts For Real time Applications Using MATLAB | 2021 | 1 | 6 Weeks | 25 |
| Sketch with Arduino | 2022 | 1 | 4 Weeks | 30 |
| VBB enabled Projects using Arduino | 2021 | 1 | 3 Weeks | 25 |
| Image Retrieval Process using MATLAB | 2021 | 1 | 3 Weeks | 31 |

SRI INDU COLLEGE OF ENGG. & TECH.

PRINCIPAL
Sri Indu College of Engineering and Technology
(Vill): SHERIGUDA-501 510,
Ibrahimpatnam(M), R.R.Dist.

# Sri Indu College of Engineering & Technology

**An Autonomous Institution under UGC**

Recognized under 2(f) and 12(B) of UGC Act 1956
NBA & NAAC Accredited, Approved by AICTE and
Permanently affiliated to JNT University, Hyderabad.

Estd.: 2001 | An ISO 9001 : 2000 CERTIFIED COLLEGE

---

## Sri Indu College of Engineering & Technology

UGC Autonomous Institution
Recognized Under 2(f) & 12(B) of UGC Act, 1956,
Accredited by NAAC & NBA approved by AICTE,
Permanently Affiliated to JNTUH

# 6 DAY CRT TRAINING

**As a Value Added Course**

22-05-2023 TO 27-05-2023 Online from 10:00 AM to 3:30PM

**FOR THIRD YEAR STUDENTS**

Note : It's mandatory for everyone to attend the classes in formal attire.
There should be no breakage of power and network.
Make your Zoom ID's as your Roll No.
Course duration 28 Hours.
Contact Department Co-ordinators for further details.
Attendance is mandatory

---

## Sri Indu College of Engineering & Technology

UGC Autonomous Institution
Recognized Under 2(f) & 12(B) of UGC Act, 1956,
Accredited by NAAC & NBA approved by AICTE,
Permanently Affiliated to JNTUH

# 6 DAY CRT TRAINING

**As a value added course**

29-05-2023 TO 03-06-2023 from 10:00 AM to 04:00PM

**FOR THIRD YEAR STUDENTS**

Note : It's mandatory for everyone to attend the classes in formal attire.
There should be no breakage of power and network.
Course duration 30 Hours.
Contact Department Co-ordinators for further details.
Attendance is mandatory

# Sri Indu College of Engineering & Technology

**An Autonomous Institution under UGC**

Recognized under 2(f) and 12(B) of UGC Act 1956
NBA & NAAC Accredited, Approved by AICTE and
Permanently affiliated to JNT University, Hyderabad.

Estd.: 2001
ISO 9001 : 2000 CERTIFIED COLLEGE

---

## Sri Indu College of Engineering & Technology

**UGC Autonomous Institution**

Recognized Under 2(f) & 12(B) of UGC Act. 1956,
Accredited by NAAC & NBA approved by AICTE,
Permanently Affiliated to JNTUH

# 4 DAY CRT TRAINING

**As a value added course**

*23-11-2021 TO 26-11-2021 from 10:00 AM to 04:00PM*

**FOR FINAL YEAR STUDENTS**

**Note : It's mandatory for everyone to attend the classes in formal attire.**
**There should be no breakage of power and network.**
**Course duration 24 Hours.**
**Contact Department Co-ordinators for further details.**
**Attendance is mandatory**

SRI INDU COLLEGE OF ENGG & TECH.

PRINCIPAL
Sri Indu College of Engineering and Technology
(Vill): SHERIGUDA-501 510,
Ibrahimpatnam(M), R.R.Dist.

# Sri Indu College of Engineering & Technology

**An Autonomous Institution under UGC**

Recognized under 2(f) and 12(B) of UGC Act 1956
NBA & NAAC Accredited, Approved by AICTE and
Permanently affiliated to JNT University, Hyderabad.

Estd.: 2001

ISO 9001 : 2000 CERTIFIED COLLEGE

---

## Sri Indu College of Engineering & Technology

**UGC Autonomous Institution**

Recognized Under 2(f) & 12(B) of UGC Act. 1956,
Accredited by NAAC & NBA approved by AICTE,
Permanently Affiliated to JNTUH

# Value Added Course

## on Python Programming

**12-12-2022 TO 18-12-2022 from 10:00 AM to 04:00PM**

**FOR SECOND YEAR STUDENTS**

**Note : It's mandatory for everyone to attend the classes in formal attire.**
**There should be no breakage of power and network.**
**Course duration 32 Hours.**
**Contact Department Co-ordinators for further details.**
**Attendance is mandatory**

---

Estd.2001

**Sri Indu College of Engineering & Technology**
**UGC Autonomous Institution**
Recognized under 2(f) & 12(B) of UGC Act 1956
NAAC, Approved by AICTE &
Permanently Affiliated to JNTUH

NATIONAL ASSESSMENT AND ACCREDITATION COUNCIL

INSTITUTION'S INNOVATION COUNCIL
(Ministry of HRD Initiative)

## CERTIFICATE OF APPRECIATION

*This is to certify that*

**A.Yashwanth reddy of CSE (Cyber Security)**

*has successfully completed the Two Week Value Added Course on* ***PYTHON PROGRAMMING*** *organized by the department of CSE of Sri Indu College of Engineering and Technology under IQAC held on 12.12.2022 to 18.12.2022.*

| Dr. K. Sampath | Prof.G.Uma Maheshwari | Prof. B.Surekha | Prof.K.Ashok Babu | Dr.G.Suresh |
|---|---|---|---|---|
| HoD/IOT & DS | HoD/AI ML & CS | HoD/CSIT & IT | Convener | Principal |

# Sri Indu College of Engineering & Technology

**An Autonomous Institution under UGC**

Recognized under 2(f) and 12(B) of UGC Act 1956
NBA & NAAC Accredited, Approved by AICTE and
Permanently affiliated to JNT University, Hyderabad.

Estd.: 2001
An ISO 9001 : 2000 CERTIFIED COLLEGE

Sri Indu College of Engineering & Technology
UGC Autonomous Institution
Recognized under 2(f) & 12(B) of UGC Act 1956
NAAC, Approved by AICTE & Permanently Affiliated to JNTUH
Estd.2001

NAAC
NATIONAL ASSESSMENT AND ACCREDITATION COUNCIL

NBA

## CERTIFICATE OF APPRECIATION

INSTITUTION'S INNOVATION COUNCIL
(Ministry of HRD Initiative)

*This is to certify that*

**T. Sri charan of CSE (Cyber Security)**

*has successfully completed the Two Week Value Added Course on* ***PYTHON PROGRAMMING*** ***organized by the department of CSE*** *of Sri Indu College of Engineering and Technology under IQAC held on 12.12.2022 to 18.12.2022.*

| Dr. K.Sampath | Prof.G.Uma Maheshwari | Prof. B.Surekha | Prof.K.Ashok Babu | Dr.G.Suresh |
|---|---|---|---|---|
| HoDMOT & DS | HoD/AI ML & CS | HoD/CSIT & IT | Convenor | Principal |

SRI INDU COLLEGE OF ENGG. & TECH.

PRINCIPAL
Sri Indu College of Engineering and Technology
(Vill): SHERIGUDA-501 510,
Ibrahimpatnam(M), R.R.Dist.

# Sri Indu College of Engineering & Technology

**An Autonomous Institution under UGC**

Recognized under 2(f) and 12(B) of UGC Act 1956
NBA & NAAC Accredited, Approved by AICTE and
Permanently affiliated to JNT University, Hyderabad.

Estd.: 2001

ISO 9001 : 2000 CERTIFIED COLLEGE

---

## Sri Indu

College of Engineering & Technology

**UGC Autonomous Institution**

Recognized Under 2(f) & 12(B) of UGC Act. 1956,
Accredited by NAAC & NBA approved by AICTE,
Permanently Affiliated to JNTUH

NAAC NATIONAL ASSESSMENT AND ACCREDITATION COUNCIL

NBA

# Value Added Course

## on Java Programming

21-03-2022 TO 26-03-2022 from 10:00 AM to 04:00PM

**FOR THIRD YEAR STUDENTS**

**Note : It's mandatory for everyone to attend the classes in formal attire.**
**There should be no breakage of power and network.**
**Course duration 28 Hours.**
**Contact Department Co-ordinators for further details.**
**Attendance is mandatory**

---

Estd.2001

## Sri Indu

College of Engineering &
UGC Autonomous Institu
Recognized under 2(f) & 12(B) of U 19
NAAC, Approved by AICTE &
Permanently Affiliated to JNTUH

NAAC NATIONAL ASSESSMENT AND ACCREDITATION COUNCIL

NBA

## CERTIFICATE OF APPRECIATION

*This is to certify that*

**K. Arun Teja of Third Year CSE**

*has successfully completed the One Week Value Added Course on* ***JAVA PROGRAMMING*** ***organized by the department of Information Technology*** *of Sri Indu College of Engineering and Technology under IQAC held on 21.03.2022 to 26.03.2022.*

Surekha

**Prof. B. Surekha**
HoD/CSIT & IT

**Prof.K.Ashok Babu**
Convener

**Dr.G.Suresh**
Principal

# Sri Indu College of Engineering & Technology

**An Autonomous Institution under UGC**

Recognized under 2(f) and 12(B) of UGC Act 1956
NBA & NAAC Accredited, Approved by AICTE and
Permanently affiliated to JNT University, Hyderabad.

Estd.: 2001
ISO 9001 : 2000

Estd.2001

**Sri Indu**
College of Engineering
UGC Autonomous Institu
Recognized under 2(f) & 12(B) of
NAAC, Approved by AICTE &
Permanently Affiliated to JNTUH

NAAC
NATIONAL ASSESSMENT AND ACCREDITATION COUNCIL
NBA

## CERTIFICATE OF APPRECIATION

*This is to certify that*

**R. Meghana of Third Year IT**

*has successfully completed the One Week Value Added Course on* ***JAVA PROGRAMMING*** ***organized by the department of Information Technology*** *of Sri Indu College of Engineering and Technology under IQAC held on 21.03.2022 to 26.03.2022.*

Prof. B.Surekha
HoD/CSIT & IT

Prof.K.Ashok Babu
Convener

Dr.G.Suresh
Principal

SRI INDU COLLEGE OF ENG. & TECH.

PRINCIPAL
Sri Indu College of Engineering and Technology
(Vill): SHERIGUDA-501 510,
Ibrahimpatnam(M), R.R.Dist.

Estd.2001

**Sri Indu**
College of Engineering & Technology
UGC Autonomous Institution
Recognized under 2(f) & 12(B) of UGC Act 1956,
NAAC, Approved by AICTE &
Permanently Affiliated to JNTUH


**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

# HANDS ON TRAINING COURSE ON IMPLEMENTATION OF IMAGE PROCESSING CONCEPTS FOR REALTIME APPLICATIONS USING MATLAB

## STARTS ON September 19, 2021

SLOT-I REGISTRATION OPEN

Registration : Rs.150
Course Duration : 24 Hours
Weekend Course (Saturday)
Invited Participants: Third Year ECE, EEE, CSE
Restricted to 25 Participants/Slot
Resource Persons: In-house Trainers


Coordinators
Dr.N.C.Sendhilkumar
Dr.P.Mukunthan

Convener
Prof.k.Ashok Babu

Principal
Dr.G.Suresh

Contact: 9443968958,9894145701

**SRI INDU COLLEGE OF ENGINEERING AND TECHNOLOGY**

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**HANDS ON TRAINING COURSE**

**ON**

# IMPLEMENTATION OF IMAGE PROCESSING CONCEPTS FOR REALTIME APPLICATIONS USING MATLAB

Date: From 19.09.2021 (6 Week Course, Only on Saturdays)

## COURSE CONTENTS

| MODULE -1 | | |
|---|---|---|
| Durations | Topics | Resource Person |
| **Week 1** | Basics on Image Processing | Dr.G.Suresh |
| | Introduction to Image Processing Tools | |
| | Types of Image Representation | |
| | Waveform and Amplitude Spectrum | |
| | False Contouring | |
| | Circular correlation between two signals | |
| | Assignment-1 | |
| **Week 2** | Program to Interchage phase between two images | Dr.N.C.Sendhilkumar |
| | Program to adjust brightness and contrast level of an image | |
| | Histogram Analysis of an Image | |
| | Types of noises and removal | |
| | Assignment-2 | |
| **Week 3** | Bit-plane slicing of an Image | Dr.G.Suresh |
| | Analysis of Zoom Factors | |
| | Image blending | |
| | Assignment-3 | |
| MODULE -2 | | |
| Durations | Topics | Resource Person |
| | Program to compute the edges | |

| | | |
|---|---|---|
| **Week 4** | watershed transform | Dr.N.C.Sendhilkumar |
| | Program for erosion and dilation then edge detection | |
| | Program to separate R-G-B from RGB | |
| | Program to separate Missing R-G-B from RGB | |
| | Code that runs conversion of color image to YCbCr | |
| | Assignment-4 | |
| MODULE -3 | | |
| Durations | Topics | Resource Person |
| **Week 5** | DWT based compression | Dr. G. Suresh |
| | Implementation of Arithmetic Coding | |
| | Implementation of Wavelet Transform | |
| | Assessment -1 | |
| | Assignment-5 | |
| **Week 6** | Implementation of Image Retrieval Schemes | Dr.G.Suresh<br>Dr.N.C.Sendhilkumar |
| | Implementation of Image Segmentation Schemes | |
| | Assessment -2 | |
| | Conclusion | |

PRINCIPAL
Sri Indu College of Engineering and Technology
(Vill): SHERIGUDA-501 510,
Ibrahimpatnam(M), R.R.Dist.

Sri Indu
College of Engineering & Technology
UGC Autonomous Institution
Recognized under 2(f) & 12(B) of UGC Act 1956,
NAAC, Approved by AICTE &
Permanently Affiliated to JNTUH
Estd.2001

NAAC
NATIONAL ASSESSMENT AND ACCREDITATION COUNCIL

# HANDS ON TRAINING COURSE

# ON

# IMPLEMENTATION OF IMAGE PROCESSING CONCEPTS FOR REALTIME APPLICATIONS USING MATLAB

## DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

**SRI INDU COLLEGE OF ENGINEERING AND TECHNOLOGY**

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**HANDS ON TRAINING COURSE**

**ON**

# IMPLEMENTATION OF IMAGE PROCESSING CONCEPTS FOR REALTIME APPLICATIONS USING MATLAB

Date: From 19.09.2021 (6 Week Course, Only on Saturdays)

## COURSE CONTENTS

| MODULE -1 | | |
|---|---|---|
| Durations | Topics | Resource Person |
| Week 1 | Basics on Image Processing | Dr.G.Suresh |
| | Introduction to Image Processing Tools | |
| | Types of Image Representation | |
| | Waveform and Amplitude Spectrum | |
| | False Contouring | |
| | Circular correlation between two signals | |
| | Assignment-1 | |
| Week 2 | Program to Interchage phase between two images | Dr.N.C.Sendhilkumar |
| | Program to adjust brightness and contrast level of an image | |
| | Histogram Analysis of an Image | |
| | Types of noises and removal | |
| | Assignment-2 | |
| Week 3 | Bit-plane slicing of an Image | Dr.G.Suresh |
| | Analysis of Zoom Factors | |
| | Image blending | |
| | Assignment-3 | |
| MODULE -2 | | |
| Durations | Topics | Resource Person |
| | Program to compute the edges | |

| | | |
|---|---|---|
| Week 4 | watershed transform | Dr.N.C.Sendhilkumar |
| | Program for erosion and dilation then edge detection | |
| | Program to separate R-G-B from RGB | |
| | Program to separate Missing R-G-B from RGB | |
| | Code that runs conversion of color image to YCbCr | |
| | Assignment-4 | |
| MODULE -3 | | |
| Durations | Topics | Resource Person |
| Week 5 | DWT based compression | Dr. G. Suresh |
| | Implementation of Arithmetic Coding | |
| | Implementation of Wavelet Transform | |
| | Assessment -1 | |
| | Assignment-5 | |
| Week 6 | Implementation of Image Retrieval Schemes | Dr.G.Suresh<br>Dr.N.C.Sendhilkumar |
| | Implementation of Image Segmentation Schemes | |
| | Assessment -2 | |
| | Conclusion | |

PRINCIPAL
Sri Indu College of Engineering and Technology

# DEMONSTRATIVE MODE

# SIGNAL AND IMAGE PROCESSING

## Fourier Transforms

Every signal can be written as a sum of sinusoids with di®erent amplitudes and frequencies. The MATLAB command to compute the Fourier Transform and its inverse are respectively fft and ifft, for example:

```
>> x = rand(1,10); % suppose 10 samples of a random signal
>> y = fft(x); % Fourier transform of the signal
>> iy = ifft(y); % inverse Fourier transform
>> x2 = real(iy); % chop off tiny imaginary parts
>> norm(x-x2); % compare original with inverse of transformed
```

The fft is the abbreviation of Fast Fourier Transform. This algorithm implements the discrete Fourier transform to transform data from time into the frequency domain. The study of this algorithm is normally covered in a good linear algebra course. First we give an example of the meaning of the Fourier transform before showing how Fourier transforms can be used to filter noise from signals.

## Waveform and Amplitude Spectrum

Suppose we sample a signal during 4 seconds, at a sampling rate of 0.01:

```
>> dt = 1/100; % sampling rate
>> et = 4; % end of the interval
>> t = 0:dt:et; % sampling range
>> y = 3*sin(4*2*pi*t) + 5*sin(2*2*pi*t); % sample the signal
```

A natural plot is that of amplitude versus time:

```
>> subplot(2,1,1); % first of two plots
>> plot(t,y); grid on % plot with grid
>> axis([0 et -8 8]); % adjust scaling
>> xlabel('Time (s)'); % time expressed in seconds
>> ylabel('Amplitude'); % amplitude as function of time
```

With the Fourier Transform we can visualize what characterizes this signal the most. From the Fourier transform we compute the amplitude spectrum:

```
>> Y = fft(y); % compute Fourier transform
>> n = size(y,2)/2; % 2nd half are complex conjugates
>> amp_spec = abs(Y)/n; % absolute value and normalize
```

To visualize the amplitude spectrum, we execute the following commands

```
>> subplot(2,1,2); % second of two plots
>> freq = (0:79)/(2*n*dt); % abscissa viewing window
>> plot(freq,amp_spec(1:80)); grid on % plot amplitude spectrum
>> xlabel('Frequency (Hz)'); % 1 Herz = number of cycles/second
>> ylabel('Amplitude'); % amplitude as function of frequency
```

On the amplitude spectrum we see two peaks: at 2 and 4. The location of the peaks occurs at the two frequencies in the signal. The heights of the peaks (5 and 3) are the amplitudes of the sines in the signal.

%Program:

```
x = rand(1,10);          % suppose 10 samples of a random signal
y = fft(x);              % Fourier transform of the signal
iy = ifft(y);            % inverse Fourier transform
x2 = real(iy);           % chop off tiny imaginary parts
norm(x-x2);              % compare original with inverse of transformed
dt = 1/100;              % sampling rate
et = 4;                  % end of the interval
t = 0:dt:et;             % sampling range
y = 3*sin(4*2*pi*t) + 5*sin(2*2*pi*t);   % sample the signal
subplot(2,1,1);                          % first of two plots
plot(t,y); grid on                       % plot with grid
axis([0 et -8 8]);                       % adjust scaling
xlabel('Time (s)');                      % time expressed in seconds
ylabel('Amplitude');                     % amplitude as function of time
Y = fft(y);                              % compute Fourier transform
n = size(y,2)/2;                         % 2nd half are complex conjugates
amp_spec = abs(Y)/n;                     % absolute value and normalize
subplot(2,1,2);                          % second of two plots
freq = (0:79)/(2*n*dt);                  % abscissa viewing window
plot(freq,amp_spec(1:80)); grid on       % plot amplitude spectrum
xlabel('Frequency (Hz)');                % 1 Herz = number of cycles/second
ylabel('Amplitude');                     % amplitude as function of frequency
% Filtering Noise from Signals
noise = randn(1,size(y,2)); % random noise
ey = y + noise;             % samples with noise
eY = fft(ey);               % Fourier transform of noisy signal
n = size(ey,2)/2;           % use size for scaling
amp_spec = abs(eY)/n;       % compute amplitude spectrum
figure                      % plots in new window
subplot(2,1,1);             % first of two plots
plot(t,ey); grid on         % plot noisy signal with grid
axis([0 et -8 8]);          % scale axes for viewing
xlabel('Time (s)');         % time expressed in seconds
```

```
ylabel('Amplitude');          % amplitude as function of time
subplot(2,1,2);               % second of two plots
freq = (0:79)/(2*n*dt);                  % abscissa viewing window
plot(freq,amp_spec(1:80)); grid on       % plot amplitude spectrum
xlabel('Frequency (Hz)');  % 1 Herz = number of cycles per second
ylabel('Amplitude');          % amplitude as function of frequency
figure                % new window for plot
plot(Y/n,'r+')        % Fourier transform of original
hold on               % put more on same plot
plot(eY/n,'bx')               % Fourier transform of noisy signal
fY = fix(eY/100)*100;         % set numbers < 100 to zero
ifY = ifft(fY);               % inverse Fourier transform of fixed data
cy = real(ifY);
figure                % new window for plot
plot(t,cy); grid on    % plot corrected signal
axis([0 et -8 8]);     % adjust scale for viewing
xlabel('Time (s)');   % time expressed in seconds
ylabel('Amplitude');
```

```
% Matlab code for White Gaussian Noise
clc;
clear all;
close all;
randn('state',0);
x=randn(100,1);
subplot(2,1,1)
plot(x)
xlabel('n')
ylabel('x[n]')
grid
subplot(2,1,2)
hist(x)
xlabel('x')
ylabel('no of outcome out of 100')
title('white gaussian noise')
figure
N=100;
nbins=10;
xmin=-3;
xmax=3;
ymax=1;
```

```
[y,xx]=hist(x(1:N),nbins);
delx=xx(2)-xx(1);
bar(xx,y/(N*delx))
grid
axis([xmin xmax 0 ymax]);
xlabel('x')
ylabel('PDF, p(x)')
title('white gaussian noise')
```

1. Consider the following sequence of instructions:

```
>> t = 0:0.1:10;
>> y1 = sin(2*pi*t);
>> y2 = sin(20*pi*t);
>> plot(t,y1);
>> hold on;
>> plot(t,y2);
```

   Why is the output of the second plot like this? Find a better range for `t` to plot `sin(20*pi*t)` right. Can you find a good lower bound for the sampling interval in terms of the frequency?

2. Give the MATLAB commands to plot the amplitude spectrum for the signal

$$f(t) = \sum_{k=10}^{20} (20 - k) \sin(2\pi k t).$$

   In addition, plot the waveform spectrum of this signal.

3. Make a function to plot waveform and amplitude spectrum of a signal. The function has prototype:

```
function specplot ( t, dt, et, y )
%
%  Opens a new figure window with two plots:
%  the waveform and amplitude spectrum of a signal.
%
%  On entry :
%    t          sampling range of the signal;
%    dt         sampling rate;
%    et         end of the range;
%    y          samples of the signal over the range t.
%
```

   So `specplot` computes the amplitude spectrum of the signal. For the abscissa viewing window you may take half of the range of t.

   Test your `specplot` with the signal of the previous assignment.

4. With fft we can decompose a signal in low and high frequencies. Take the example signal from page 1. As noise we now add a sine of amplitude 4 and with frequency 50. Plot the waveform and amplitude spectrum of the new signal. Use fft and ifft to remove this high frequency noise.

**Example 1:**

```
%this program illustrates false contouring
clc
clear all
close all
a=imread('boat.jpg');
subplot(3,2,1);
imshow(a)
title('original image')
%using 128 gray level
%figure,
subplot(3,2,2);
imshow(grayslice(a,128),gray(128)),
title('image with 128 gray level')
%using 64 gray level
subplot(3,2,3);
imshow(grayslice(a,64),gray(64)),
title('image with 64 gray level')
%using 32 gray level
%figure,
subplot(3,2,4);
imshow(grayslice(a,32),gray(32)),
title('image with 32 gray level')
%using 16 gray level
%figure,
 subplot(3,2,5);
imshow(grayslice(a,16),gray(16)),
title('image with 16 gray level')
%using 8 gray level
%figure,
subplot(3,2,6);
imshow(grayslice(a,8),gray(8)),
title('image with 8 gray level')
```

---

Output:


**Example 2:**

```
%frequency response
clc
clear all
close all
[x y]=meshgrid(-pi:0.09:pi);
z=2*cos(x)+2*cos(y);
surf(x,y,z)
axis([-4 4,-4 4,-4 3])
```

**Example 3:**

```
%frequency response
clc
clear all
close all
[x y]=meshgrid(-pi:0.05:pi);
z=2-cos(x)-cos(y);
surf(x,y,z)
axis([-4 4,-4 4,-0.5 4])
```

**Example 4:**

```
%application of circular convolution
x=[1 0;0 0]
h=[1 1;1 1]
x1=fft2(x)
h1=fft2(h)
y1=x1.*h1
res=ifft2(y1)
```

**Example 5:%circular correlation between two signals**

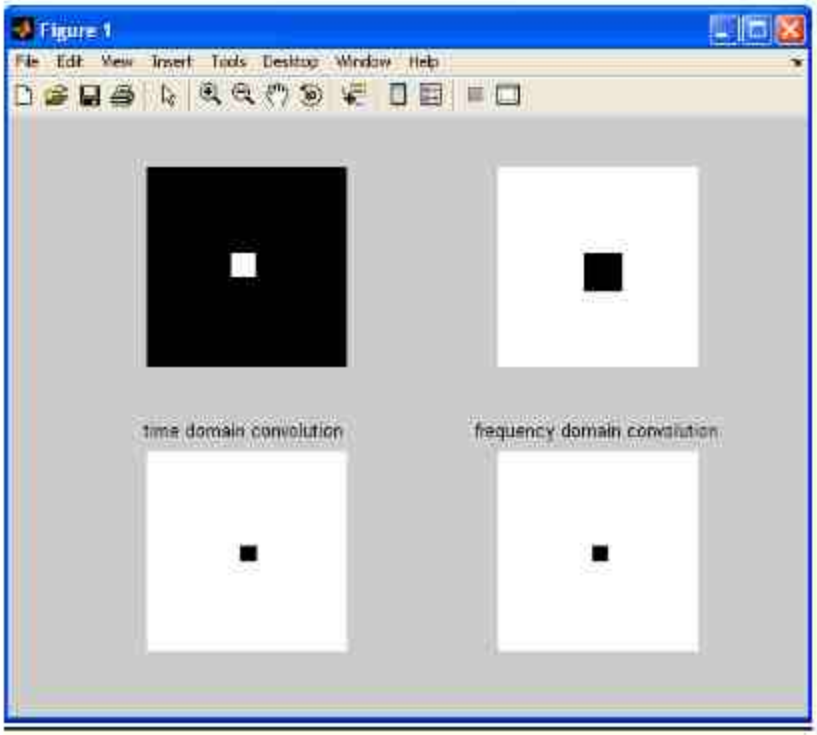
```
clc
clear all
close all
x=[5 10;15 20]
h=[3 6;9 12]
h1=fliplr(h)%fold signal along column-wise
h2=flipud(h1)%fold signal along row-wise
x1=fft2(x);
h3=fft2(h2);
y1=x1.*h3
y2=ifft2(y1)
```

**Example 6:**

```
clc
clear all
close all
%generation of first image A
a=zeros(256);
[m n]=size(a);
for i=110:140
   for j=110:140
      a(i,j)=255;
   end
end
subplot(2,2,1)
imshow(a)
%generation of second  image B
b=ones(256);
[m n]=size(b);
for i=110:160
   for j=110:160
      b(i,j)=0;
   end
```

```
end
subplot(2,2,2)
imshow(b)
%convolution in time domain
c=conv2(a,b,'same');
%multiplication in frequency domain
a1=fft2(a);
b1=fft2(b);
c1=a1.*b1;
d1=fftshift(ifft2(c1));
subplot(2,2,3)
imshow(c)
title('time domain convolution')
subplot(2,2,4)
imshow(d1)
title('frequency domain convolution')
```

**output:**




**Example 8:**

```
clc
clear all
close all
%generation of first image A
a=imread('boat.jpg');
```

```
subplot(3,1,1);
imshow(a)
title('original image')
b=imrotate(a,45,'bilinear','crop');
subplot(3,1,2);
imshow(b)
title('45 degree rotational image')
c=imcrop(b);
%figure;
subplot(3,1,3);
imshow(c)
title('cropped image')
```

**Output:**

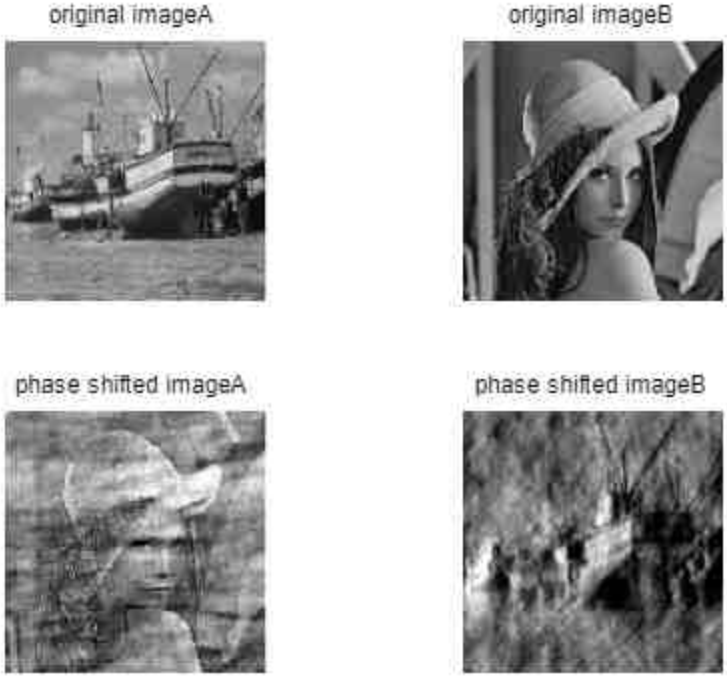

---

**Example 8:**

```
%program to interchage phase between two images
clc
clear all
close all
%generation of first image A
a=imread('boat.jpg');
b=imread('lena.jpg');
ffta=fft2(double(a));
fftb=fft2(double(b));
%get the magnitude and phase components
mag_a=abs(ffta);
ph_a=angle(ffta);
mag_b=abs(fftb);
ph_b=angle(fftb);
```

```
%determine new FFT by interchanging the phase
newfft_a=mag_a.*(exp(i*ph_b));
newfft_b=mag_b.*(exp(i*ph_a));
%reconstruct the original image using inverse FFT
rec_a=ifft2(newfft_a);
rec_b=ifft2(newfft_b);
subplot(2,2,1)
imshow(a)
title('original imageA');
subplot(2,2,2)
imshow(b)
title('original imageB');
subplot(2,2,3)
imshow(uint8(rec_a))
title('phase shifted imageA');
subplot(2,2,4)
imshow(uint8(rec_b))
title('phase shifted imageB');
```

**Output:**




---

**Example 9:**

```
% Fourier transform of Fourier Transform
clc
clear all
```

```
close all
%generation of first image A
%a=imread('boat.jpg');
a=imread('lena.jpg');
[m n]=size(a);
b=fft2(a);
%spectrum of spectrum
c=(1/(m*n))*fft2(b);
subplot(2,2,1),imshow(a),title('input image');
subplot(2,2,2),imshow(uint8(c)+40),title('spectrum of spectrum');
```

**Output:**


---

**Example 10**

%program to adjust brightness and contrast level of an image

```
clc
clear all
close all
a=imread('lena.jpg');
[m n]=size(a);
b=double(a)+50;
c=double(a)-70;
subplot(3,2,1);
imshow(a)
title('original image');
subplot(3,2,2);
imshow(uint8(b))
title('brightness enhanced image');
subplot(3,2,3);
```

```
imshow(uint8(c))
title('brightness supressed image');
d=a*.5;
e=a*20;
subplot(3,2,4);
imshow(uint8(d))
title('contrast increased image');
subplot(3,2,5);
imshow(uint8(e))
title('contrast decreased image');
```

**output:**

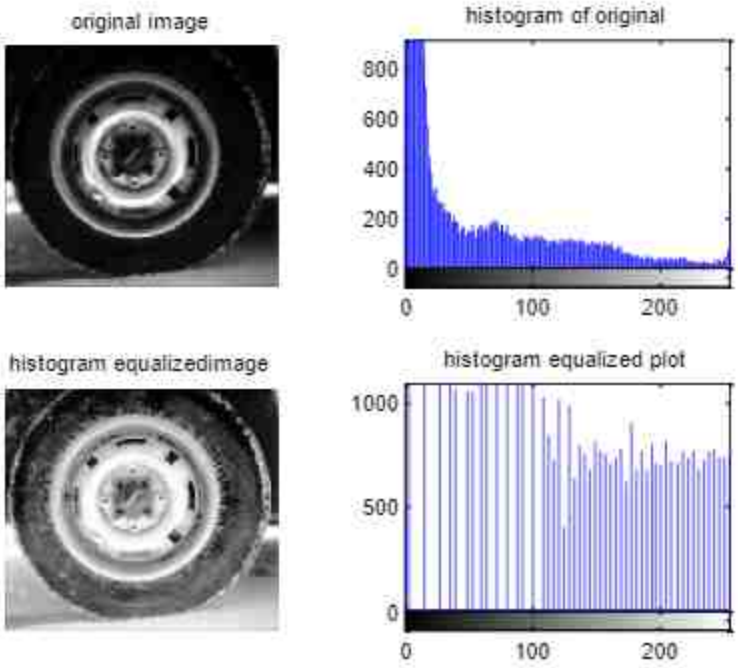

---

**Example 11:**

```
clc
clear all
close all
I = imread('tire.tif');
K = histeq(I);
subplot(2,2,1);
imshow(I)
title('original image');
subplot(2,2,2);
imhist(I)
title('histogram of original');
subplot(2,2,3);
```

```
imshow(K)
title('histogram equalizedimage');
subplot(2,2,4);
imhist(K)
title('histogram equalized plot');
```

**output;**



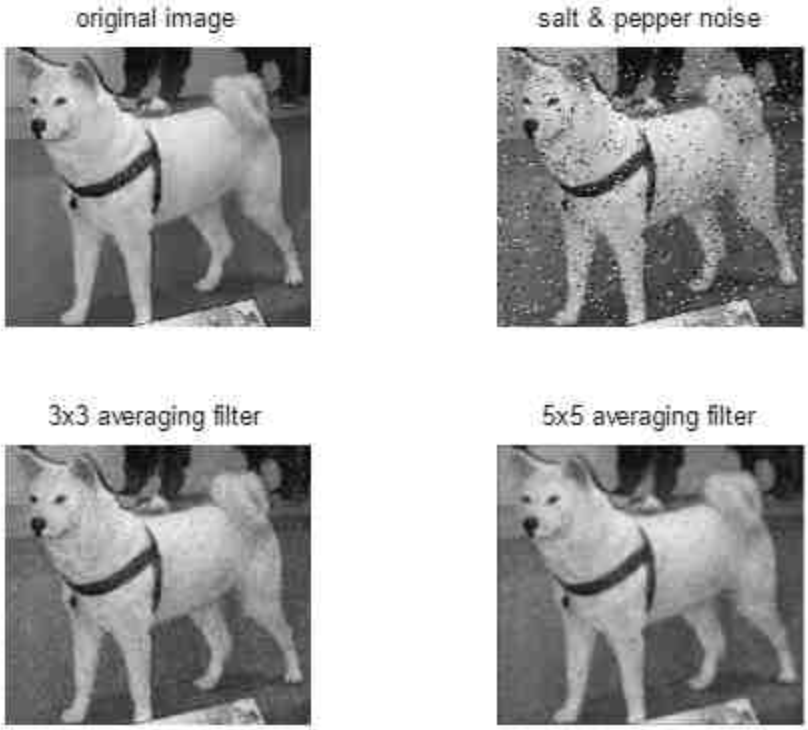

---

**Example 12:**

```
% Types of noises and removal
a=imread('dog.jpg');
a=rgb2gray(a);
b=imnoise(a,'salt & pepper');
c=imnoise(a,'gaussian');
d=imnoise(a,'speckle');
%defining 3x3 and 5x5 kernal
h1=1/9*ones(3,3);
h2=1/25*ones(5,5);
%attempt to recover the image
b1=conv2(b,h1,'same');
b2=conv2(b,h2,'same');
c1=conv2(c,h1,'same');
c2=conv2(c,h2,'same');
d1=conv2(d,h1,'same');
d2=conv2(d,h2,'same');
```

```
figure,subplot(2,2,1),imshow(a),title('original image'),
subplot(2,2,2),imshow(b),title('salt & pepper noise'),
subplot(2,2,3),imshow(uint8(b1)),title('3x3 averaging filter'),
subplot(2,2,4),imshow(uint8(b2)),title('5x5 averaging filter')
%-------------------------------------------------------------
figure,subplot(2,2,1),imshow(a),title('original image'),
subplot(2,2,2),imshow(c),title('Gaussian noise'),
subplot(2,2,3),imshow(uint8(c1)),title('3x3 averaging filter'),
subplot(2,2,4),imshow(uint8(c2)),title('5x5 averaging filter')
%-------------------------------------------------------------
figure,subplot(2,2,1),imshow(a),title('original image'),
subplot(2,2,2),imshow(d),title('speckle noise'),
subplot(2,2,3),imshow(uint8(d1)),title('3x3 averaging filter'),
subplot(2,2,4),imshow(uint8(d2)),title('5x5 averaging filter')
```

**output:**



original image

Gaussian noise

3x3 averaging filter

5x5 averaging filter

---

original image

speckle noise

3x3 averaging filter

5x5 averaging filter

---

**Example 13:**

```
% bitplane slicing
clc
clear all
close all
a=imread('dog.jpg');
a=rgb2gray(a);
subplot(2,2,1)
imshow(a);
title('original image')
[m n]=size(a);
n1=input('enter the bit plane number (8 to 1 that to be removed:');
s=255-(2^(n1-1));
for i=1:m,
   for j=1:n,
      out_I(i,j)=bitand(a(i,j),s);
   end
end
subplot(2,2,2)
imshow(uint8(out_I));title(sprintf(' plane eliminated is %g',n1))
n1=input('enter the bit plane number (8 to 1 that to be removed:');
s=255-(2^(n1-1));
for i=1:m,
   for j=1:n,
      out_I(i,j)=bitand(a(i,j),s);
   end
end
subplot(2,2,3)
imshow(uint8(out_I));title( sprintf('plane eliminated is %g',n1));
n1=input('enter the bit plane number (8 to 1 that to be removed:');
s=255-(2^(n1-1));
for i=1:m,
   for j=1:n,
      out_I(i,j)=bitand(a(i,j),s);
   end
end
subplot(2,2,4)
imshow(uint8(out_I));title(sprint(' plane eliminated is %g',n1));
```

**Output:**

original image

6th plane eliminated

7th plane eliminated

8th plane eliminated

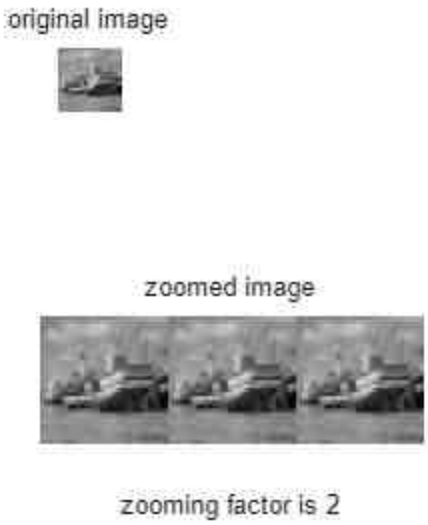
---

**Example 14:**

```
clc;
clear all;
close all;
a=imread('boat.jpg');a=imresize(a,[32 32]);
[m n]=size(a);
p=input('Enter the size you want: ');
for i=1:m %loop to extract every row
  for j=1:n %loop to extract every column
    for k=1:p %loop to control the number of replication
       b(i,(j-1)*p+k)=a(i,j); %replication of pixels in row wise
    end
  end
end
c=b;
[m n]=size(c);
for i=1:n %loop to extract every column
  for j=1:m %loop to extract every row
    for k=1:p %loop to control the number of replication
       b((j-1)*p+k,i)=c(j,i); %replication of pixels in column wise
```

```
   end
  end
end
imshow(a),title('original image')
figure,imshow(b),title('zoomed image')
xlabel(sprintf('zooming factor is %g',p))
```

**Output:**




**Example 15:**

```
a=imread('lena.jpg');%a=imresize(a,[64 64]);
zooming_factor=input('enter the zooming facttor:');
num=zooming_factor;den=1;
while(num-floor(num)~=0)
   num=num*2;den=den*2;
end
[m n]=size(a);s1=num*m;
re=zeros(s1,num*n);
for i=1:m,
   for j=1:n,
      k=num*(i-1);
      l=num*(j-1);
      re(k+1,l+1)=a(i,j);
   end
end
   i=1;
   while(i<=(s1))
      j=1;
      while(j<=(num*n))
```

```
        x=ones(num,num);
        for p=1:num,
           for q=1:num,
              c(p,q)=re(i,j);
              j=j+1;
           end
           i=i+1;j=j-num;
        end
        z=ifft2(fft2(c).*fft2(x));
        i=i-num;
        for p=1:num,
           for q=1:num,
              re(i,j)=z(p,q);
              j=j+1;end
           i=i+1;j=j-num;end
        i=i-num;j=j+num;end
     i=i+num;end
  if(den>1)
     m=den;[p q]=size(re);
     a=double(re);
     for i=1:ceil(p/m),
        for j=1:ceil(q/m),
           if(((m*i)<p)&((m*j)<q))
              b(i,j)=re(m*i,m*j);
           else b(i,j)=0;
           end
        end
     end
  else b=re;end
  figure,imshow(uint8(b));
```

**Output:**

**Example 16:**

```
% Image blending
clc
clear all
close all
%c=(1-x)a+xb
a=imread('lena.jpg');
a=rgb2gray(a);subplot(2,2,1);
imshow(a)
[m n]=size(a);
title('Image 1');
b=imread('boat.jpg');
b=rgb2gray(b);
b1=imresize(b,[256 256]);subplot(2,2,2);
imshow(b1)
title('Image 2');
c1=a+b1;
subplot(2,2,3);
imshow(c1)
title('blended Image');
x=input('enter x value:')
for i=1:m,
   for j=1:n,
      c2(i,j)=(1-x)*a(i,j)+x*b1(i,j);
   end
end
subplot(2,2,4);
imshow(c2)
title(sprintf('blended Image of %g',x));
```

**output:**

Image 1

Image 2

blended Image

blended Image of 0.7

**Example 17:**

```
%this program is to perform median filtering of the image
clc
clear all
close all
a=imread('dog.jpg');
a=rgb2gray(a);
b=imnoise(a,'salt & pepper',0.2);
b=double(b);
[m n]=size(b);
N=input('enter the window size:');
out_img=b;
if(mod(N,2)==1)
   Start=(N+1)/2;
   End=Start;
else
   Start=N/2;
End=Start+1;
end
if(mod(N,2)==1)
   limit1=(N-1)/2;
```

```
   limit2=limit1;
else
   limit1=(N/2)-1;
   limit2=limit1+1;
end
for i=Start:(m-End+1),
   for j=Start:(n-End+1),
      I=1;
      for k=-limit1:limit2,
         for l=-limit1:limit2,
            mat(I)=a(i+k,j+1);
            I=I+1;
         end
      end
      mat=sort(mat);
      if(mod(N,2)==1)
         out_img(i,j)=(mat(((N^2)+1)/2));
      else
         out_img(i,j)=(mat((N^2)/2)+mat(((N^2)/2)+1))/2;
      end
   end
end
subplot(1,3,1)
imshow(a)
title('original image');
subplot(1,3,2)
imshow(uint8(b))
title('noissy image')
subplot(1,3,3)
imshow(uint8(out_img))
title(sprintf('median filtered with window size %gx%g',N));
```

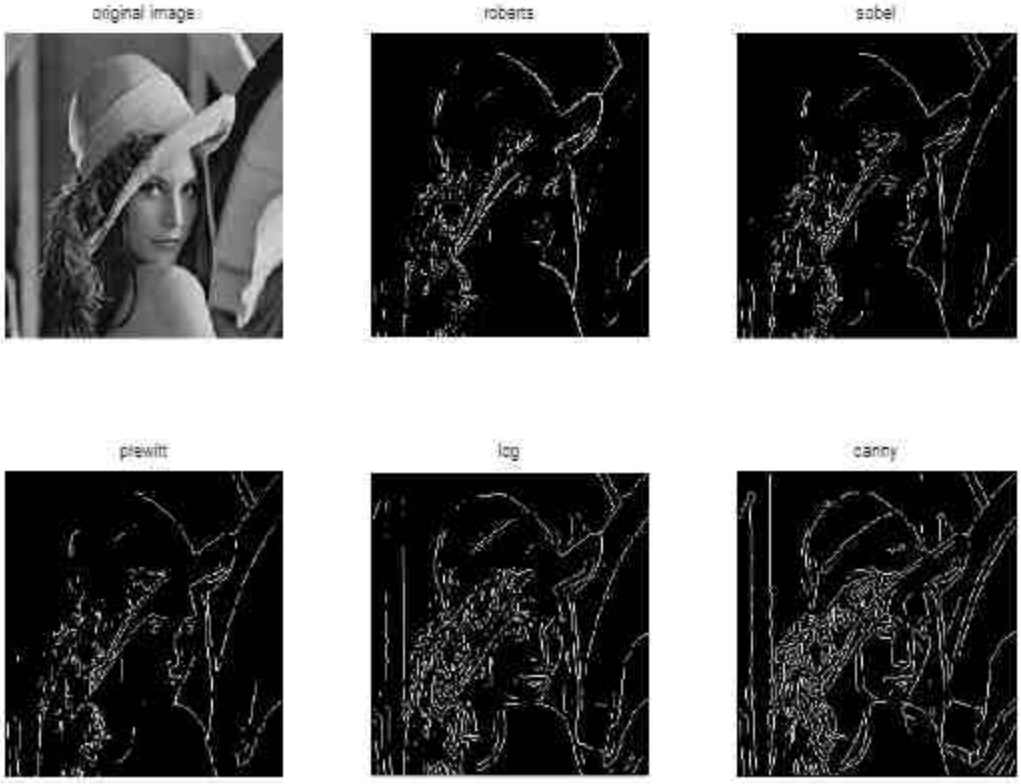
**Output:**


**Example 18:**

```
%program to compute the edges
clc
clear all
close all
a=imread('lena.jpg');
a=rgb2gray(a);
b=edge(a,'roberts');
c=edge(a,'sobel');
d=edge(a,'prewitt');
e=edge(a,'log');
f=edge(a,'canny');
%b=edge(a,'roberts');
subplot(2,3,1)
imshow(a)
title('original image')
subplot(2,3,2)
imshow(b)
title('roberts')
subplot(2,3,3)
imshow(c)
title('sobel')
subplot(2,3,4)
imshow(d)
```

```
title('prewitt')
subplot(2,3,5)
imshow(e)
title('log')
subplot(2,3,6)
imshow(f)
title('canny')
```

**Output:**

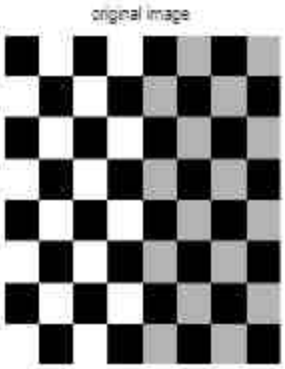


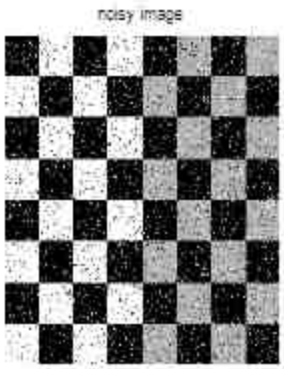

**Example 19:**

```
%watershed transform
clc
clear all
close all
a=checkerboard(32);
a1=imnoise(a,'salt & pepper',0.1);
b=watershed(a,4);
b1=watershed(a1,4);
subplot(2,2,1)
```

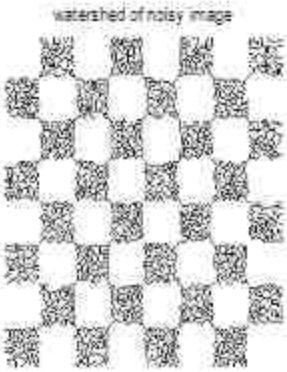
```
imshow(a),title('original image');
subplot(2,2,2);
imshow(a1),title('noisy image');
subplot(2,2,3);
imshow(b),title('watershed of original');
subplot(2,2,4);
imshow(b1),title('watershed of noisy image');
```

**Output:**





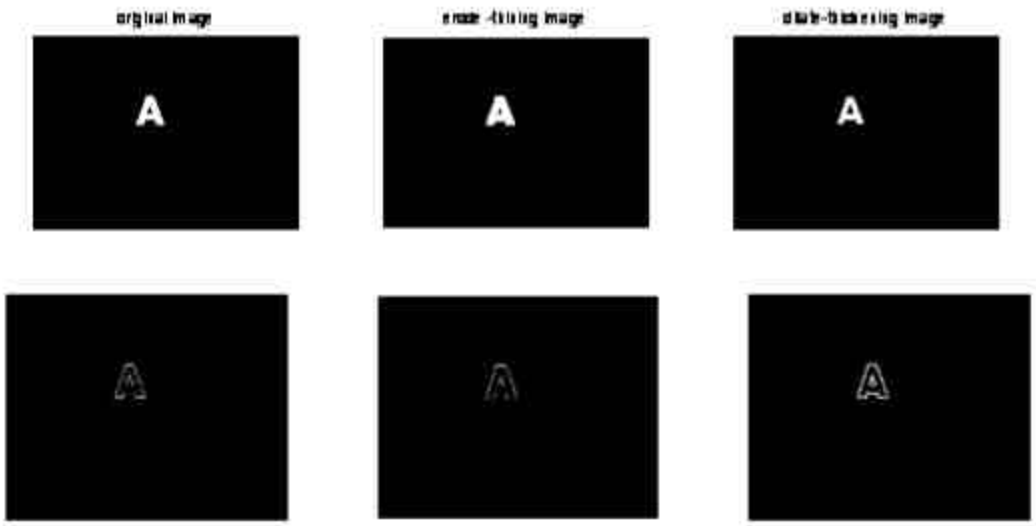



**Example 20:**

```
%program for erosion and dilation then edge detection
clc
clear all
close all
a=imread('sur.jpg');
b=[1 1 1;1 1 1;1 1 1];;
a1=imdilate(a,b);
a2=imerode(a,b);
```

```
subplot(1,3,1)
imshow(a),title('original image');
subplot(1,3,2)
imshow(a1),title('erode -thining image');
subplot(1,3,3)
imshow(a2),title('dilate-thickening image');
a3=a-a2;
a4=a1-a;
a5=a1-a2;
figure
subplot(1,3,1)
imshow(a3),%title('');
subplot(1,3,2)
imshow(a4),%title('erode -thining image');
subplot(1,3,3)
imshow(a5),%title('dilate-thickening image');
```

**Output:**

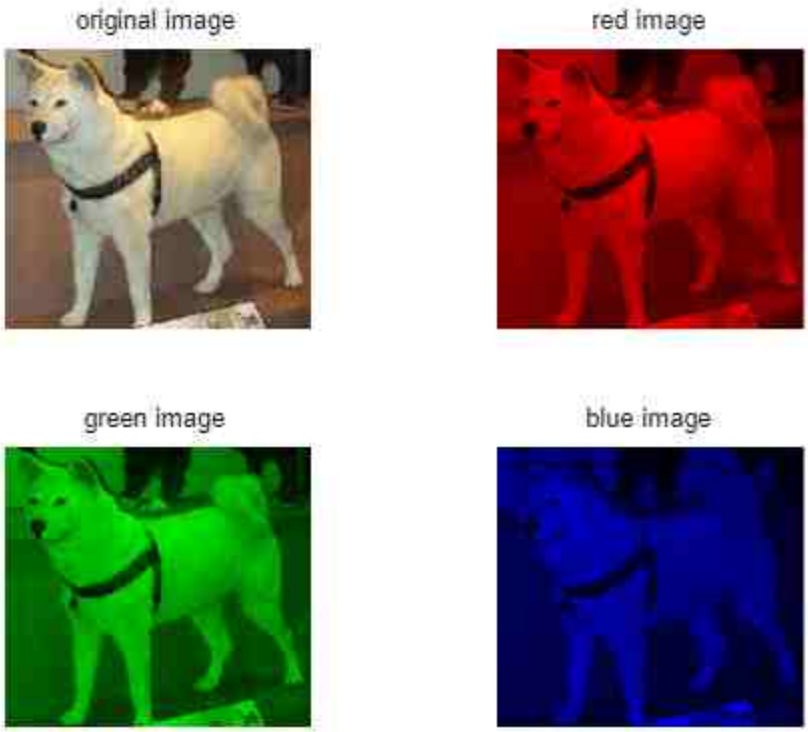



**Example 21:**

```
%program to separate R-G-B from RGB
RGB=imread('dog.jpg');
R=RGB;
G=RGB;
B=RGB;
R(:,:,2)=0;
R(:,:,3)=0;
G(:,:,1)=0;
```

```
G(:,:,3)=0;
B(:,:,1)=0;
B(:,:,2)=0;
subplot(2,2,1),imshow(RGB),title('original image')
subplot(2,2,2),imshow(R),title('red image')
subplot(2,2,3),imshow(G),title('green image')
subplot(2,2,4),imshow(B),title('blue image')
```

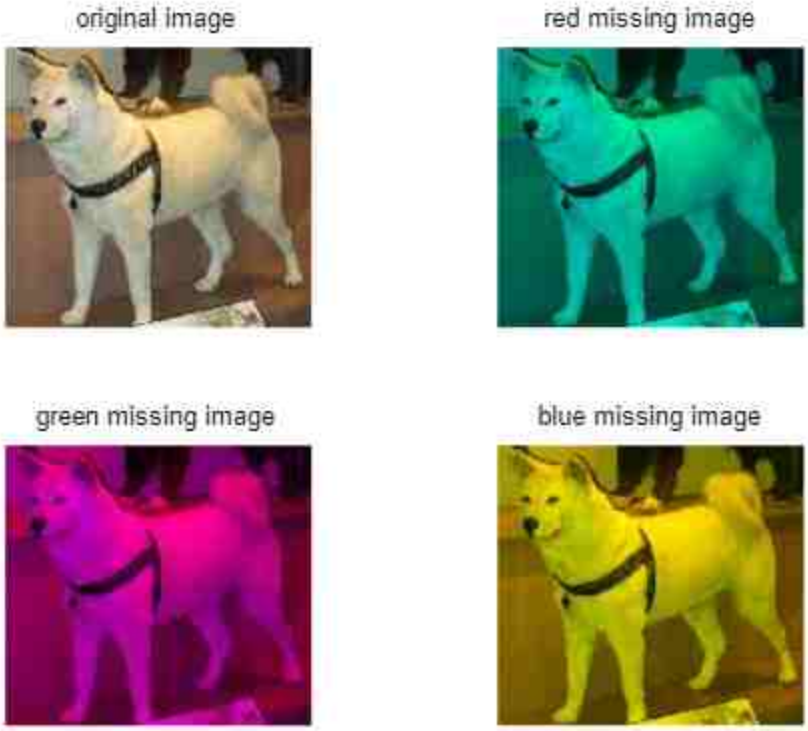
**output:**




**Example 22:**

```
%program to separate Missing R-G-B from RGB
RGB=imread('dog.jpg');
R=RGB;
G=RGB;
B=RGB;
R(:,:,1)=0;
G(:,:,2)=0;
B(:,:,3)=0;
subplot(2,2,1),imshow(RGB),title('original image')
subplot(2,2,2),imshow(R),title('red missing image')
subplot(2,2,3),imshow(G),title('green missing image')
```

subplot(2,2,4),imshow(B),title('blue missing image')




**Example 23:**

%Code that runs conversion of color image to YCbCr

%read in image filename
%inimage = input('Enter image file name with extension (like jennifer.bmp): ', 's');

%open image file
inimage = imread('dog.jpg');

%display on screen the image
figure(1), imshow(inimage); title('Original Image');

%the command size returns the size of the matrix/image
%A semi-colon suppresses the screen output of the variable
%values, while the lack of semi-colon prints it to the screen

size(inimage)

```
U = rgb2ycbcr(inimage);
figure(1), imshow(inimage); title('RGB image');
figure(2), imshow(U); title('YCBCR Image');
size(U)

%Here pick off the 256x256 luminance part of the ycbcr image
Y = U(:,:,1);
figure(3), imshow(Y); title('Y part of Image');
size(Y)

%Here pick off the 256x256 Cb part of the ycbcr image
CB = U(:,:,2);
figure(4), imshow(CB); title('Cb part of Image');
size(CB)

%Here pick off the 256x256 Cr part of the ycbcr image
CR = U(:,:,3);
figure(5), imshow(CR); title('Cr part of Image');
size(CR)
```

**Example 24:**

```
%dwt based compression
clc
clear all
close all
a=imread('dog.jpg');
[p q r t]=dwt2(a,'db1')
b=[uint8(p),q;r t];
[p1 q1 r1 t1]=dwt2(p,'db1');
b1=[p1 q1; r1 t1];imshow(b);
% b2=[b1,q;r,t];
% imshow(b2);
```

**Output:**

**Example 25:**

```
%C:\Documents and Settings\gsuresh\Desktop\Desktop 09-08-2012\code1
%boat.jpg %lena.jpg

%This program hides a message image in the lower
%bit planes of a cover image
%read in cover image filename
covername = input('Enter image file name with extension (like jennifer.bmp): ', 's');
%read in message image filename
messagename = input('Enter message image file name with extension: ', 's');
%open cover and message image files
cover = imread(covername);
message = imread(messagename);
%display on screen the two images
figure(1), imshow(cover); title('Original Image (Cover Image)');
figure(2), imshow(message);title('Image to Hide (Message Image)');
%change to double to work with addition below
cover=double(cover);
message=double(message);
%imbed = no. of bits of message image to embed in cover image
imbed=4;
%shift the message image over (8-imbed) bits to right
messageshift=bitshift(message,-(8-imbed));
%show the message image with only embed bits on screen
%must shift from LSBs to MSBs
showmess=uint8(messageshift);
showmess=bitshift(showmess,8-imbed);
figure(3),imshow(showmess);title('4 Bit Image to Hide');
%now zero out imbed bits in cover image
coverzero = cover;
```

```
for i=1:imbed
coverzero=bitset(coverzero,i,0);
end
%now add message image and cover image
stego = uint8(coverzero+messageshift);
figure(4),imshow(stego);title('Stego image');
%save files if need to
%4 bit file that was embedded = same as file extracted
imwrite(showmess,'showmess4.bmp'); %use bmp to preserve lower bits
%jpg will get rid of them
%stego file
imwrite(stego,'stego4.bmp');
```

**output:**

Original Image (Cover Image)

4 Bit Image to Hide

Image to Hide (Message Image)

Stego image

| Email Address | Hall Ticket Number | Name of the Student (as per SSC) | Department | Willing to choose to the Course | Contact Number |
|---|---|---|---|---|---|
| mittureddy2707@gmail.com | 18D41A04F9 | P.vivek vardhan reddy | ECE | Yes | 9390392706 |
| mohammedadnan2000.ad@gmail.com | 18D41A04E0 | mohammed adnan | ECE | Yes | 9490074747 |
| neeleshgoud45@gmail.com | 18D41A04F2 | NEELESH KUMAR GOUD | ECE | Yes | 7993292705 |
| ganeshparitala2701@gmail.com | 18D41A04G0 | PARITALA GANESH | ECE | Yes | 9121816459 |
| abhishekramavath167@gmail.com | 18D41A04N8 | Ranavath Abhishek Raj Naik | ECE | Yes | 7780526087 |
| vamshidharreddy709@gmail.com | 18D41A04L5 | THALASANI VAMSHIDHAR REDDY | ECE | Yes | 7095880325 |
| tharunreddy2000t@gmail.com | 18D41A04E9 | N.Tharun kumar reddy | ECE | Yes | 9652590841 |
| masadiajay@gmail.com | 18D41A04D7 | Masadi ajay | ECE | Yes | 8303582156 |
| sudheervenkat2000@gmail.com | 18D41A04G7 | Pirla venkata sudheer | ECE | Yes | 9381785078 |
| manasareddy1813@gmail.com | 17D41A0483 | Gudipally.manasa | ECE | Yes | 9951456583 |
| varunnaidu1638@gmail.com | 17D41A04J0 | RAMISHETTI VARUN KUMAR | ECE | Yes | 9278299278 |
| Rishwanth1212@gmail.com | 17D41A0408 | Angoli Rishwanth Goud | ECE | Yes | 7036708818 |
| punnakarthik91@gmail.com | 18D41A04Q0 | PUNNA KARTHIK | ECE | Yes | 8008053834 |
| varshniamrutha222000@gmail.com | 18D41A04H1 | AMRUTHA VARSHINI PONNA | ECE | Yes | 7995765935 |
| indrakumardasari571@gmail.com | 19D45A0418 | D INDRA KUMAR | ECE | Yes | 7680960312 |
| chandu9polo@gmail.com | 18D41A04H0 | POLOJU POORNA CHANDAR | ECE | Yes | 7995005935 |
| vaishnavipandillaplly27@gmail.com | 18D41A04P8 | Vaishnavi | ECE | Yes | 8096881024 |
| manasareddy1813@gmail.com | 17D41A0483 | Gudipally.manasa | ECE | Yes | 9951456583 |
| roshansamala4@gmail.com | 18D41A04J8 | SAMALA ROSHAN | ECE | Yes | 9985447531 |
| umeshoutlook010@gmail.com | 18D41A04F3 | P UNESHCHANDRA | ECE | Yes | 7674933730 |
| reddysathwika51@gmail.con | 18D41A04M5 | Vadyala sathwika | ECE | Yes | 7207000753 |
| nishithamuppa68@gmail.com | 18D41A04C2 | M Nishitha | ECE | Yes | 9848437534 |
| surepallymounika123@gmail.com | 18D41A04L3 | SUREPALLY MOUNIKA | ECE | Yes | 9347128860 |
| surepallymounika123@gmail.com | 18D41A04L3 | Surepally.Mounika | ECE | Yes | 9347128860 |
| bhavaniyamsani07@gmail.com | 18D41A0425 | Bhavani yamsani | ECE | Yes | 9912007166 |
| karthikrajpyarsani92@gmail.com | 18D41A04H5 | Pyarasani karthik | ECE | Yes | 9553441135 |
| naniraju1438@gmail.com | 18D41A04H3 | Pusapati Narasimha Raju | ECE | Yes | 9346437530 |
| ashrithputtoju@gmail.com | 18D41A04H4 | PUTTOJU ASHRITH CHAITHANYA | ECE | Yes | 8466833744 |
| aishuckanth01@gmail.com | 18D41A04G3 | PASULA AISHWARYA | ECE | Yes | 9059558421 |
| Surepallypavan555@gmail.com | 18D41A04L4 | Surepally Pavan | ECE | Yes | 9966447967 |
| preethiarchu16@gmail.com | 18D41A04J9 | Sampati archana | ECE | Yes | 8304734128 |
| rangineniaparna@gmail.com | 18D41A04J0 | Rangineni Aparna | ECE | Yes | 8303011252 |
| nishureddy974@gmail.com | 18D41A04F1 | NARSAPOLA NISHANTH REDDY | ECE | Yes | 7893212245 |
| chandanareddy1307@gmail.com | 18D41A04P2 | R Chandana | ECE | Yes | 7569596720 |
| kmadhukar52@gmail.com | 18D41A04A5 | KONDURI MADHUKAR | ECE | Yes | 8499991740 |
| naveenkumar84242@gmail.com | 18D41A04H2 | Poshanaboina Naveen Kumar | ECE | Yes | 8247569634 |
| sairohithk27@gmail.com | 18D41A04A0 | KODAMARTHI SAI ROHITH | ECE | Yes | 9959311982 |
| devakrishnakanth18@gmail.com | 18D41A0481 | Deva Krishna kanth | ECE | Yes | 9390862989 |
| chandanareddy1307@gmail.com | 18D41A04P2 | R.Chandana | ECE | Yes | 7569596720 |
| sarayuredii27@gmail.com | 18D41A04F4 | PAKALA SARAYU | ECE | Yes | 7675819696 |
| sumanthreddybagannagari@gmail.com | 17D41A0419 | Bagannagari Sumanth Reddy | ECE | Yes | 9182339625 |
| preetham78333@gmail.com | 18D41A04E4 | Munagala Sai Preetham | ECE | Yes | 8309102520 |

| | | | | | |
|---|---|---|---|---|---|
| ruchikareddy03@gmail.com | 18D41A04P4 | NANNURI RUCHIKA REDDY | ECE | Yes | 7997078366 |
| harshaaravilli@gmail.com | 17D41A0411 | Aravilli Venkata Chakra Shree Harsha | ECE | Yes | 8121271605 |
| kondalreddy738@gmail.com | 17D41A04D1 | MAILA KONDAL REDDY | ECE | Yes | 7702261384 |
| abhinavreddy1191@gmail.com | 18d41a0431 | B ABHINAV REDDY | ECE | Yes | 7675817201 |
| ksandeepreddy8825@gmail.com | 17D41A0499 | KALLURI SANDEEP REDDY | ECE | Yes | 9100503064 |
| pavankasina29@gmail.com | 17D41A04A3 | K PAVAN KUMAR | ECE | Yes | 8374483524 |
| nitishkotagiri@gmail.com | 17D41A04B7 | Kotagiri nitish | ECE | Yes | 8099302577 |
| asairuthwik@gmail.com | 18D41A0497 | AAVULA SAIRUTHWIK VAMSHIKAR | ECE | Yes | 8306588622 |
| sandeepmandadi877@gmail.com | 18D41A04D3 | MANDADI SANDEEP | ECE | Yes | 9347103348 |
| pasulaakhila2@gmail.com | 18D41A04G4 | PASULA AKHILA | ECE | Yes | 9381783056 |
| ajayreddyajji143@gmail.com | 17D41A0493 | JANGA AJAY | ECE | No | 9581568831 |
| ssamhitha66@gmail.com | 18D41A04K4 | Samhitha | ECE | Yes | 9978895966 |
| bharathsimhareddy786@gmail.com | 17D41A04B3 | Koppula Bharath Simha Reddy | ECE | Yes | 7799636959 |
| sejaljoshi148@gmail.com | 18D41A04K0 | Sejal joshi | ECE | Yes | 9393929494 |
| sharathgupta120@gmail.com | 17D41A04N4 | YAMA SHARATH KUMAR | ECE | Yes | 7036454741 |
| bhavaniyamsani87@gmail.com | 18D41A0425 | Bhavani yamsani | ECE | Yes | 9912007186 |
| ankithareddygunreddy@gmail.com | 17D41A0488 | Hundreds Ankitha | ECE | Yes | 7036417013 |
| dendijeevanprakashreddy@gmail.com | 18D41A0457 | Jeevan Prakash Reddy | ECE | Yes | 9398719812 |
| neeraja2442@gmail.com | 17D41A0478 | Golla Neeraja | ECE | Yes | 6304445840 |
| kallakurihemanthsai@gmail.com | 18D45A0401 | KALLAKURI HEMANTH SAI | ECE | Yes | 7382041663 |
| gounisindhu123@gmail.com | 17D41A0480 | Gowni Sindhuja | ECE | Yes | 9553316488 |
| medipallypranay@gmail.com | 17D41A04H4 | Pranay Reddy M | ECE | Yes | 9010666362 |
| ar889958@gmail.com | 17d41a04f9 | INUKOLLU ANIRUDH REDDY | ECE | Yes | 8688610625 |
| bhuvanasri273@gmail.com | 17D41A04C6 | Lingala Bhuvana Sri | ECE | Yes | 9640476559 |
| ankithkumar7286@gmail.com | 17D41A0404 | Ageer Ankith kumar | ECE | Yes | 7286923210 |
| aksharalaya8@gmail.com | 17D41A0414 | B Laya | ECE | Yes | 7330816770 |
| Nenavathpraveen10@gmail.com | 17D41A04F6 | NENAVATH PRAVEEN | ECE | Yes | 8186868183 |
| sujithnaik001@gmail.com | 17D41A04F7 | NENAVATH SUJITH NAIK | ECE | Yes | 8309284211 |
| prashantherra7@gmail.com | 17D41A0464 | Erra Prashanth | ECE | Yes | 9849269539 |
| madipeddivarshith.99@gmail.com | 17D41A04D0 | MADIPEDDI VARSHITH | ECE | Yes | 7996734837 |
| harshapawana1385@gmail.com | 18D41A04L2 | Sunkara. Harsha pawana | ECE | Yes | 7995530648 |
| vamsi.megavath@gmail.com | 17D41A04E0 | Megavath Sai Vamshi | ECE | Yes | 7893283765 |
| kolukolapallybikku888@gmail.com | 19D45A0406 | KOLKULAPALLY BIKKU | ECE | Yes | 7995049810 |
| sainath.reddy4242@gmail.com | 17D41A04D4 | MANDADI SAINATH REDDY | ECE | Yes | 8309609415 |
| vaishnavi.goud06@gmail.com | 18D41A04L6 | Thalla vaishnavi | ECE | Yes | 8686228893 |
| keerthanas819@gmail.com | 17D41A04H1 | Peepalpati sai keerthana | ECE | Yes | 7036920436 |
| narendraraju7033@gmail.com | 17D41A04J9 | REDDICHARLA NARENDRA RAJU | ECE | Yes | 8367457033 |
| pavancaeser321@gmail.com | 17d41a0461 | Durishety Pavan sai | ECE | Yes | 9110747797 |
| reddyrohith217@gmail.com | 17D41A0409 | Annapureddy Rohithreddy | ECE | Yes | 9182773949 |
| akhilogitla163@gmail.com | 17D41A04C7 | LOGITLA AKHIL KUMAR | ECE | Yes | 7036944812 |
| dendijeevanprakashreddy@gmail.com | 18D41A0457 | Dendi Jeevan Prakash Reddy | ECE | Yes | 9398719812 |
| bethapudiraghuvarma@gmail.com | 17D41A0427 | Bethapudi Raghu varma | ECE | Yes | 8374038521 |
| nagatipavankalyan@gmail.com | 17D41A04F2 | NAGATI PAVAN KLAYAN | ECE | Yes | 9550308441 |
| keshagoniprasanna123@gmail.com | 17D41A04A5 | Keshagoni Prasanna | ECE | Yes | 7893421223 |
| arjunreddy8555@gmail.com | 18D41A04A3 | Kolli Leela Krishna Reddy | ECE | Yes | 8309558801 |
| kumarnikhil0606@gmail.com | 18D41A04A6 | KOPPU NIKHIL KUMAR | ECE | Yes | 9676685972 |

| | | | | | |
|---|---|---|---|---|---|
| kamarsingh881@gmail.com | 18D41A0485 | K Amarsingh | ECE | Yes | 8340991337 |
| nivas.nlg@gmail.com | 18D41A04C4 | Maddala Srinivas | ECE | Yes | 9.19642E+11 |
| mohammedsameer902@gmail.com | 18D41A04D9 | MOHAMMAD SAMEER | ECE | Yes | 9848526190 |
| mahalsasai333@gmail.com | 18D41A04C6 | MAHALSA SAI DONTHA | ECE | Yes | 9398409172 |
| mnagaadithya8434@gmail.com | 17D41A04E8 | Munagapati naga adithya | ECE | Yes | 8522962442 |
| pauldayakar6125@gmail.com | 17D41A0420 | BALUSULA PAUL DAYAKAR | ECE | Yes | 9963968461 |
| bvenkatesh0432@gmail.com | 17D41A0432 | BOGGAVARAPU VENKATESH | ECE | Yes | 9866278613 |
| koresandhya04@gmail.com | 17D41A04B4 | Kore Sandhya Rani | ECE | Yes | 9100897924 |
| madduri.sameer39@gmail.com | 18D41A04C5 | Madduri Venkata Sameer Kumar | ECE | Yes | 9849656622 |
| madhavaramsriram@gmail.com | 18D41A04P5 | Madhavaram sriram | ECE | Yes | 9949089587 |
| anilkumardara222@gmail.com | 17D41A0447 | D Anil Kumar | ECE | Yes | 9676389440 |
| rajithasatyam2@gmail.com | 19D41A0492 | Koppula Rajitha | ECE | Yes | 9618101271 |
| sushmareddy320@gmail.com | 17D41A04D3 | Manda sushma | ECE | Yes | 7013247898 |
| kunchalavenkatesh007@gmail.com | 17D41A04C1 | Kunchala venkatesh | ECE | Yes | 9182441549 |
| madduri.sameer39@gmail.com | 18D41A04C5 | Madduri Venkata Sameer Kumar | ECE | Yes | 9849656622 |
| lasyavadala@gmail.com | 18D41A04M2 | V Lasya | ECE | Yes | 8301636405 |
| nimireddie@gmail.com | 17D41A04G1 | Nimisha Reddy | ECE | Yes | 8096005406 |
| dathupraneeth@gmail.com | 17D41A04M2 | TATAVARTHY SATYADATHA PRANEETH | ECE | Yes | 9959461749 |
| chilukurisaihiranmayi@gmail.com | 18D41A0439 | C. Sai hiranmayi | ECE | No | 8304531337 |
| vinishareddy1287@gmail.com | 19D41A04C6 | Vinisha | ECE | Yes | 7780578005 |
| bharadwajaenumula@gmail.com | 18D41A0466 | BHARADWAJA ENUMULA | ECE | No | 8886868873 |
| bgayathri0606@gmail.com | 17D41A0434 | Bommidi Gayathri | ECE | Yes | 8555088937 |
| gellaharini2000@gmail.com | 17D41A0474 | GELLA HARINI | ECE | Yes | 9705823567 |
| puneeshwaryerra@gmail.com | 17D41A04N5 | Yerra Puneeshwar | ECE | Yes | 9652885597 |
| tmraju1999@gmail.com | 17D41A04M1 | TANGELLA MALLAM RAJU | ECE | Yes | 9533362745 |
| ramadugupranathi@gmail.com | 18D41A04H7 | R.Pranathi | ECE | Yes | 9290902931 |
| preethichari001@gmail.com | 19D41A0405 | AERRAGINNELA PREETHI | ECE | Yes | 7095522853 |
| Saichanderrao72@gmail.com | 18D41A04N1 | VELGAPURI SAI CHANDER RAO | ECE | Yes | 9959496360 |
| salgutinikitha13@gmail.com | 18D45A0420 | Salguti Nikitha | ECE | Yes | 9381459699 |
| kolukolapallybikku688@gmail.com | 19D45A0405 | KOLKULAPALLY BIKKU | ECE | Yes | 7995049810 |
| Shivaramallenki2001@gmail.com | 19D41A0408 | ALLENKI SHIVA RAM | ECE | Yes | 9390392078 |
| guntivarun2000@gmail.com | 17d41a0489 | Gunti Varun | ECE | Yes | 7330752010 |
| bhavanareddy3568@gmail.com | 18D41A0420 | Bajjuri Bhavana reddy | ECE | Yes | 9381093568 |
| vickyhaldekar@gmail.com | 18D41A04G2 | Paspolla vikas | ECE | Yes | 7036476271 |
| pramodpadala260@gmail.com | 17D41A04G5 | PADALA PRAMOD | ECE | Yes | 9640546310 |
| tejaswinireddy0310@gmail.com | 17D41A0406 | Alugubelly Tejaswini | ECE | Yes | 9121096309 |
| chindhammahesh17@gmail.com | 18D45A0406 | CH. MAHESH | ECE | Yes | 9392368955 |
| anilnani1912@gmail.com | 18D41A04E2 | MOKTHALA ANIL | ECE | Yes | 8309415585 |
| gouds1712@gmail.com | 18D41A0481 | J SHIVA PRASAD | ECE | Yes | 9640442934 |
| sairohithk27@gmail.com | 18D41A04A0 | KODAMARTHI SAI ROHITH | ECE | Yes | 9959311982 |
| arthamsravani@gmail.com | 17D41A04N9 | Artham Sravani | ECE | Yes | 8106334278 |
| poojithachilukuri2000@gmail.com | 17D41A0442 | CHILUKURI POOJITHA | ECE | Yes | 8897342191 |
| manoharroyal123@gmail.com | 18D41A04H6 | R MANOHAR | ECE | Yes | 9703533038 |
| boggitirajeev@gmail.com | 18D41A0430 | Boggiti rajeev rathan | ECE | Yes | 9502019831 |
| oblue1661@gmail.com | 18D41A04K5 | Sharat Chandra | ECE | Yes | 9391102328 |

| | | | | | |
|---|---|---|---|---|---|
| nagasai313@gmail.com | 18D41A04J1 | REDROWTHU NAGA SAI SRAVAN | ECE | Yes | 9133227319 |
| bsowmya1890@gmail.com | 18D41A0429 | Sowmya Boddu | ECE | Yes | 9573931880 |
| bhikshapathiyadav1976@gmail.com | 19D41A0474 | KADARI BIXAPATHI | ECE | Yes | 9398424409 |
| manireddy2366@gmail.com | 17D41A04K6 | Sankepally Mani kumar reddy | ECE | Yes | 9059527286 |
| maanyaraj21@gmail.com | 18D41A04N5 | Yata Maanya Raj | ECE | Yes | 9490330204 |
| pallavinari.1999@gmail.com | 17D41A04F4 | NARAYANAPURAPU PALLAVI NARI | ECE | Yes | 8328354990 |
| mounikaborra411@gmail.com | 19D41A0434 | Borra Mounika | ECE | Yes | 8888272130 |
| shivanigoud688@gmail.com | 17D41A04G2 | P Shivani | ECE | Yes | 7675877888 |
| sahithireddy2929@gmail.com | 17D41A0457 | Depa Sahithi | ECE | Yes | 8801518275 |
| divyabolla12@gmail.com | 19D41A0432 | Bolla Divya | ECE | Yes | 9390057371 |
| shyamreddy6969@gmail.com | 19D41A0495 | KOTA MEGHA SHYAM REDDY | ECE | Yes | 9849536303 |
| manishgoud578@gmail.com | 18D41A04M0 | Tolupunoori Maneesh Goud | ECE | Yes | 9515741683 |
| saikumarrepala3@gmail.com | 17D41A04K2 | Repala Saikumar | ECE | Yes | 9703288155 |
| suppuminnu0418@gmail.com | 19D45A0408 | Bajjuri Supriya | ECE | Yes | 8309663385 |
| sailakshmanmadala@gmail.com | 18D41A04C3 | MADALA SAI LAKSHMAN | ECE | Yes | 8309089804 |
| himabindugoud0@gmail.com | 18D41A04J5 | ROLLA HIMA BINDU GOUD | ECE | Yes | 7386764612 |
| thummalaabhijithreddy@gmail.com | 18D41A04H4 | THUMMALA ABHIJITH REDDY | ECE | Yes | 7032222509 |
| gundagouthamreddy@gmail.com | 17D41A0486 | GUNDA GOUTHAM REDDY | ECE | Yes | 9700303599 |
| gundagouthamreddy@gmail.com | 17D41A0486 | GUNDA GOUTHAM REDDY | ECE | Yes | 9700303599 |
| dineshreddydornala@gmail.com | 17D41A0460 | Dornala Dinesh Reddy | ECE | Yes | 8309610587 |
| gopaldaspavanipavani@gmail.com | 19D41A0457 | Gopaldasu pavani | ECE | Yes | 9390865993 |
| nandinireinthala@gmail.com | 18D41A04J3 | REINTHALA NANDINI | ECE | Yes | 9381965084 |
| archanapuppireddy@gmail.com | 17D41A04H6 | Puppireddy Archana | ECE | Yes | 8309822602 |
| saikota246@gmail.com | 17D41A04B6 | Kota sai chandu | ECE | Yes | 9701098400 |
| Tejvardhan119@gmail.com | 18D41A0447 | CHILKUR TEJA VARDHAN REDDY | ECE | Yes | 8639426145 |
| etikyalaarunkumar@gmail.com | 18D45A0403 | ETIKYALA ARUNKUMAR | ECE | Yes | 7093052335 |
| vinayk8188@gmail.com | 18D41A0490 | KAMISETTY VINAY | ECE | No | 9110382539 |
| subhash143naik@gmail.com | 17D41A0402 | Jatavath Subhash Naik | ECE | Yes | 8309101503 |
| bharathkumarsaluvala003@gmail.com | 18D41A04J7 | SALUVALA BHARATH KUMAR | ECE | No | 7658935412 |
| sanjanabheemreddy21@gmail.com | 19D41A0427 | BHEEMREDDY SANJANA | ECE | Yes | 9491519071 |
| sindhunamani788@gmail.com | 19D41A04C9 | NAMANI SINDHU | ECE | Yes | 9515591426 |
| bsowmya1890@gmail.com | 18D41A0429 | Sowmya Boddu | ECE | Yes | 9573931880 |
| maheshbabukmb2000@gmail.com | 17D41A0496 | Kalavakuri Mahesh babu | ECE | Yes | 8309919419 |
| kartheekreddy811@gmail.com | 17D41A04A4 | Kathi Kartheek Reddy | ECE | Yes | 9.19909E+11 |
| tandrasanjanareddy4089@gmail.com | 17D41A04L8 | T sanjana | ECE | Yes | 9704420973 |
| muppidipramod619@gmail.com | 17D41A04E9 | Muppidi pramod | ECE | Yes | 8464085540 |
| jangiliraghavendra2@gmail.com | 18D45A0432 | JANGILI RAGHAVENDRA | ECE | Yes | 9100461432 |
| saiabhinav.ranga@gmail.com | 17D41A04H9 | R Sai Abhinav Goud | ECE | Yes | 8179239234 |
| arshsiddique191@gmail.com | 17D41A04H7 | Quazi Mohammad Abdul Raheem Siddique | ECE | Yes | 9553435774 |
| mosesabhishek191@gmail.com | 17D41A04N2 | Vemuri Moses Abhishek | ECE | Yes | 7036378749 |
| anemonishirisha1234@gmail.com | 19D41A0412 | Anemoni shirisha mudhiraj | ECE | Yes | 9390966285 |
| tkavya2000@gmail.com | 17D41A04L7 | T Kavya | ECE | Yes | 9948910614 |
| rahulrangaraj2000@gmail.com | 17D41A04J5 | Rangaraju Rahul | ECE | Yes | 7386052615 |
| saisridachepalli05@gmail.com | 19D41A0441 | Dachepally Saisri | ECE | Yes | 7995323325 |

| | | | | | |
|---|---|---|---|---|---|
| rahulnaikjaan@gmail.com | 19D41A04A0 | LUNAVATH RAHUL NAIK | ECE | Yes | 9177157542 |
| buchikundhasainath@gmail.com | 19D41A0435 | B.Sainath | ECE | Yes | 9542763976 |
| madhavshiva629@gmail.com | 17D41A04G9 | Pasunuri shiva | ECE | Yes | 8885902922 |
| gayatrisagar2001@gmail.com | 19D41A0436 | BUDURU GAYATRI | ECE | Yes | 7036429488 |
| amadhuri567@gmail.com | 19D41A0417 | Ashwala Madhuri | ECE | Yes | 9390425141 |
| bhukyamahesh7386@gmail.com | 19D41A04A3 | MAHESH BHUKYA | ECE | Yes | 7386770284 |
| Rudhinikareddychinna@gmail.com | 18D41A0451 | Rudhinika | ECE | Yes | 9121746888 |
| nimireddie@gmail.com | 17D41A04G1 | Nimisha Reddy | ECE | Yes | 8096005408 |
| dedeepyabeemagani@gmail.com | 19D41A0424 | BEEMAGANI DEDEEPYA | ECE | Yes | 9542013099 |
| pruthvikiran106@gmail.com | 17D41A04C5 | L.pruthvi kiran | ECE | Yes | 9398817764 |
| lasyareddykontham1999@gmail.com | 17D41A04B2 | KONTHAM LASYA REDDY | ECE | Yes | 8185075819 |
| manishkesidi@gmail.com | 18D45A0422 | KESIDI MANISH KUMAR | ECE | Yes | 7013394005 |
| snehasree591@gmail.com | 17D41A0436 | B. SNEHA | ECE | Yes | 8374569428 |
| sailakshmanmadala@gmail.com | 18D41A04C3 | MADALA SAI LAKSHMAN | ECE | Yes | 8303089804 |
| mdivyasree2804@gmail.com | 17D41A04E4 | MONDRU DIVYA SREE | ECE | Yes | 8247446606 |
| madhurireddy0311@gmail.com | 17D41A04C9 | MADHURI NERAVETLA | ECE | Yes | 9398164151 |

ENGG. & TECH. * SRI INDU COLLEGE OF

PRINCIPAL
Sri Indu College of Engineering and Technology
(Vill): SHERIGUDA-501 510,
Ibrahimpatnam(M), R.R.Dist.

# SRI INDU COLLEGE OF ENGINEERING AND TECHNOLOGY

## DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

HANDS ON TRAINING COURSE

ON

IMPLEMENTATION OF IMAGE PROCESSING CONCEPTS FOR REALTIME APPLICATIONS USING MATLAB

Date: From 19.09.2021 (6 Week Course, Only on Saturdays)

SHORTLISTED STUDENTS

ATTENDANCE SHEET

| S. No | Hall Ticket Number | NAME OF THE STUDENT | Week 1 | Week 2 | Week 3 | Week 4 | Week 5 | Week 6 |
|---|---|---|---|---|---|---|---|---|
| 1 | 18D41A04F9 | P.VIVEK VARDHAN REDDY | | | | | | |
| 2 | 18D41A04E0 | MOHAMMED ADNAN | | | | ← A → | | |
| 3 | 18D41A04F2 | NEELESH KUMAR GOUD | | | | | | |
| 4 | 18D41A04G0 | PARITALA GANESH | | | | | | |
| 5 | 18D41A04N8 | RANAVATH ABHISHEK RAJ NAIK | | | | | | |
| 6 | 18D41A04L5 | THALASANI VAMSHIDHAR REDDY | | | | | | |
| 7 | 18D41A04E9 | N.THARUN KUMAR REDDY | | | | | | |
| 8 | 18D41A04D7 | MASADI AJAY | | | | | | |
| 9 | 18D41A04G7 | PIRLA VENKATA SUDHEER | | | | | | ← A → |
| 10 | 17D41A0483 | GUDIPALLY.MANASA | | | | | | |
| 11 | 17D41A04J0 | RAMISHETTI VARUN KUMAR | | | | | | |
| 12 | 17D41A0408 | ANGOLI RISHWANTH GPUD | | ← A → | | | | |
| 13 | 18D41A04Q0 | PUNNA KARTHIK | | | | | | |
| 14 | 18D41A04H1 | AMRUTHA VARSHINI PONNA | | | | | | |
| 15 | 19D45A0418 | D INDRA KUMAR | | | | | | |
| 16 | 18D41A04H0 | POLOJU POORNA CHANDAR | | | | | | |
| 17 | 18D41A04P8 | VAISHNAVI | | | | | | |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 18 | 17D41A0483 | GUDIPALLY.MANASA | | | | | | | |
| 19 | 18D41A04J8 | SAMALA ROSHAN | | | | | | | |
| 20 | 18D41A04F3 | P UNESHCHANDRA | | | | | | | |
| 21 | 18D41A04M5 | VADYALA SATHWIKA | | | | | | | |
| 22 | 18D41A04C2 | M NISHITHA | | | | | | | |
| 23 | 18D41A04L3 | SUREPALLY.MOUNIKA | | | | | | | |
| 24 | 18D41A04L3 | SUREPALLY.MOUNIKA | | | | | | | |
| 25 | 18D41A0425 | BHAVANI YAMSANI | | | | | | | |
| | | Present → | 25/25 | 24/25 | 25/25 | 23/25 | 25/25 | 24/25 | |

1.

2.

**Coordinator**

**Convener**

**HOD/ECE**

Sri Indu
College of Engineering & Technology
UGC Autonomous Institution
Recognized under 2(f) & 12(B) of UGC Act 1956,
NAAC, Approved by AICTE &
Permanently Affiliated to JNTUH
Estd.2001

INSTITUTION'S INNOVATION COUNCIL
(Ministry of Education Initiative)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

HANDS ON TRAINING COURSE
ON

# SKETCH WITH ARDUINO

## STARTS ON January 2, 2022

In association with TLC

Registration : Free
Course Duration : 4 Week
Weekend Course (Saturday)
Invited Participants: Third Year ECE, EEE, CSE
Restricted to 30 Participants/Slot
Resource Persons: In-house Trainers


| Coordinators | Convener | Principal |
|---|---|---|
| Mr. E.Parasuramu | Prof.k.Ashok Babu | Dr.G.Suresh |
| 9989575859 | | |

# SRI INDU COLLEGE OF ENGINEERING AND TECHNOLOGY

## DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

HANDS ON TRAINING COURSE

ON

## Sketch with Arduino

Date: From 02.01.2022 (4 Week Course, Only on Saturday)

SHORTLISTED STUDENTS

**ATTENDANCE SHEET**

| S. No | Hall Ticket Number | NAME OF THE STUDENT | Week 1 | Week 2 | Week 3 | Week 4 |
|---|---|---|---|---|---|---|
| 1 | 18D41A0447 | Chilkur Teja Vardhan Reddy | Teja Vardhan | Teja Vardhan | (A) | Teja Vardhan |
| 2 | 18D45A0403 | Etikyala Arunkumar | | | | |
| 3 | 18D41A0490 | Kamisetty Vinay | | | | |
| 4 | 17D41A0402 | Jatavath Subhash Naik | Subhash | Subhash | Subhash | Subhash |
| 5 | 18D41A04J7 | Saluvala Bharath Kumar | Bharath Kumar | Bharath kumar | Bharath Kumar | Bharath Kumar |
| 6 | 19D41A0427 | Bheemreddy Sanjana | B. Sanjana | B. Sanjana | B. Sanjana | B. Sanjana |
| 7 | 19D41A04C9 | Namani Sindhu | N. Sindhu | N. Sindhu | N. Sindhu | N. Sindhu |
| 8 | 18D41A0429 | Sowmya Boddu | | (A) | | |
| 9 | 17D41A0496 | Kalavakuri Mahesh Babu | K. Mahesh babu | K. Mahesh Babu | K. Mahesh babu | K. Mahesh babu |
| 10 | 17D41A04A4 | Kathi Kartheek Reddy | Kartheek | Kartheek | Kartheek | Kartheek |
| 11 | 17D41A04L8 | T.Sanjana | | | | |
| 12 | 17D41A04E9 | Muppidi Pramod | Pramod | Pramod | Pramod | Pramod |
| 13 | 18D45A0432 | Jangili Raghavendra | | | | |
| 14 | 17D41A04H9 | R Sai Abhinav Goud | Sai Abhinav Goud | Sai Abhinav Goud | Sai Abhinav Goud | Sai Abhinav Goud |
| 15 | 17D41A04H7 | Quazi Mohammad Abdul Raheem Siddique | AbdR | AbdR | AbdR | AbdR |
| 16 | 17D41A04N2 | Vemuri Moses Abhishek | VABS | VABS | VABS | (A) |
| 17 | 19D41A0412 | Anemoni Shirisha Mudhiraj | A. Shirisha | A. Shirisha | A. Shirisha | A. Shirisha |

| | | | | | | |
|---|---|---|---|---|---|---|
| 18 | 17D41A04L7 | T Kavya | T. Kavya | T. Kavya | T. Kavya | T. Kavya |
| 19 | 17D41A04J5 | Rangaraju Rahul | Rangaraju | Rangaraju | Rangaraju | Rangaraju |
| 20 | 19D41A0441 | Dachepally Saisri | Sai sri | Sai sri | Sai sri | A |
| 21 | 19D41A04A0 | Lunavath Rahul Naik | | | | |
| 22 | 19D41A0435 | B.Sainath | Sainath | Sainath | Sainath | Sainath |
| 23 | 17D41A04G9 | Pasunuri Shiva | shiva | shiva | shiva | shiva |
| 24 | 19D41A0436 | Buduru Gayatri | Gayatri | Gayatri | Gayathri | Gayatri |
| 25 | 19D41A0417 | Ashwala Madhuri | madhuri | | | |
| 26 | 19D41A04A3 | Mahesh Bhukya | Mahesh | Mahesh | Mahesh | Mahesh |
| 27 | 18D41A0451 | Rudhinika | Rudhinika | Rudhinika | Rudhinika | Rudhinika |
| 28 | 17D41A04G1 | Nimisha Reddy | Nimisha Reddy | Nimisha Reddy | Nimisha Reddy | Nimish Reddy |
| 29 | 19D41A0424 | Beemagani Dedeepya | Dedeepya | Dedeepya | Dedeepya | Dedeepya |
| 30 | 17D41A04C5 | L.Pruthvi Kiran | Pruthvi Kiran | Pruthvi Kiran | Pruthvi Kiran | Pruthvi Kiran |
| 31 | 17D41A04B2 | Kontham Lasya Reddy | Lasya Reddy | Lasya Reddy | Lasya Reddy | Lasya Reddy |
| | | | (31) | (30) | (30) | (29) |

**Coordinator** **Convener** **HOD/ECE**

Sri Indu
College of Engineering & Technology
UGC Autonomous Institution
Recognized under 2(f) & 12(B) of UGC Act 1956.
NAAC, Approved by AICTE &
Permanently Affiliated to JNTUH
Estd.2001

INSTITUTION'S INNOVATION COUNCIL

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

HANDS ON TRAINING COURSE
ON

# SKETCH WITH ARDUINO

## STARTS ON January 2, 2022

In association with TLC

Registration : Free
Course Duration : 4 Week
Weekend Course (Saturday)
Invited Participants: Third Year ECE, EEE, CSE
Restricted to 30 Participants/Slot
Resource Persons: In-house Trainers

| Coordinators | Convener | Principal |
|---|---|---|
| Mr. E.Parasuramu<br>9989575859 | Prof.k.Ashok Babu | Dr.G.Suresh |


The easiest way for beginners to get started with Arduino is by creating circuits using a solderless breadboard. These simple projects will teach you the basics of Arduino Uno, electronics and programming. In this tutorial, you will be creating circuits using the following electronic components:

- LED
- RGB LED
- Temp Sensor
- Pushbutton
- Potentiometer
- Photoresistor
- Servo
- Motor
- Buzzer
- LCD screen

This tutorial is going to allow you to jump right in and start building circuits. If you need some background on the Arduino Uno board or the tools that are needed, please check out post – Arduino Uno For Beginners.

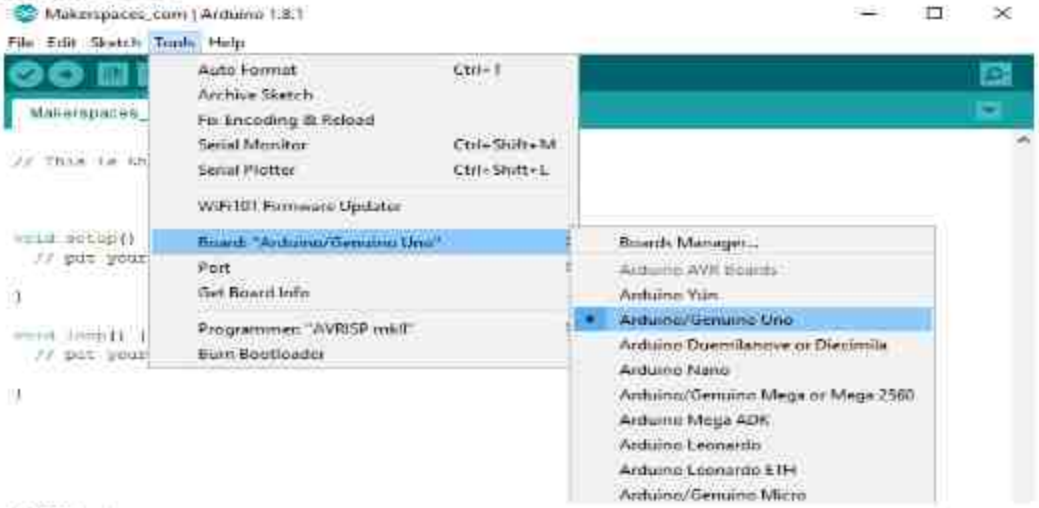
**Getting Started**

Before you can start working with Arduino, you need to make sure you have the IDE software installed on your computer. This program allows you to write, view and upload the code to your Arduino Uno board. You can download the IDE for free on Arduino's website.

Once the IDE is installed, you will need to connect your Arduino to your computer. To do this, plug one end of the USB cable to the Arduino Uno and then the other end of the USB to your computer's USB port.

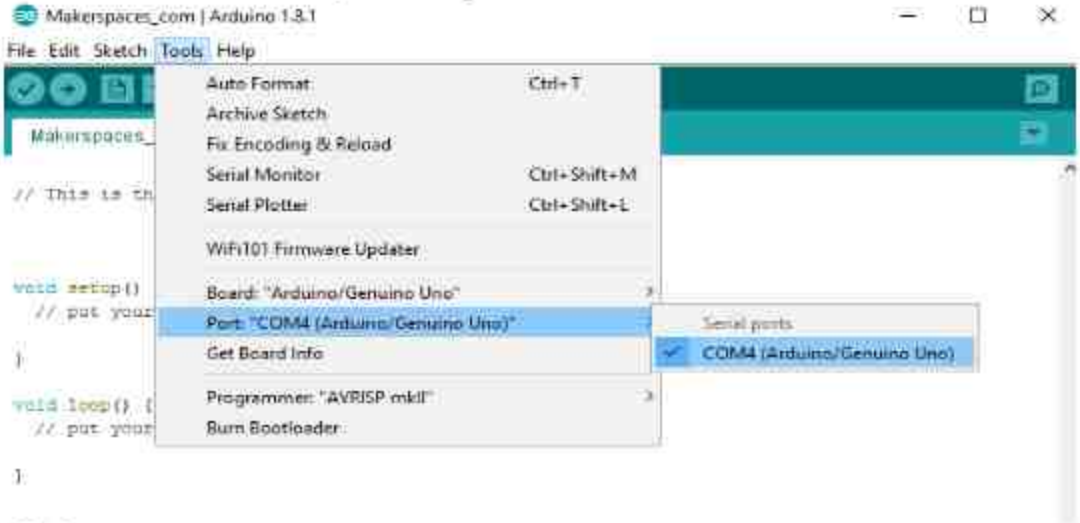
**Select The Board**

Once the board is plugged in, you will need to open the IDE and click on **Tools** > **Board** > **Arduino Uno** to select the board.




**Select Serial Port**

Next, you have to tell the Arduino which port you are using on your computer. To select the port, go to **Tools** > **Port** and then select the port that says **Arduino.**




**Project Code**

To complete the projects in this tutorial, you will need to download the project code which are known as sketches. A sketch is simply a set of instructions that tells the board what functions it needs to perform. For some of these projects, we are using open-source code that was released by the good people at Sparkfun and Arduino. Use the link below to download the zip folder containing the code.

Download Project Code – (ZIP File)

Once the file has been downloaded, you will need to unzip/extract the folder in order to use it.

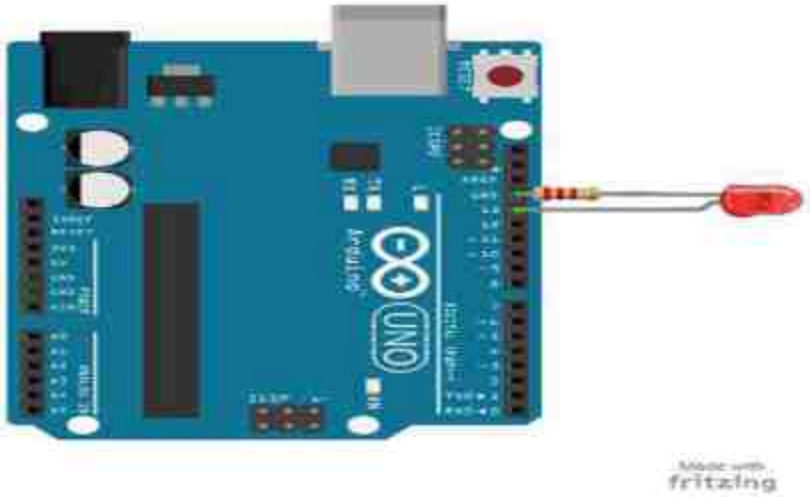
**#1 – Test Arduino**

The first project is one of the most basic and simple circuits you can create with Arduino. This project will test your Arduino by blinking an LED that is connected directly to the board.

**Parts Needed**

- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) LED 5mm
- (1) 220 Ω Resistor

**Project Diagram**




**Project Steps**

1. Twist a 220 Ω resistor to the long leg (+) of the LED.
2. Push the short leg of the LED into the ground (GND) pin on the board.
3. Push the resistor leg that's connected to the LED into the #13 pin.

**Project Code**

1. Connect the Arduino board to your computer using the USB cable.
2. Open project code – **Circuit_01_TestArduino**
3. Select the board and serial port as outlined in earlier section.
4. Click upload button to send sketch to the Arduino.

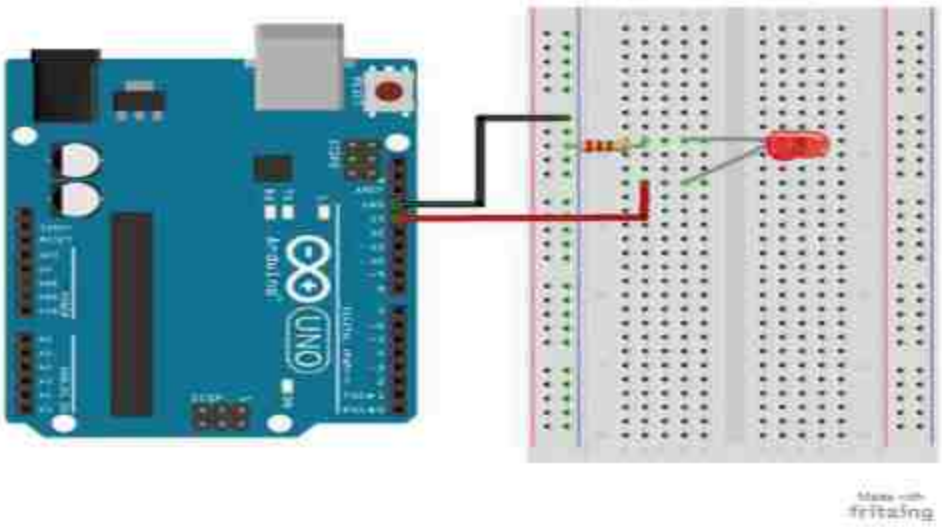
**#2 – Blink an LED**

This project is identical to project #1 except that we will be building it on a breadboard. Once complete, the LED should turn on for a second and then off for a second in a loop.

**Parts Needed**

- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) Breadboard – Half Size
- (1) LED 5mm
- (1) 220 Ω Resistor
- (2) Jumper Wires

**Project Diagram**



**Project Code**

1. Connect the Arduino board to your computer using the USB cable.
2. Open project code – **Circuit_02_Blink**
3. Select the board and serial port as outlined in earlier section.
4. Click upload button to send sketch to the Arduino.

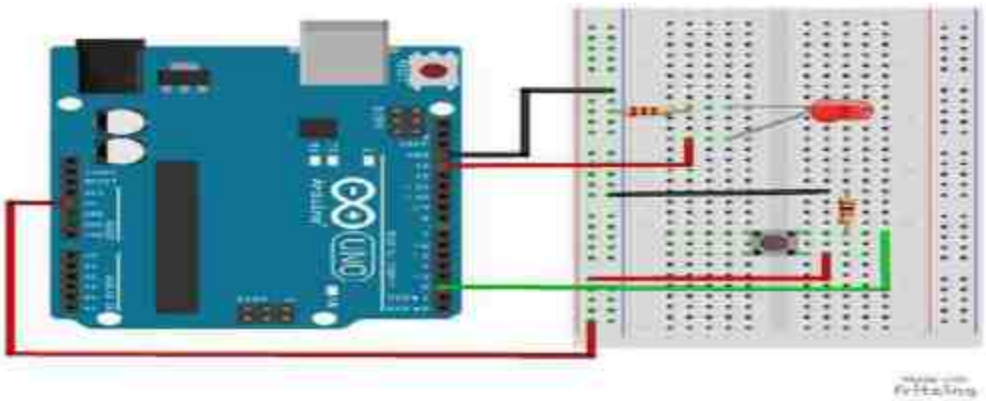
**#3 – Push Button**

Using a push button switch, you will be able to turn on and off an LED.

**Parts Needed**

- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) Breadboard – Half Size
- (1) LED 5mm
- (1) 220 Ω Resistor
- (1) 10K Ω Resistor
- (1) Push Button Switch
- (6) Jumper Wires

**Project Diagram**



**Project Code**

1. Connect the Arduino board to your computer using the USB cable.
2. Open project code – **Circuit_03_Pushbutton**

3. Select the board and serial port as outlined in earlier section.
4. Click upload button to send sketch to the Arduino.

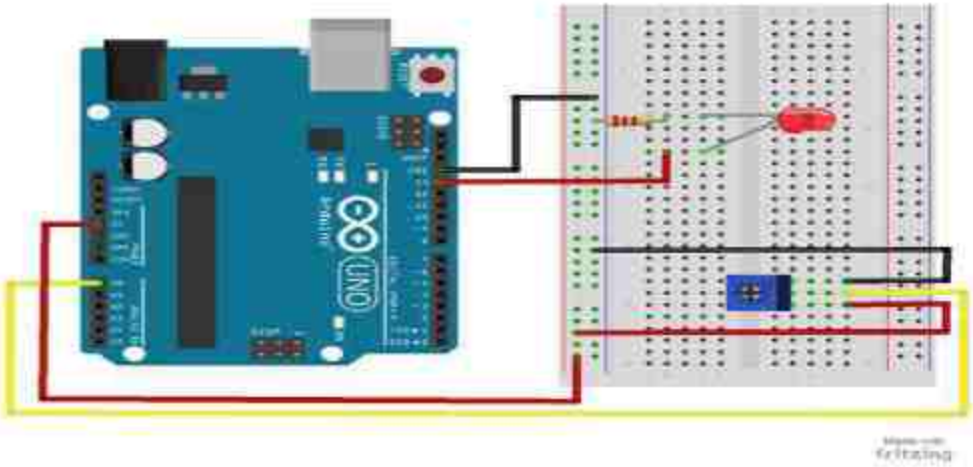
**#4 – Potentiometer**

Using a potentiometer, you will be able to control the resistance of an LED. Turning the knob will increase and decrease the frequency the LED blinks.

**Parts Needed**

- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) Breadboard – Half Size
- (1) LED 5mm
- (1) 220 Ω Resistor
- (1) Potentiometer (10k Trimpot)
- (6) Jumper Wires

**Project Diagram**



**Project Code**

1. Connect the Arduino board to your computer using the USB cable.
2. Open project code – **Circuit_04_Potentiometer**
3. Select the board and serial port as outlined in earlier section.
4. Click upload button to send sketch to the Arduino.

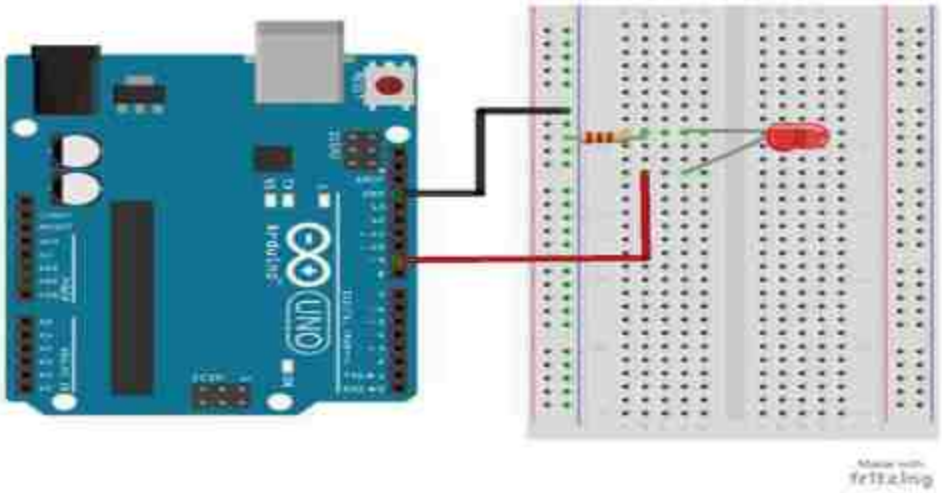
**#5 – Fade an LED**

By using a PWM pin on the Arduino, you will be able to increase and decrease the intensity of brightness of an LED.

**Parts Needed**

- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) Breadboard – Half Size
- (1) LED 5mm
- (1) 220 Ω Resistor
- (2) Jumper Wires

**Project Diagram**



**Project Code**

1. Connect the Arduino board to your computer using the USB cable.
2. Open project code – **Circuit_05_Fade**
3. Select the board and serial port as outlined in earlier section.
4. Click upload button to send sketch to the Arduino.

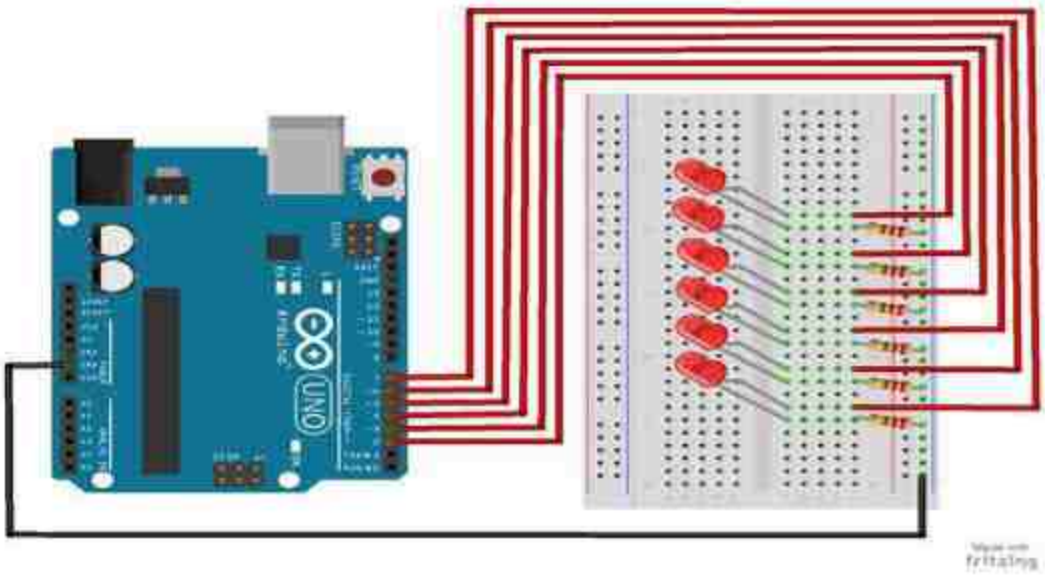
**#6 – Scrolling LED**

This project will blink 6 LEDs, one at a time, in a back and forth formation. This type of circuit was made famous by the show Knight Rider which featured a car with looping LEDs.

**Parts Needed**

- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) Breadboard – Half Size
- (6) LED 5mm
- (6) 220 Ω Resistor
- (7) Jumper Wires

**Project Diagram**



**Project Code**

1. Connect the Arduino board to your computer using the USB cable.
2. Open project code – **Circuit_06_Scrolling**
3. Select the board and serial port as outlined in earlier section.
4. Click upload button to send sketch to the Arduino.

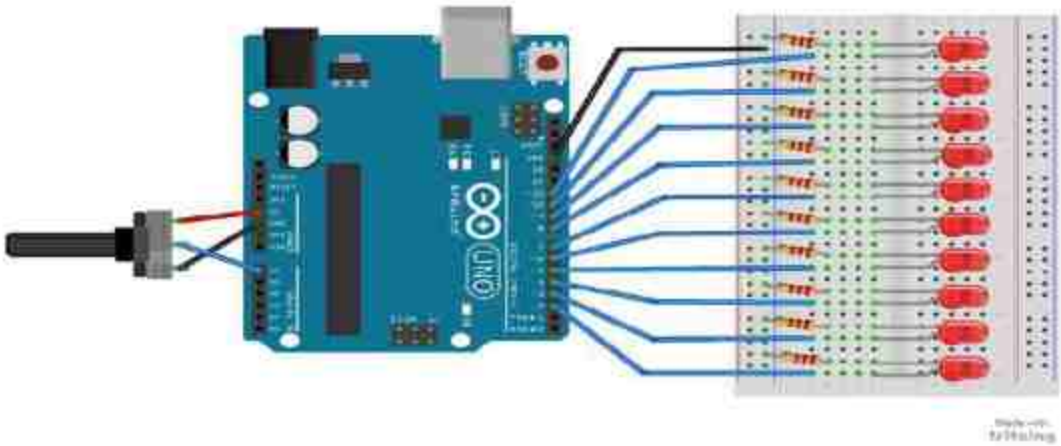
**#7 – Bar Graph**

Using a potentiometer, you can control a series of LEDs in a row. Turning the potentiometer knob will turn on or off more of the LEDs.

**Parts Needed**

- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) Breadboard – Half Size
- (1) Potentiometer – Rotary
- (10) LED 5mm
- (10) 220 Ω Resistor
- (11) Jumper Wires

**Project Diagram**



**Project Code**

1. Connect the Arduino board to your computer using the USB cable.
2. Open project code – **Circuit_07_BarGraph**
3. Select the board and serial port as outlined in earlier section.
4. Click upload button to send sketch to the Arduino.

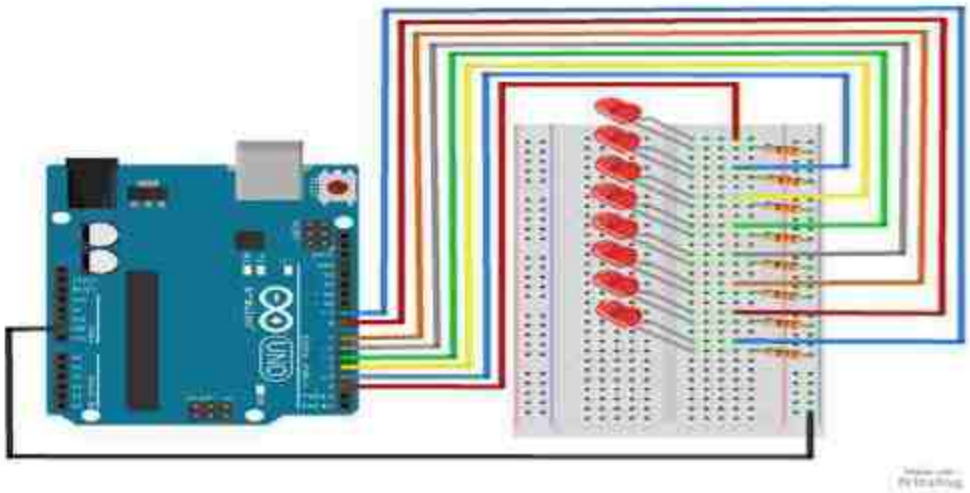
**#8 – Multiple LEDs**

This project will use 8 pins on the Arduino board to blink 8 LEDs at the same time.

**Parts Needed**

- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) Breadboard – Half Size
- (8) LED 5mm
- (8) 330 Ω Resistor
- (9) Jumper Wires

**Project Diagram**



**Project Code**

1. Connect the Arduino board to your computer using the USB cable.
2. Open project code – **Circuit_08_MultipleLEDs**
3. Select the board and serial port as outlined in earlier section.
4. Click upload button to send sketch to the Arduino.

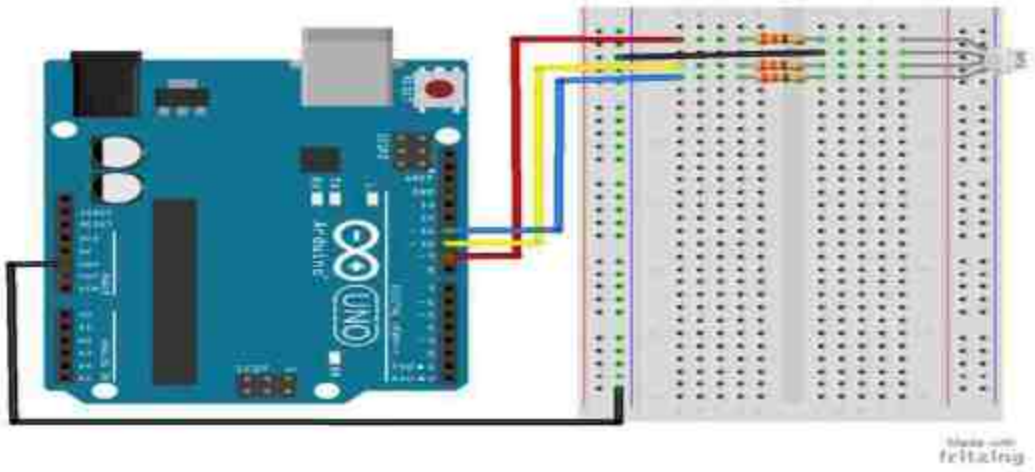
**#9 – RGB LED**

This project will be using an RGB LED to scroll through a variety of colors. RGB stands for Red, Green and Blue and this LED has the ability to create nearly unlimited color combinations.

**Parts Needed**

- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) Breadboard – Half Size
- (1) RGB LED
- (3) 330 Ω Resistor
- (5) Jumper Wires

**Project Diagram**



**Project Code**

1. Connect the Arduino board to your computer using the USB cable.
2. Open project code – **Circuit_09_RGBLED**

3. Select the board and serial port as outlined in earlier section.
4. Click upload button to send sketch to the Arduino.

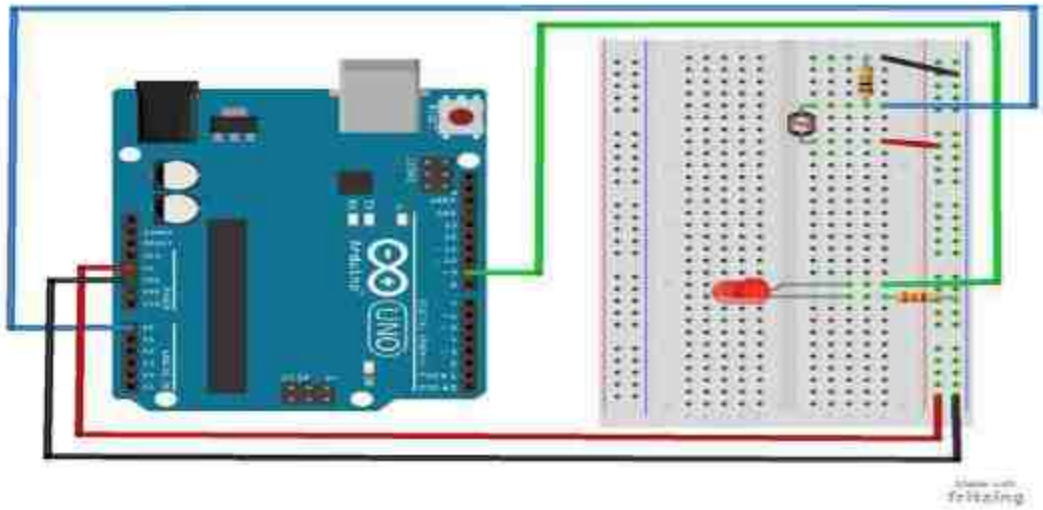
**#10 – Photoresistor**

A photoresistor changes the resistance a circuit gets based on the amount of light that hits the sensor. In this project, the brightness of the LED will increase and decrease based on the amount of light present.

**Parts Needed**

- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) Breadboard – Half Size
- (1) LED 5mm
- (1) 330 Ω Resistor
- (1) 10K Ω Resistor
- (1) Photoresistor
- (6) Jumper Wires

**Project Diagram**



**Project Code**

1. Connect the Arduino board to your computer using the USB cable.
2. Open project code – **Circuit_10_Photoresistor**
3. Select the board and serial port as outlined in earlier section.
4. Click upload button to send sketch to the Arduino.

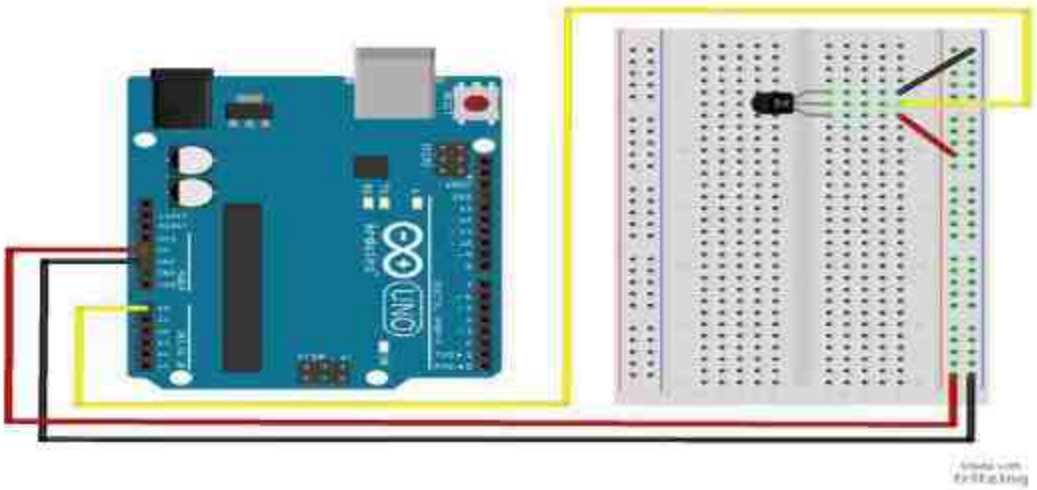
**#11 – Temp. Sensor**

A temperature sensor measures ambient temperatures of the world around it. In this project, we will be displaying the temperature in the serial monitor of the Arduino IDE.

**Parts Needed**

- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) Breadboard – Half Size
- (1) Temperature Sensor – TMP36
- (5) Jumper Wires

**Project Diagram**



**Project Code**

1. Connect the Arduino board to your computer using the USB cable.
2. Open project code – **Circuit_11_TempSensor**
3. Select the board and serial port as outlined in earlier section.
4. Click upload button to send sketch to the Arduino.

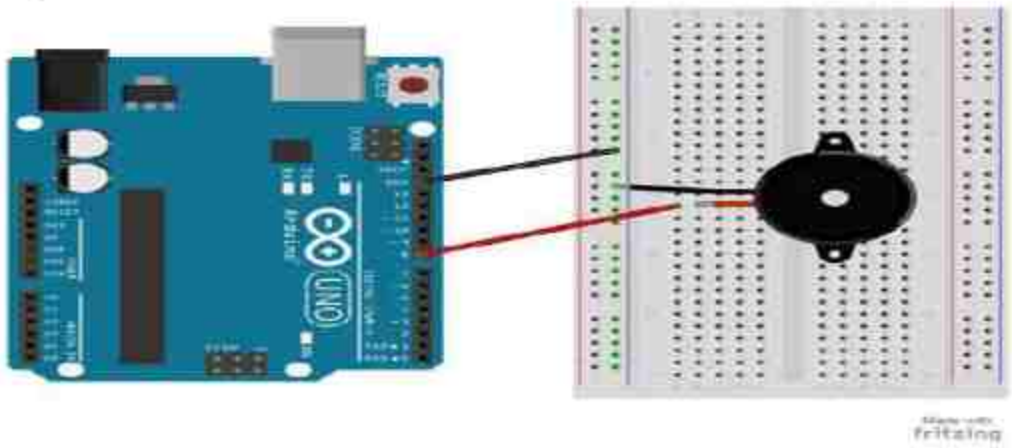
**#12 – Tone Melody**

The project will use a piezo buzzer/speaker to play a little melody.

**Parts Needed**

- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) Breadboard – Half Size
- (1) Piezo Buzzer/Speaker
- (2) Jumper Wires

**Project Diagram**



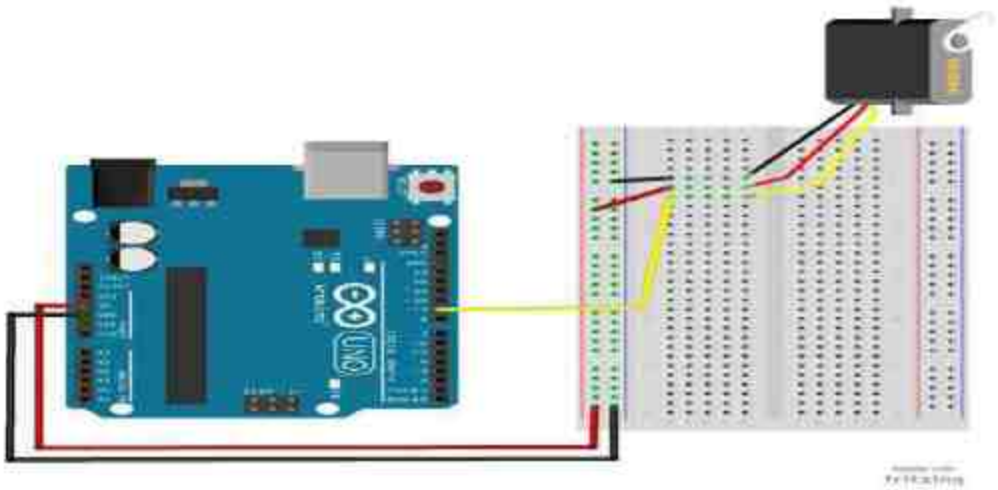
**Project Code**

1. Connect the Arduino board to your computer using the USB cable.
2. Open project code – **Circuit_12_ToneMelody**
3. Select the board and serial port as outlined in earlier section.
4. Click upload button to send sketch to the Arduino.

**#13 – Servo**
In this project, you will be able to sweep a servo back and forth through its full range of motion.

**Parts Needed**
- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) Breadboard – Half Size
- (1) Servo
- (6) Jumper Wires

**Project Diagram**

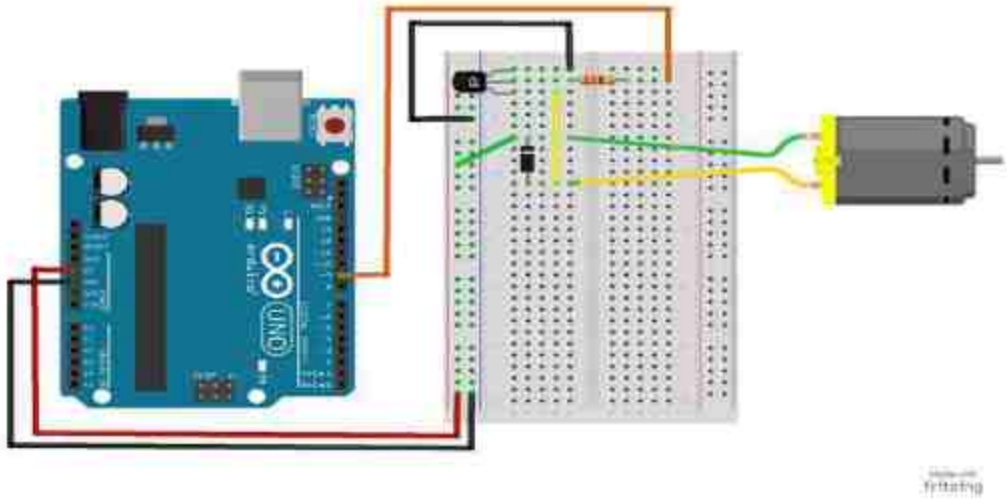


**Project Code**
1. Connect the Arduino board to your computer using the USB cable.
2. Open project code – **Circuit_13_Servo**
3. Select the board and serial port as outlined in earlier section.
4. Click upload button to send sketch to the Arduino.

**#14 – Motor**
Using a switching transistor, we will be able to control a DC motor. If everything is connected correctly, you should see the motor spinning.

**Parts Needed**
- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) Breadboard – Half Size
- (1) DC Motor
- (1) 330 Ω Resistor
- (1) Diode 1N4148
- (1) NPN Transistor
- (6) Jumper Wires

**Project Diagram**



**Project Code**

1. Connect the Arduino board to your computer using the USB cable.
2. Open project code – **Circuit_14_Motor**
3. Select the board and serial port as outlined in earlier section.
4. Click upload button to send sketch to the Arduino.

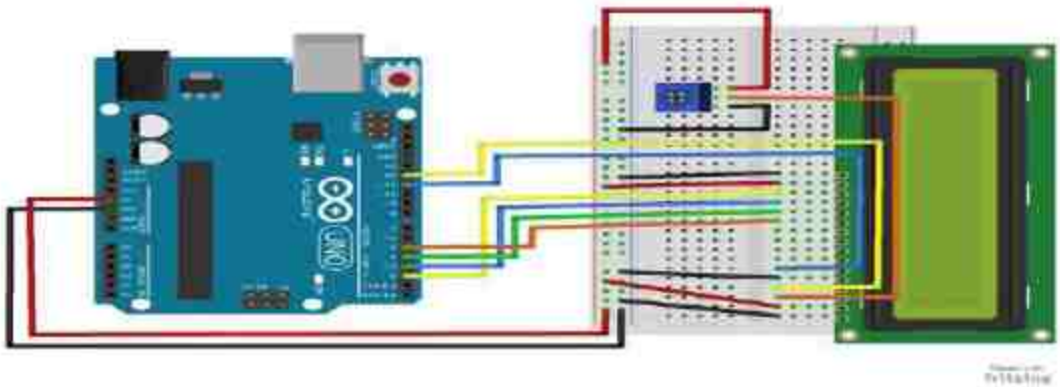
**#15 – LCD Screen**

An LCD is a liquid crystal display that is able to display text on its screen. In this project, you should see the words "hello,world!" displayed on the screen. The potentiometer is used to adjust the contrast of the display.

**Parts Needed**

- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) Breadboard – Half Size
- (1) LCD Screen
- (1) Potentiometer
- (16) Jumper Wires

**Project Diagram**



**Project Code**

1. Connect the Arduino board to your computer using the USB cable.
2. Open project code – **Circuit_15_LCD**
3. Select the board and serial port as outlined in earlier section.
4. Click upload button to send sketch to the Arduino.

**Troubleshooting**

- Make sure your board and serial port is selected in the IDE. To do this, plug your board in and go to **Tools** > **Board** >**Arduino** to select your board. Next, go to **Tools** > **Port** >**Com (Arduino)** to select your serial port.
- The long leg of the LED is the (+) positive and the short leg is the (-) negative. Make sure the correct leg of the LED is in the proper pin of the Arduino or breadboard as directed.
- It can be easy to put a component or jumper into the wrong pin on the Arduino or the breadboard. Double check the correct pin is being used.

**Experiment 1:** Turn an LED

Turn an LED on for one second, off for one second, and repeat forever.

```
void setup()
{
 pinMode(13, OUTPUT);
}

void loop()
{
 digitalWrite(13, HIGH);   // Turn on the LED
 delay(1000);              // Wait for one second
 digitalWrite(13, LOW);    // Turn off the LED
 delay(1000);              // Wait for one second
}
/*
```

**Experiment 2:** Turns on and off LED

Turns on and off a light emitting diode(LED) connected to digital
pin 13, when pressing a pushbutton attached to pin 2.

The circuit:
* LED attached from pin 13 to ground
* pushbutton attached to pin 2 from +5V
* 10K resistor attached to pin 2 from ground

* Note: on most Arduinos there is already an LED on the board
attached to pin 13.

```
// set pin numbers:
const int buttonPin = 2;     // the number of the pushbutton pin
const int ledPin =  13;      // the number of the LED pin

// variables will change:
int buttonState = 0;         // variable for reading the pushbutton status

void setup() {
  // initialize the LED pin as an output:
```

```
  pinMode(ledPin, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
}

void loop() {
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);

  // check if the pushbutton is pressed.
  // if it is, the buttonState is HIGH:
  if (buttonState == HIGH) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
  } else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
  }
}
```

**Experiment 3: Display RGB LED**

```
const int RED_PIN = 9;
const int GREEN_PIN = 10;
const int BLUE_PIN = 11;

const int DISPLAY_TIME = 1000;  // used in mainColors() to determine the
// length of time each color is displayed.

void setup()      //Configure the Arduino pins to be outputs to drive the LEDs
{
  pinMode(RED_PIN, OUTPUT);
  pinMode(GREEN_PIN, OUTPUT);
  pinMode(BLUE_PIN, OUTPUT);
}

void loop()
{
  mainColors();       // Red, Green, Blue, Yellow, Cyan, Purple, White
  //  showSpectrum();   // Gradual fade from Red to Green to Blue to Red
}

/**************************************************************************
 * void mainColors()
 * This function displays the eight "main" colors that the RGB LED
 * can produce. If you'd like to use one of these colors in your
 * own sketch, you can copy and paste that section into your code.
 **************************************************************************/
void mainColors()
{
  // all LEDs off
  digitalWrite(RED_PIN, LOW);
  digitalWrite(GREEN_PIN, LOW);
```

```
  digitalWrite(BLUE_PIN, LOW);
  delay(DISPLAY_TIME);

  // Red
  digitalWrite(RED_PIN, HIGH);
  digitalWrite(GREEN_PIN, LOW);
  digitalWrite(BLUE_PIN, LOW);
  delay(DISPLAY_TIME);

  // Green
  digitalWrite(RED_PIN, LOW);
  digitalWrite(GREEN_PIN, HIGH);
  digitalWrite(BLUE_PIN, LOW);
  delay(DISPLAY_TIME);

  // Blue
  digitalWrite(RED_PIN, LOW);
  digitalWrite(GREEN_PIN, LOW);
  digitalWrite(BLUE_PIN, HIGH);
  delay(DISPLAY_TIME);

  // Yellow (Red and Green)
  digitalWrite(RED_PIN, HIGH);
  digitalWrite(GREEN_PIN, HIGH);
  digitalWrite(BLUE_PIN, LOW);
  delay(DISPLAY_TIME);

  // Cyan (Green and Blue)
  digitalWrite(RED_PIN, LOW);
  digitalWrite(GREEN_PIN, HIGH);
  digitalWrite(BLUE_PIN, HIGH);
  delay(DISPLAY_TIME);

  // Purple (Red and Blue)
  digitalWrite(RED_PIN, HIGH);
  digitalWrite(GREEN_PIN, LOW);
  digitalWrite(BLUE_PIN, HIGH);
  delay(DISPLAY_TIME);

  // White (turn all the LEDs on)
  digitalWrite(RED_PIN, HIGH);
  digitalWrite(GREEN_PIN, HIGH);
  digitalWrite(BLUE_PIN, HIGH);
  delay(DISPLAY_TIME);
}

/*****************************************************************************
* void showSpectrum()
*
* Steps through all the colors of the RGB LED, displaying a rainbow.
* showSpectrum() calls a function RGB(int color) that translates a number
* from 0 to 767 where 0 = all RED, 767 = all RED
*
```

```
* Breaking down tasks down into individual functions like this
* makes your code easier to follow, and it allows.
* parts of your code to be re-used.
*****************************************************************/

void showSpectrum()
{
 for (int x = 0; x <= 767; x++)
 {
  RGB(x);      // Increment x and call RGB() to progress through colors.
  delay(10);   // Delay for 10 ms (1/100th of a second) - to help the "smoothing"
 }
}

/*****************************************************************
* void RGB(int color)
*
* RGB(###) displays a single color on the RGB LED.
* Call RGB(###) with the number of a color you want
* to display. For example, RGB(0) displays pure RED, RGB(255)
* displays pure green.
*
* This function translates a number between 0 and 767 into a
* specific color on the RGB LED. If you have this number count
* through the whole range (0 to 767), the LED will smoothly
* change color through the entire spectrum.
*
* The "base" numbers are:
* 0   = pure red
* 255 = pure green
* 511 = pure blue
* 767 = pure red (again)
*
* Numbers between the above colors will create blends. For
* example, 640 is midway between 512 (pure blue) and 767
* (pure red). It will give you a 50/50 mix of blue and red,
* resulting in purple.
*****************************************************************/
void RGB(int color)
{
 int redIntensity;
 int greenIntensity;
 int blueIntensity;

 color = constrain(color, 0, 767); // constrain the input value to a range of values from 0 to 767

  // if statement breaks down the "color" into three ranges:
 if (color <= 255)       // RANGE 1 (0 - 255) - red to green
 {
  redIntensity = 255 - color;    // red goes from on to off
  greenIntensity = color;        // green goes from off to on
  blueIntensity = 0;             // blue is always off
 }
```

```
 else if (color <= 511)  // RANGE 2 (256 - 511) - green to blue
 {
  redIntensity = 0;                  // red is always off
  greenIntensity = 511 - color;      // green on to off
  blueIntensity = color - 256;       // blue off to on
 }
 else                 // RANGE 3 ( >= 512)- blue to red
 {
  redIntensity = color - 512;        // red off to on
  greenIntensity = 0;                // green is always off
  blueIntensity = 767 - color;       // blue on to off
 }

 // "send" intensity values to the Red, Green, Blue Pins using analogWrite()
 analogWrite(RED_PIN, redIntensity);
 analogWrite(GREEN_PIN, greenIntensity);
 analogWrite(BLUE_PIN, blueIntensity);
}
```

**Experiment 4: Dancing LED**

```
int ledPins[] = {2,3,4,5,6,7,8,9};   // Defines an array to store the pin numbers of the 8 LEDs.
// An array is like a list variable that can store multiple numbers.
// Arrays are referenced or "indexed" with a number in the brackets [ ]. See the examples in
// the pinMode() functions below.

void setup()
{
 // setup all 8 pins as OUTPUT - notice that the list is "indexed" with a base of 0.
 pinMode(ledPins[0],OUTPUT);  // ledPins[0] = 2
 pinMode(ledPins[1],OUTPUT);  // ledPins[1] = 3
 pinMode(ledPins[2],OUTPUT);  // ledPins[2] = 4
 pinMode(ledPins[3],OUTPUT);  // ledPins[3] = 5
 pinMode(ledPins[4],OUTPUT);  // ledPins[4] = 6
 pinMode(ledPins[5],OUTPUT);  // ledPins[5] = 7
 pinMode(ledPins[6],OUTPUT);  // ledPins[6] = 8
 pinMode(ledPins[7],OUTPUT);  // ledPins[7] = 9
}

void loop()
{

 // This loop() calls functions that we've written further below.
 // We've disabled some of these by commenting them out (putting
 // "//" in front of them). To try different LED displays, remove
 // the "//" in front of the ones you'd like to run, and add "//"
 // in front of those you don't to comment out (and disable) those
 // lines.

 oneAfterAnother();  // Light up all the LEDs in turn

 //oneOnAtATime();       // Turn on one LED at a time
```

```
  //pingPong();           // Same as oneOnAtATime() but change direction once LED reaches edge

  //marquee();           // Chase lights like you see on theater signs

  //randomLED();          // Blink LEDs randomly
}


/**************************************************************************
 * oneAfterAnother()
 *
 * This function turns all the LEDs on, pauses, and then turns all
 * the LEDS off. The function takes advantage of for() loops and
 * the array to do this with minimal typing.
 **************************************************************************/
void oneAfterAnother()
{
  int index;
  int delayTime = 100; // milliseconds to pause between LEDs
                       // make this smaller for faster switching

  // Turn all the LEDs on:
  for(index = 0; index <= 7; index = ++index)  // step through index from 0 to 7
  {
    digitalWrite(ledPins[index], HIGH);
    delay(delayTime);
  }

  // Turn all the LEDs off:
  for(index = 7; index >= 0; index = --index)  // step through index from 7 to 0
  {
    digitalWrite(ledPins[index], LOW);
    delay(delayTime);
  }
}

/**************************************************************************
 * oneOnAtATime()
 *
 * This function will step through the LEDs, lighting only one at
 * a time. It turns each LED ON and then OFF before going to the
 * next LED.
 **************************************************************************/

void oneOnAtATime()
{
  int index;
  int delayTime = 100; // milliseconds to pause between LEDs
                       // make this smaller for faster switching

  for(index = 0; index <= 7; index = ++index)   // step through the LEDs, from 0 to 7
  {
```

```
  digitalWrite(ledPins[index], HIGH);  // turn LED on
  delay(delayTime);                    // pause to slow down
  digitalWrite(ledPins[index], LOW);   // turn LED off
 }
}

/**************************************************************************
 * pingPong()
 *
 * This function will step through the LEDs, lighting one at at
 * time in both directions. There is no delay between the LED off
 * and turning on the next LED. This creates a smooth pattern for
 * the LED pattern.
 **************************************************************************/
void pingPong()
{
 int index;
 int delayTime = 100; // milliseconds to pause between LEDs

 for(index = 0; index <= 7; index = ++index)   // step through the LEDs, from 0 to 7
 {
  digitalWrite(ledPins[index], HIGH);  // turn LED on
  delay(delayTime);                    // pause to slow down
  digitalWrite(ledPins[index], LOW);   // turn LED off
 }

 for(index = 7; index >= 0; index = --index)   // step through the LEDs, from 7 to 0
 {
  digitalWrite(ledPins[index], HIGH);  // turn LED on
  delay(delayTime);                    // pause to slow down
  digitalWrite(ledPins[index], LOW);   // turn LED off
 }
}

/**************************************************************************
 * marquee()
 *
 * This function will mimic "chase lights" like those around
 * theater signs.
 **************************************************************************/
void marquee()
{
 int index;
 int delayTime = 200; // milliseconds to pause between LEDs

 // Step through the first four LEDs
 // (We'll light up one in the lower 4 and one in the upper 4)

 for(index = 0; index <= 3; index++) // Step from 0 to 3
 {
  digitalWrite(ledPins[index], HIGH);    // Turn a LED on
  digitalWrite(ledPins[index+4], HIGH);  // Skip four, and turn that LED on
  delay(delayTime);                      // Pause to slow down the sequence
```

```
    digitalWrite(ledPins[index], LOW);    // Turn the LED off
    digitalWrite(ledPins[index+4], LOW);  // Skip four, and turn that LED off
  }
}

/**********************************************************************
 * randomLED()
 *
 * This function will turn on random LEDs. Can you modify it so it
 * also lights them for random times?
 **********************************************************************/
void randomLED()
{
  int index;
  int delayTime;

  index = random(8);  // pick a random number between 0 and 7
  delayTime = 100;

  digitalWrite(ledPins[index], HIGH);  // turn LED on
  delay(delayTime);                    // pause to slow down
  digitalWrite(ledPins[index], LOW);   // turn LED off
}
```

**Experiment 5: Running Motor**

```
const int motorPin = 9;  // Connect the base of the transistor to pin 9.
                         // Even though it's not directly connected to the motor,
                         // we'll call it the 'motorPin'

void setup()
{
  pinMode(motorPin, OUTPUT);  // set up the pin as an OUTPUT
  Serial.begin(9600);         // initialize Serial communications
}


void loop()
{ // This example basically replicates a blink, but with the motorPin instead.
  int onTime = 3000;  // milliseconds to turn the motor on
  int offTime = 3000; // milliseconds to turn the motor off

  analogWrite(motorPin, 255); // turn the motor on (full speed)
  delay(onTime);              // delay for onTime milliseconds
  analogWrite(motorPin, 0);   // turn the motor off
  delay(offTime);             // delay for offTime milliseconds

  // Uncomment the functions below by taking out the //. Look below for the
  // code examples or documentation.

  // speedUpandDown();
  // serialSpeed();
}
```

```
// This function accelerates the motor to full speed,
// then decelerates back down to a stop.
void speedUpandDown()
{
 int speed;
 int delayTime = 20; // milliseconds between each speed step

 // accelerate the motor
 for(speed = 0; speed <= 255; speed++)
 {
  analogWrite(motorPin,speed);          // set the new speed
  delay(delayTime);                     // delay between speed steps
 }
 // decelerate the motor
 for(speed = 255; speed >= 0; speed--)
 {
  analogWrite(motorPin,speed);          // set the new speed
  delay(delayTime);                     // delay between speed steps
 }
}

// Input a speed from 0-255 over the Serial port
void serialSpeed()
{
 int speed;

 Serial.println("Type a speed (0-255) into the box above,");
 Serial.println("then click [send] or press [return]");
 Serial.println();  // Print a blank line

 // In order to type out the above message only once,
 // we'll run the rest of this function in an infinite loop:

 while(true)  // "true" is always true, so this will loop forever.
 {
  // Check to see if incoming data is available:
  while (Serial.available() > 0)
  {
   speed = Serial.parseInt();  // parseInt() reads in the first integer value from the Serial Monitor.
   speed = constrain(speed, 0, 255); // constrains the speed between 0 and 255
                                     // because analogWrite() only works in this range.
   Serial.print("Setting speed to ");  // feedback and prints out the speed that you entered.
   Serial.println(speed);

   analogWrite(motorPin, speed);  // sets the speed of the motor.
  }
 }
}
```

**Experiment 6: Potentiometer**

```
int sensorPin = A0;    // select the input pin for the potentiometer
```

```
int ledPin = 13;      // select the pin for the LED
int sensorValue = 0;  // variable to store the value coming from the sensor

void setup() {
  // declare the ledPin as an OUTPUT:
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // read the value from the sensor:
  sensorValue = analogRead(sensorPin);
  // turn the ledPin on
  digitalWrite(ledPin, HIGH);
  // stop the program for <sensorValue> milliseconds:
  delay(sensorValue);
  // turn the ledPin off:
  digitalWrite(ledPin, LOW);
  // stop the program for for <sensorValue> milliseconds:
  delay(sensorValue);
}
```

**Experiment 7: Scrolling LED**

```
int timer = 100;          // The higher the number, the slower the timing.

void setup() {
  // use a for loop to initialize each pin as an output:
  for (int thisPin = 2; thisPin < 8; thisPin++) {
    pinMode(thisPin, OUTPUT);
  }
}

void loop() {
  // loop from the lowest pin to the highest:
  for (int thisPin = 2; thisPin < 8; thisPin++) {
    // turn the pin on:
    digitalWrite(thisPin, HIGH);
    delay(timer);
    // turn the pin off:
    digitalWrite(thisPin, LOW);
  }

  // loop from the highest pin to the lowest:
  for (int thisPin = 7; thisPin >= 2; thisPin--) {
    // turn the pin on:
    digitalWrite(thisPin, HIGH);
    delay(timer);
    // turn the pin off:
    digitalWrite(thisPin, LOW);
  }
}
```

**Experiment 8: Potentiometer**

```
int sensorPin = A0;    // select the input pin for the potentiometer
int ledPin = 13;      // select the pin for the LED
int sensorValue = 0;  // variable to store the value coming from the sensor

void setup() {
  // declare the ledPin as an OUTPUT:
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // read the value from the sensor:
  sensorValue = analogRead(sensorPin);
  // turn the ledPin on
  digitalWrite(ledPin, HIGH);
  // stop the program for <sensorValue> milliseconds:
  delay(sensorValue);
  // turn the ledPin off:
  digitalWrite(ledPin, LOW);
  // stop the program for for <sensorValue> milliseconds:
  delay(sensorValue);
}
```

**Experiment 9: LED with PWM**

```
int led = 9;           // the PWM pin the LED is attached to
int brightness = 0;    // how bright the LED is
int fadeAmount = 5;    // how many points to fade the LED by

// the setup routine runs once when you press reset:
void setup() {
  // declare pin 9 to be an output:
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  // set the brightness of pin 9:
  analogWrite(led, brightness);

  // change the brightness for next time through the loop:
  brightness = brightness + fadeAmount;

  // reverse the direction of the fading at the ends of the fade:
  if (brightness <= 0 || brightness >= 255) {
    fadeAmount = -fadeAmount;
  }
  // wait for 30 milliseconds to see the dimming effect
  delay(30);
}
```

```
Experiment 10: To measure the temperature sensor's
// signal pin.

const int temperaturePin = A0;

void setup()
{

        Serial.begin(9600); //Initialize serial port & set baud rate to 9600 bits per second (bps)
}

void loop()
{

        float voltage, degreesC, degreesF; //Declare 3 floating point variables

        voltage = getVoltage(temperaturePin); //Measure the voltage at the analog pin

        degreesC = (voltage - 0.5) * 100.0; // Convert the voltage to degrees Celsius

        degreesF = degreesC * (9.0 / 5.0) + 32.0; //Convert degrees Celsius to Fahrenheit

        //Now print to the Serial monitor. Remember the baud must be 9600 on your monitor!
        // These statements will print lines of data like this:
        // "voltage: 0.73 deg C: 22.75 deg F: 72.96"

        Serial.print("voltage: ");
        Serial.print(voltage);
        Serial.print("  deg C: ");
        Serial.print(degreesC);
        Serial.print("  deg F: ");
        Serial.println(degreesF);

        delay(1000); // repeat once per second (change as you wish!)
}
float getVoltage(int pin)          //Function to read and return
                                                        //floating-point value (true voltage)
                                                        //on analog pin
{

        return (analogRead(pin) * 0.004882814);
        // This equation converts the 0 to 1023 value that analogRead()
        // returns, into a 0.0 to 5.0 value that is the true voltage
        // being read at that pin.
}

// Other things to try with this code:

//   Turn on an LED if the temperature is above or below a value.

//   Read that threshold value from a potentiometer - now you've
//   created a thermostat!
```

Sri Indu
College of Engineering & Technology
UGC Autonomous Institution
Recognized under 2(f) & 12(B) of UGC Act 1956,
NAAC, Approved by AICTE &
Permanently Affiliated to JNTUH
Estd.2001

INSTITUTION'S INNOVATION COUNCIL
(Ministry of Education Initiative)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

# HANDS ON TRAINING COURSE ON IMAGE RETRIEVAL PROCESS USING MATLAB

## STARTS ON February 13, 2021

Registration : Free

Course Duration : 4 Week

Weekend Course (Saturday)

Invited Participants: Third Year ECE, EEE, CSE

Restricted to 30 Participants/Slot

Resource Persons: In-house Trainers

| Coordinators | Convener | Principal |
|---|---|---|
| Dr.P.Epsiba | Prof.k.Ashok Babu | Dr.G.Suresh |

## SRI INDU COLLEGE OF ENGINEERING AND TECHNOLOGY

### DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

HANDS ON TRAINING COURSE

ON

# IMAGE RETRIEVAL PROCESS USING MATLAB

Date: From 13.02.2021 (4 Week Course, Only on Saturdays)

SHORTLISTED STUDENTS

ATTENDANCE SHEET

| S. No | Hall Ticket Number | NAME OF THE STUDENT | Week 1 | Week 2 | Week 3 | Week 4 |
|---|---|---|---|---|---|---|
| 1 | 18D41A04C5 | Madduri Venkata Sameer Kumar | | | | |
| 2 | 18D41A04P5 | Madhavaram Sriram | | | | |
| 3 | 18D41A0447 | D Anil Kumar | | | | |
| 4 | 19D41A0492 | Koppula Rajitha | | | | |
| 5 | 18D41A04D3 | Manda Sushma | | | | |
| 6 | 18D41A04C1 | Kunchala Venkatesh | | | | |
| 7 | 18D41A04C5 | Madduri Venkata Sameer Kumar | | | | |
| 8 | 18D41A04M2 | V.Lasya | | | | |
| 9 | 18D41A04G1 | Nimisha Reddy | | | | |
| 10 | 18D41A04M2 | Tatavarthy Satyadatha Praneeth | | | | |
| 11 | 18D41A0439 | C. Sai Hiranmayi | | | | |
| 12 | 19D41A04C5 | Vinisha | | | | |
| 13 | 18D41A0468 | Bharadwaja Enumula | | | | |
| 14 | 18D41A0434 | Bommidi Gayathri | | | | |
| 15 | 18D41A0474 | Gella Harini | | | | |
| 16 | 18D41A04N5 | Yerra Puneeshwar | | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| 17 | 18D41A04M1 | Tangella Mallam Raju | Mallam Raju | Mallam Raju | Mallam Raju | Mallam Raju. |
| 18 | 18D41A04H7 | R.Pranathi | R. Pranathi | P. Pranathi | P. Pranathi | P. Pranathi |
| 19 | 19D41A0405 | Aerraginnela Preethi | A. Preethi | A. Preethi | A. Preethi | A. Preethi |
| 20 | 18D41A04N1 | Velgapuri Sai Chander Rao | | | | |
| 21 | 18D45A0420 | Salguti Nikitha | Nikitha | Nikitha | Nikitha | Nikitha |
| 22 | 19D45A0406 | Kolkulapally Bikku | K. Bikku | K. Bikku | ← A → | K. Bikku |
| 23 | 19D41A0408 | Allenki Shiva Ram | Shiva Ram | Shiva Ram | Shiva Ram | Shiva Ram |
| 24 | 18D41A0489 | Gunti Varun | Gunti Varun | Gunti Varun | Gunti Varun | Gunti Varun |
| 25 | 18D41A0420 | Bajjuri Bhavana Reddy | (A) | Bhavana Reddy | Bhavana Reddy | Bhavana Reddy |
| 26 | 18D41A04G2 | Paspolla Vikas | Vikas | Vikas | Vikas | Vikas |
| 27 | 18D41A04G5 | Padala Pramod | P. Pramod | P. Pramod | P. Pramod | P. Pramod |
| 28 | 18D41A0406 | Alugubelly.Tejaswini | Tejaswini | Tejaswini | Tejaswini | Tejaswini. |
| 29 | 18D45A0406 | Ch. Mahesh | ch. Mahesh | ch. Mahesh | ch. Mahesh | ch. Mahesh |
| 30 | 18D41A04E2 | Mokthala Anil | Anil | Anil | Anil | Anil |
| 31 | 18D41A0481 | J Shiva Prasad | J. Shiva | J. Shiva | J. Shiva | J. Shiva |
| | | | 29 | 31 | 30 | 30 |

Coordinator

Convener

HOD/ECE

Sri Indu
College of Engineering & Technology
UGC Autonomous Institution
Recognized under 2(f) & 12(B) of UGC Act 1956,
NAAC, Approved by AICTE &
Permanently Affiliated to JNTUH
Estd.2001

INSTITUTION'S INNOVATION COUNCIL

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

# HANDS ON TRAINING COURSE ON IMAGE RETRIEVAL PROCESS USING MATLAB

## STARTS ON February 13, 2021

Registration : Free
Course Duration : 4 Week
Weekend Course (Saturday)
Invited Participants: Third Year ECE, EEE, CSE
Restricted to 30 Participants/Slot
Resource Persons: In-house Trainers

| Coordinators | Convener | Principal |
|---|---|---|
| Dr.P.Epsiba | Prof.k.Ashok Babu | Dr.G.Suresh |

1. Foreground Background Extraction

```
clc;
close all;
clear;
%Read Background Image
Background=imread('background.jpg');%Background=rgb2gray(Background);
Background=imresize(Background,[160,160]);
%Read Current Frame
%x1=size(Background);
CurrentFrame=imread('original.jpg');%CurrentFrame=rgb2gray(CurrentFrame)
CurrentFrame=imresize(CurrentFrame,[160 160]);
%Display Background and Foreground
subplot(2,2,1);imshow(Background);title('BackGround');
subplot(2,2,2);imshow(CurrentFrame);title('Current Frame');
%Convert RGB 2 HSV Color conversion
[Background_hsv]=round(rgb2hsv(Background));
[CurrentFrame_hsv]=round(rgb2hsv(CurrentFrame));
Out = bitxor(Background_hsv,CurrentFrame_hsv);
%Convert RGB 2 GRAY
Out=rgb2gray(Out);
%Read Rows and Columns of the Image
[rows columns]=size(Out);
%Convert to Binary Image
for i=1:rows
for j=1:columns
if Out(i,j) >0
BinaryImage(i,j)=1;
else
BinaryImage(i,j)=0;
end
end
end
%Apply Median filter to remove Noise
FilteredImage=medfilt2(BinaryImage,[5 5]);
%Boundary Label the Filtered Image
[L num]=bwlabel(FilteredImage);
STATS=regionprops(L,'all');
cc=[];
removed=0;
%Remove the noisy regions
for i=1:num
dd=STATS(i).Area;
```

```
if (dd < 500)
L(L==i)=0;
removed = removed + 1;
num=num-1;
else
end
end
[L2 num2]=bwlabel(L);
% Trace region boundaries in a binary image.
[B,L,N,A] = bwboundaries(L2);
%Display results
subplot(2,2,3),  imshow(L2);title('BackGround Detected');
subplot(2,2,4),  imshow(L2);title('Blob Detected');
hold on;
for k=1:length(B),
if(~sum(A(k,:)))
boundary = B{k};
plot(boundary(:,2), boundary(:,1), 'r','LineWidth',2);
for l=find(A(:,k))'
boundary = B{l};
plot(boundary(:,2), boundary(:,1), 'g','LineWidth',2);
end
end
end
```

2. Round Object Detection

```
%RGB = imread('10.bmp');
%RGB=imread('E:\D Drive Files 03.11.2014\epsiba phd\Project Code\fusion of local global
estimation\project\Codes\snaps\075.jpg');
imshow(RGB);
I = rgb2gray(RGB);
threshold = graythresh(I);
bw = im2bw(I,threshold);
imshow(bw)
% remove all object containing fewer than 30 pixels
bw = bwareaopen(bw,30);

% fill a gap in the pen's cap
se = strel('disk',2);
bw = imclose(bw,se);

% fill any holes, so that regionprops can be used to estimate
% the area enclosed by each of the boundaries
%bw = imfill(bw,'holes');
figure;
```

```
imshow(bw)
[B,L] = bwboundaries(bw,'noholes');
figure;
% Display the label matrix and draw each boundary
imshow(label2rgb(L, @jet, [.5 .5 .5]))
hold on
for k = 1:length(B)
  boundary = B{k};
  plot(boundary(:,2), boundary(:,1), 'w', 'LineWidth', 2)
end
stats = regionprops(L,'Area','Centroid');

threshold = 0.94;

% loop over the boundaries
for k = 1:length(B)

  % obtain (X,Y) boundary coordinates corresponding to label 'k'
  boundary = B{k};

  % compute a simple estimate of the object's perimeter
  delta_sq = diff(boundary).^2;
  perimeter = sum(sqrt(sum(delta_sq,2)));

  % obtain the area calculation corresponding to label 'k'
  area = stats(k).Area;

  % compute the roundness metric
  metric = 4*pi*area/perimeter^2;

  % display the results
  metric_string = sprintf('%2.2f',metric);

  % mark objects above the threshold with a black circle
  if metric > threshold
    centroid = stats(k).Centroid;
    plot(centroid(1),centroid(2),'ko');
  end

  text(boundary(1,2)-35,boundary(1,1)+13,metric_string,'Color','y',...
      'FontSize',14,'FontWeight','bold');

end

title(['Metrics closer to 1 indicate that ',...
      'the object is approximately round']);
```

3. Edge object detection

```
clc;
clear all;
k=input('Enter the file name','s'); % input image; color image
im=imread(k);
im1=rgb2gray(im);
im1=medfilt2(im1,[3 3]); %Median filtering the image to remove noise%
BW = edge(im1,'sobel'); %finding edges
[imx,imy]=size(BW);
msk=[0 0 0 0 0;
    0 1 1 1 0;
    0 1 1 1 0;
    0 1 1 1 0;
    0 0 0 0 0;];
B=conv2(double(BW),double(msk)); %Smoothing  image to reduce the number of connected
components
L = bwlabel(B,8);% Calculating connected components
mx=max(max(L))
% There will be mx connected components.Here U can give a value between 1 and mx for L or
in a loop you can extract all connected components
% If you are using the attached car image, by giving 17,18,19,22,27,28 to L you can extract the
number plate completely.
[r,c] = find(L==17);
rc = [r c];
[sx sy]=size(rc);
n1=zeros(imx,imy);
for i=1:sx
   x1=rc(i,1);
   y1=rc(i,2);
   n1(x1,y1)=255;
end % Storing the extracted image in an array
figure,imshow(im);
figure,imshow(im1);
figure,imshow(B);
figure,imshow(n1,[]);
```

4. Content Based Image Retrieval

```
[filename, pathname] = uigetfile('*.bmp', 'Pick an Image');
 a=imread(filename);
 figure(1),imshow(a);
 X1=a;
```

```
 [r c]=size(X1);
 a=X1(:,:,1);
 b=X1(:,:,2);
 c=X1(:,:,3);
 [r c]=size(a);
 M=r*c;
 N=reshape(a,[1 M]);
 N=double(N);
 p=[];
 for i=1:M
    p(i)=N(i)./M;
 end
 P=sum(sum(p));
 HSVmap1 = rgb2ycbcr(X1);
 figure(2),imshow(HSVmap1);
  fid = fopen('database.txt');

resultValues = [];     % Results matrix...
resultNames = {};
i = 1;                 % Indices...
j = 1;

while 1
   imagename = fgetl(fid);
   if ~ischar(imagename), break, end      % Meaning: End of File...

%    [X, RGBmap] = imread(imagename);
%    HSVmap = rgb2hsv(RGBmap);
    [X] = imread(imagename);
    figure(3),imshow(X);
    HSVmap = rgb2ycbcr(X);
    figure(4),imshow(HSVmap);


[D1,D2,D3] = quadratic1(X1,  HSVmap1, X, HSVmap);
   resultValues1(i) = D1;
   resultValues2(i) = D2;
   resultValues3(i) = D3;
   resultNames(j) = {imagename};
   i = i + 1;
   j = j + 1;
end

fclose(fid);
[sortedValues1, index1] = sort(resultValues1);     % Sorted results... the vector index
[sortedValues2, index2] = sort(resultValues2);
```

```
[sortedValues3, index3] = sort(resultValues3); % is used to find the resulting files.

%------------------RED-----------------------------------------------

fid = fopen('colourResults_R_C.txt', 'w+');        % Create a file, over-write old ones.

for i = 1:10                                % Store top 10 matches...

   tempstr = char(resultNames(index1(i)));
   fprintf(fid, '%s\r', tempstr);

   disp(resultNames(index1(i)));
   disp(sortedValues1(i));
   disp(' ');
end

fclose(fid);

%------------------GREEN-----------------------------------------------

fid = fopen('colourResults_G_C.txt', 'w+');        % Create a file, over-write old ones.

for i = 1:10                                % Store top 10 matches...
   tempstr = char(resultNames(index2(i)));
   fprintf(fid, '%s\r', tempstr);

   disp(resultNames(index2(i)));
   disp(sortedValues2(i));
   disp(' ');
end

fclose(fid);

%------------------BLUE-----------------------------------------------

fid = fopen('colourResults_B_C.txt', 'w+');        % Create a file, over-write old ones.

for i = 1:10                                % Store top 10 matches...
   tempstr = char(resultNames(index3(i)));
   fprintf(fid, '%s\r', tempstr);

   disp(resultNames(index3(i)));
   disp(sortedValues3(i));
   disp(' ');
end
```

```
fclose(fid);

%return;

disp('Colour part done...');
disp('Colour results saved...');
disp('');

% displayResults1('colourResultsR.txt', 'Colour Results_r...');
%  displayResults1('colourResultsG.txt', 'Colour Results_g...');
%  displayResults1('colourResultsB.txt', 'Colour Results_b...');
% displayResults1('textureResults_r.txt', 'Texture Results_r...');
% displayResults2('textureResults_g.txt', 'Texture Results_g...');

filename='colourResults_R_C.txt';

fid = fopen(filename);

i = 1;              % Subplot index on the figure...

while 1
      imagename = fgetl(fid);
      if ~ischar(imagename), break, end      % Meaning: End of File...

      [x, map] = imread(imagename);

%     subplot(4,5,i);
      if i==1;
     subplot(3,10,1);
%       figure()
      imshow(x);
      end

      if i==2

      subplot(3,10,2);
      imshow(x);
      end

      if i==3
      subplot(3,10,3);
      imshow(x);
      end

      if i==4
      subplot(3,10,4);
```

```
        imshow(x);
        end

        if i==5
       subplot(3,10,5);
        imshow(x);
        end

        if i==6
        subplot(3,10,6);
        imshow(x);
        end

        if i==7
        subplot(3,10,7);
        imshow(x);
        end

        if i==8
        subplot(3,10,8);
        imshow(x);
        end

        if i==9
        subplot(3,10,9);
        imshow(x);
        end

        if i==10
        subplot(3,10,10);
        imshow(x);
        end

        i = i + 1;

end

fclose(fid);

% displayResults1('textureResults_b.txt', 'Texture Results_b...');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

filename='colourResults_G_C.txt';

fid = fopen(filename);
```

```
i = 1;                  % Subplot index on the figure...

while 1
   imagename = fgetl(fid);
   if ~ischar(imagename), break, end       % Meaning: End of File...

   [x, map] = imread(imagename);

%    subplot(4,5,i);
if i==1;
subplot(3,10,11);
imshow(x);
end

if i==2

subplot(3,10,12);

 imshow(x);
end

if i==3
subplot(3,10,13);
 imshow(x);
end
if i==4

subplot(3,10,14);
 imshow(x);
end

if i==5

subplot(3,10,15);
 imshow(x);
end
if i==6

subplot(3,10,16);
 imshow(x);
end
if i==7
subplot(3,10,17);
imshow(x);
end
```

```
if i==8

subplot(3,10,18);
 imshow(x);
end
if i==9

subplot(3,10,19);
 imshow(x);
end

if i==10

subplot(3,10,20);
imshow(x);
end

%    subimage(x, map);
% imshow(x);
%    xlabel(imagename);

   i = i + 1;

end

fclose(fid);

% displayResults1('textureResults_b.txt', 'Texture Results_b...');
filename='colourResults_B_C.txt';

fid = fopen(filename);

i = 1;                % Subplot index on the figure...

while 1
   imagename = fgetl(fid);
   if ~ischar(imagename), break, end       % Meaning: End of File...

   [x, map] = imread(imagename);

%    subplot(4,5,i);

if i==1;
subplot(3,10,21);
imshow(x);
end
```

```
if i==2
subplot(3,10,22);
imshow(x);
end

if i==3
subplot(3,10,23);
imshow(x);
end

if i==4
subplot(3,10,24);
imshow(x);
end

if i==5
subplot(3,10,25);
imshow(x);
end

if i==6
subplot(3,10,26);
imshow(x);
end

if i==7
subplot(3,10,27);
imshow(x);
end

if i==8
subplot(3,10,28);
imshow(x);
end

if i==9
subplot(3,10,29);
imshow(x);
end

if i==10
subplot(3,10,30);
imshow(x);
end
```

```
%    subimage(x, map);
% imshow(x);
%     xlabel(imagename);

   i = i + 1;

end

fclose(fid);
```

Sri Indu
College of Engineering & Technology
UGC Autonomous Institution
Recognized under 2(f) & 12(B) of UGC Act 1956,
NAAC, Approved by AICTE &
Permanently Affiliated to JNTUH
Estd.2001

INSTITUTION'S INNOVATION COUNCIL
(Ministry of Education Initiative)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

# VBB enabled Projects using Arduino

## STARTS ON March 12, 2021

SLOT-I REGISTRATION OPEN

Registration : Free
Course Duration : 12 Hours
Weekend Course (Friday & Saturday)
Invited Participants: Third Year ECE, EEE, CSE
Restricted to 25 Participants/Slot
Resource Persons: In-house Trainers


Coordinators
Dr.P.Mukunthan
Contact: 9894145701

Convener
Prof.K.Ashok Babu

Principal
Dr.G.Suresh

## SRI INDU COLLEGE OF ENGINEERING AND TECHNOLOGY

### DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

HANDS ON TRAINING COURSE

ON

**VBB Enabled Projects using Arduino**

Date: From 12.03.2021 (3 Week Course, Only on Friday & Saturday)

SHORTLISTED STUDENTS

ATTENDANCE SHEET

| S. No | Hall Ticket Number | NAME OF THE STUDENT | Week 1 | | Week 2 | | Week 3 | |
|---|---|---|---|---|---|---|---|---|
| 1 | 18D41AO4H5 | Pyarasani Karthik | Karthik | Karthik | Karthik | Karthik | Karthik | Karthik |
| 2 | 18D41A04H3 | Pusapati Narasimha Raju | P. Raju | P. Raju | P. Raju | P. Raju | P. Raju | P. Raju |
| 3 | 18D41A04H4 | Puttoju Ashrith Chaithanya | Chaithanya | Chaithanya | Chaithanya | Chaithanya | Chaithanya | Chaithanya |
| 4 | 18D41A04G3 | Pasula Aishwarya | Aishwarya | Aishwarya | Aishwarya | Aishwarya | Aishwarya | Aishwarya |
| 5 | 18D41A04L4 | Surepally Pavan | Spavan | Spavan | SPavan | SPavan | SPavan | SPavan |
| 6 | 18D41A04J9 | Sampati Archana | Archana | Archana | Archana | Archana | Archana | Archana |
| 7 | 18D41A04J0 | Rangineni Aparna | Aparana | Aparana | Aparana | Aparana | ← A → | ← A → |
| 8 | 18D41A04F1 | Narsapola Nishanth Reddy | N. Reddy | N. Reddy | N. Reddy | N. Reddy | N. Reddy | N. Reddy |
| 9 | 18D41A04P2 | R.Chandana | | | | | | |
| 10 | 18D41A04A5 | Konduri Madhukar | Madhukar | Madhukar | Madhukar | Madhukar | Madhukar | Madhukar |
| 11 | 18D41A04H2 | Poshanaboina Naveen Kumar | P. kumar | P. kumar | P. kumar | P. kumar | P. kumar | P. kumar |
| 12 | 18D41A04A0 | Kodamarthi.Sai Rohith | Rohith | Rohith | Rohith | Rohith | Rohith | Rohith |
| 13 | 18D41A0461 | Deva Krishna Kanth | Krishnakanth | Krishnakanth | Krishnakanth | Krishnakanth | Krishnakanth | Krishnakanth |
| 14 | ~~18D41A04P2~~ | ~~R.Chandana~~ | | | | | | |
| 15 | 18D41A04F4 | Pakala Sarayu | P. Sarayu | P. Sarayu | P. Sarayu | P. Sarayu | P. Sarayu | P. Sarayu |
| 16 | 17D41A0419 | Bagannagari Sumanth Reddy | S. Reddy | S. Reddy | S. Reddy | S. Reddy | S. Reddy | S. Reddy |
| 17 | 18D41A04E4 | Mu ala Sai Preetham | Preetham | ← A → | Preetham | Preetham | Preetham | Preetham |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 18 | 18D41A04P4 | Nannuri Ruchika Reddy | Ruchika | Ruchika | Ruchika | Ruchika | Ruchika | Ruchika |
| 19 | 17D41A0411 | Aravilli Venkata Chakra Shree Harsha | Shree Harsha | Shree Harsha | Shree Harsha | Shree Harsha | Shree Harsha | Shree Harsha |
| 20 | 17D41A04D1 | Maila Kondal Reddy | | | | | | |
| 21 | 18d41a0431 | B Abhinav Reddy | | | | | | |
| 22 | 17D41A0499 | Kalluri Sandeep Reddy | Sandeep | Sandeep | Sandeep | Sandeep | Sandeep | Sandeep |
| 23 | 17D41A04A3 | K Pavan Kumar | | | A | A | | |
| 24 | 17D41A04B7 | Kotagiri Nitish | | | | | | |
| 25 | 18D41A0497 | Aavula Sairuthwik Vamshikar | Vamshika | Vamshika | Vamshika | Vamshika | Vamshika | Vamshika |
| | | | (24) | (23) | (23) | (23) | (23) | (23) |

Coordinator

Convener

HOD/ECE

SRI INDU COLLEGE OF ENGG. & TECH.

PRINCIPAL
Sri Indu College of Engineering and Technology

Sri Indu
College of Engineering & Technology
UGC Autonomous Institution

INSTITUTION'S INNOVATION COUNCIL

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

# VBB enabled Projects using Arduino

## STARTS ON March 12, 2021

SLOT-I REGISTRATION OPEN

Registration : Free
Course Duration : 12 Hours
Weekend Course (Friday & Saturday)
Invited Participants: Third Year ECE, EEE, CSE
Restricted to 25 Participants/Slot
Resource Persons: In-house Trainers

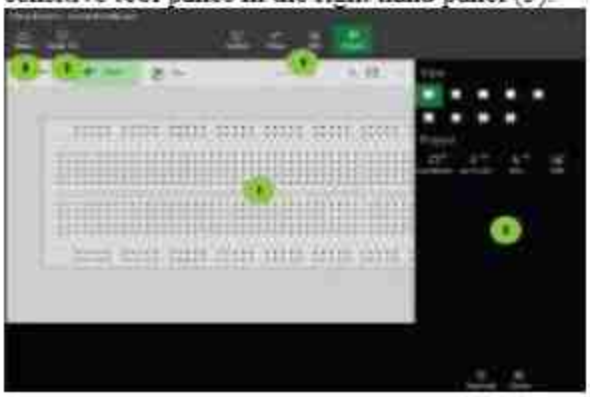
| Coordinators | Convener | Principal |
|---|---|---|
| Dr.P.Mukunthan | Prof.K.Ashok Babu | Dr.G.Suresh |
| Contact: 9894145701 | | |

## Introducing Virtual Breadboard Windows App

The Virtual Breadboard ( VBB ) modern App for the Windows Universal Platform (UWP) uses the Fluent Design System modelled on the Windows Paint 3D App that we all know and love.

In VBB the main design area

(1) is a Design Sheet where you layout Virtual Breadboard circuits which can be virtualized by powering on the circuit (2). The menu (3) opens a navigation view where standard file dialogs along with examples, trainings and account management can be accessed. While designing the toolbar ribbon (4) selects the design mode which shows context sensitive tool panes in the right hand panel (5).



Main functional regions of the App:

1. Design Sheets : Are where you design your Breadboards
2. Power On : Power Up the circuit to start the interactive Virtualization
3. Menu : Open and save projects, find examples and take training and access your account
4. Tools Ribbon : Access the tools for editing and managing your project from the tools ribbon
5. Tools Pane : Tools from the Tools Ribbon are accessed from the Tools Pane

### Avatar Hardware

Your real microcontroller is inserted into a Virtual Breadboard via an Avatar Hardware interface.

Your microcontroller cannot tell the difference between virtual or real.

Other real components can also be connected to your microcontroller creating micro-mixed-reality.

### App Menu Navigation View

The Menu Navigation view slide out panel provides access to the file management, account, settings and other management features.

The platform workspace environment features:

1. File Menu : New, Open, Save, Save as file options
2. Remotify : Publish virtual hardware to the Cloud
3. Student Manager : Create and manage student accounts
4. Export : Export and exchange VBB designs in SVG and KiCad formats
5. Examples : Browse reference examples for quickstarting a project.
6. Training : Awards based training system for getting started.
7. Keyboard Shortcuts : Awards based training system for getting started.
8. Software Store : Make In-App purchases from the Software Store.
9. Hardware Store : Browse the available Avatar interface modules.
10. What's New : Displays the Splash Screen which contains news and quick access tutorials.
11. Learn and Feedback : Access Documentation, YouTube and CodeLab Tutorials and Forum.
12. Account : Account status and login.
13. Settings : Project wide settings and App information.

**Files**

Virtual Breadboard projects are stored in .VBB files. Standard file dialogs are used to open and save .VBB files.

**New**

Click New to create a New blank project. If you already have a project open you will be prompted to save the project before the New project is created

**Open**

Click Open displays the Open panel with tools to open existing project. If you already have a project

open you will be prompted to save the project before the existing project is opened.


1. Browse Files : Click to open a file dialog to locate .vbb files to open
2. Import Previous Project : Click to open a folder dialog to locate and import project folders from earlier versions of VBB
3. Recent Breadboard Files : A listing of recently used .vbb project files. Click a project from the list to open directly

**Save**

Saves the current project to the currently selected .vbb file. If this is the first time you are saving the project the Save as will be activate instead.

**Save as**

Click to open a Save File Dialog to provide a new name and location for your .vbb file and save to the new file.

**Remotify**


Remotify is a publishing system that enables virtual circuits to be published to the internet and played in a Browser.

You can think of Remotify as a Cloud File and Folder manager where the Files are VBB Projects and the folders are Groups of VBB Projects.

Remotify manager maintains a tree heirachy view of the Groups and Projects.

The basic procedure is

1. Create Group
2. Add Project to Group
3. Publish Group or Individual Projects in the group
4. Paste the publish Link to the browser

When publishing url link is copied to the clipboard which can be used in a Browser to Play the virtual hardware project

**HTML5 WebPlayer**


The WebPlayer is a lightweight Html javascript client that connects a Html5 Canvas renderer to a docker container instance of the .Net CORE version of the VBB runtime connected over SignalR. The docker container is hosted as an Azure container instance and spun up on demand per user session so all users have their own dedicated instances giving consistent scaleable performance for each user.

Despite being an economical approach there is a cost associated with hosting cloud sessions and hence the basic subscriptions covers development and casual personal use of the WebPlayer only. If useage grows beyond fair use a seperate usage based susbcription will apply.

HTML5 Canvas currently works on the latest versions of

- Chrome
- Edge

**Published Group**

When a group is published a url will be saved to the clipboard. The url can then be pasted directly to a browser or an alternate such as notepad. The url can then be shared by email or a link in a content management system. When the url is opened or pasted into a browser it will load the group access page with the Name, Description and Thumbnail taken from the group editor fields and the projects.

There are two types of Browser view

- Collections of Groups
- Collections of Projects

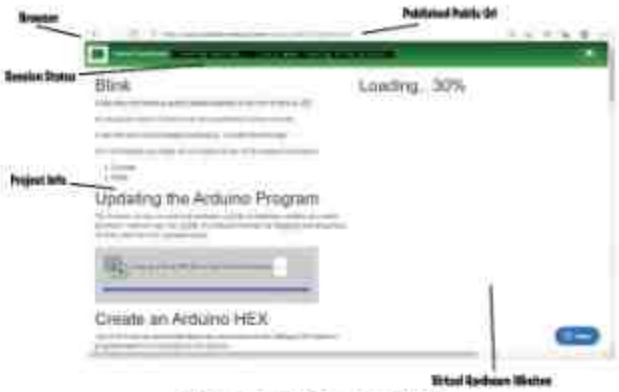
## Group Collections Viewer


## Project Collections Viewer


## Published Project

When a published project is opened from a Group view or by directly pasting or linking the project url the project will load into a project Viewer. If this is the first project accessed in a browser session then a session container will be spun up which currently takes around 30 seconds. In the future standby containers will be used to reduce this intial spin up time. If when switching to a new project within the same session the browser uses can track the current session and reuses the same container. Hence there is no additional spin up time when switching between projects as long as the same browser window is used and standard navigate back button is used to navigate the groups collection heirachy. Each session has a time limit after which it will expire. This can depend on the user account.

Project Viewer Load




## Project Viewer Run


## Remotify Manager Tools

1. Subscription : A subscription and Virtual Breadboard account is required to activate Remotify
2. Remotify : Remotify is accessed from the Navigation View Menu
3. WebPlayer Play Lists : The root directory of the web group collections
4. Group : A Group is a collection of Groups or Projects
5. Project : A Virtual Breadboard Project stored in a Group
6. Group/Project Toolbar : The Toolbar of the currently selected Group or Project
7. Current Project : The VBB project currently open when Remotify Menu was selected

### 1. Subscription Status

To activate the Remotify Manager requires :

- PRO or CLASSROOM Subscription
- Virtual Breadboard Account

For more information about accounts see here : Account

If not activated the Remotify Manager will show the following message.


**2. Remotify Menu**

Remotify is accessed from the Navigation View Menu. This

**3. WebPlayer Play Lists**

The root remotify group contains all the Group or Project collections. There can be only groups or projects at the same level of the heirachy. The Name, Description and Thumbnail are not editable for the root remotify group.

| Icon | Function |
|---|---|
| | Adds a new Group Child to the root directiory |
| | Publishes the project, making it public and copying a url link to the clipboard |
| | UnPublishes the project making it private and not visible in the browser |

**4. Group**

A Group is a collection of Groups or Projects. When selected a group will be highlighted with a blue strip in the heirachy and the group information will be displayed in the Group pane.

**5. Project**

A Virtual Breadboard Project stored in a Group. When selected a project will be highlighted with a blue strip in the heirachy and the project information will be displayed in the Project pane.

**6. Selected Group/Project Toolbar**

**6.1 Selected Group Toolbar**

The Group definition is edited in the Group Pane. The Name, Description can be edited to be displayed in the Browser Group Viewer to inform the nature of the projects in the group. The Thumbnail is also editable and must be a 290x200 png image.

| Icon | Function |
|---|---|
| | Adds a new Group as a child of this group. |
| | Decrements the selected group order moving it up one place in the list |
| | Increments the selected group order moving it down one place in the list |
| | Deletes the selected group if empty removing the group from the cloud. |
| | Publishes the project, making it public and copying a url link to the clipboard |
| | UnPublishes the project making it private and not visible in the browser |
| Update Thu | Opens a dialog to select an 290x200 png image to be the new thumbnail. |
| Save Ch | Saves the new group definition to the remotify cloud. |

**6.2 Selected Project Toolbar**

The Project definition is viewed in the Project Pane. The Name, Description are copied over from the Info designsheet in the VBB Project. The remotify project can be opened and later saved directly to the remotify cloud making remotify a type of cloud drive for VBB projects. The toolbar tools are used to manage the visibility and membership of the project in it's group.


| Icon | Function |
|---|---|
|  | Decrements the selected project order moving it up one place in the list |
|  | Increments the selected project order moving it down one place in the list |
|  | Deletes the selected project removing it from the cloud store |
|  | Publishes the project, making it public and copying a url link to the clipboard |
|  | UnPublishes the project making it private and not visible in the browser |
|   | Opens the remote remotify project as the current project |

Opening Remote Project

When opening a remote project the title of the project will be prefixed with remotify.


When a remotify project is saved it it will be updated directly in the cloud. In this way remotify acts as a cloud drive for VBB projects.

**7. Current Project**


When navigating to the Remotify Manager there is a project currently open in the background. A snapshot of this project is shown in the Current Project pane. The Name and Summary are taken from the project information of the current project and the Thumnail is automatically generated from the active Breadboard. The Project tools are used to connect the project with a Remotify Group.

| Icon | Function |
|---|---|
|  | Add Project to the currently selected group |

**Troubleshooting**

- Firefox does not seem to work with the way VBB uses the Html5 Canvas at this time.
- Projects should be single Breadboard projects.
- A Group should contain only projects or groups but not both otherwise the web player will not function correctly.

**Roadmap**

- ESP8266 is not currently available - backend version needs to be updated.
- Connect Avatars to Remotify Browsers via RasPi Server to create remote labs

**Student Manager**

Classroom administrators use the Student Manager to create and manage Student Accounts. The task of the Student Manager is to enable an Adminstrator to register Student Accounts by following these steps.

1. Edit or import the list of Student Names
2. Set a password for all Students.
3. Click the Register Students Button

Notes:

- Student Names should be unique names.
- The password should be unique to your classroom.
- Student names and accounts can be changed and updated as required.
- Students logged on when registrations are updated will have to logon again.
- Only one student can be logged onto each student name at a time.

**Student Accounts**

Student Accounts are suitable for school use where there are often privacy concerns. A Student license only uses a nickname and is maintained by classroom administrator so there is no student information, email or microsoft account information stored with a Student Account. A Student Account has full access to VBB except with only a few restrictions:

Student Account Restricted Access:

- Student Manager

- Remotify Publishing
- CDK Publishing

**Student Manager Tools**

1. Classroom Subscription : A Classroom subscription is required to activate the Student Manager
2. Student Manager : The Student Manager is access from the Student Manager Navigation Menu
3. Student Names : The Student Manager maintains a list of editable student names representing a class
4. PassPhrase : A shared password is created by editing the password textbox
5. Import Student Names Button : Imports a list of student names
6. Export Student Names Button : Export the current list of students can be d using this button
7. Register Students Button : Create a Virtual Breadboard Student account for each named student with the shared passphrase

**1. Classroom Subscription**

To activate the Student Manager requires two accounts.

- Microsoft Classroom Subscription
- Virtual Breadboard Account

For more information about accounts see here : Account

If not activated the Student Manager will show the following message.


**Microsoft Classroom Subscription**

The Microsoft Classroom Subscription account manages the subscription information. There are 4 classroom subscriptions available to suit different class sizes of 10,20,30 and 60 students. These subscriptions are annual subscription and we have partnered with Microsoft using their In-App purchasing system to simplify the aquistion, invoicing, localised sales taxes and management of these licenses. When you start the Virtual Breadboard App your account information is already known from your Microsoft Windows 10 Logon and so the subscription information is queried automatically using this account.

A classroom subscriber will have the CLASSROOM name shown in their Account stutus


Note : There are actually 2 account required to work with the Student Manager so if only the subscription account is available '*' is shown in the status name to show the account is only partially activated.

**Virtual Breadboard Account**

The Virtual Breadboard Account manages the access to Virtual Breeadboard Cloud backend. The Student Accounts are registered in the cloud which enables students to logon from any location. For more information on creating a Virtual Breadboard account see : Account

When both accounts are registered your account status will display CLASSROOM without the '*' showing the CLASSROOM account is fully activated.


**2. Student Manager**

The Student Manager is access from the Navigation Menu.

**3. Student Names**

An editable collection of student names is managed by the Student Manager.

- Clicking a name will highlight the text ready for editing
- Use the Arrow keys to navigate between names

**4. Pass Phrase**

A single password is shared by all users. The Password should be longer than 6 letters and should be easy to remember but not super obvious.

**5. Import Student Names**

To make it easy to maintain class lists you can import a named list of students.

- Should be a .txt file
- One student per line
- If more students are in the list than are available in the subscription the list will be truncated.

**6. Export Student Names**

To make it easy to maintain class lists you can Export a named list of students.

- The file will be .txt file
- One student per line

**7. Register Students**

Registers the current list of Student names creating a unique account for each student.

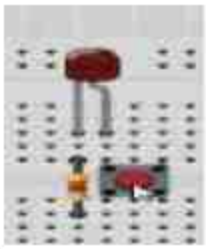
If you have students names that are not unique or there is some other error then you will receive a warning.

If the registration is sucessful you will receive a success message.

**VBB101**

**Contents:**

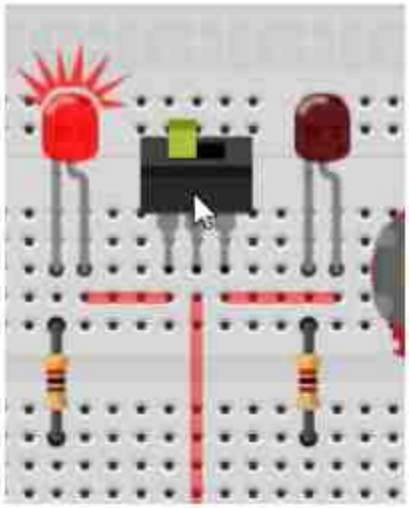
- Interactive



**Toggle a Switch**

A 3 pin switch has two possible configurations. It can connect the command center pin with the left pin or the right pin. When connected to one pin the other pin is disconnected or open circuit. The switch position toggles to show which pin is connected to the center pin.

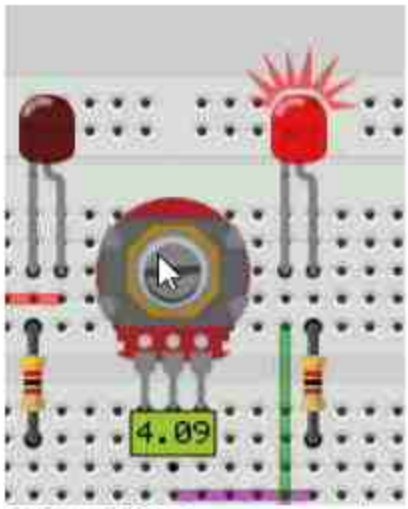
**Award**: While Powered Up locate the Toggle Switch and click the component on the switch location. The position of the switch slider will toggle and the alternate circuit will be completed toggling which L.E.D is activated.



**Rotate a Potentiometer**

A potentiometer is a variable resistor which when you rotate the dial it's resistance changes which changes the output voltage of the central pin. Virtual Breadboard models the voltage output of the potentiometer voltage divider which can be read by instruments like the digital voltmeter and visualised by components like the L.E.D which are sensitive to analog voltage levels.

**Award:** While Powered Up locate the Potentiometer and press and hold the mouse down on the potentiometer dial. Rotate the dial around the center for the full range of voltage values. To receive this reward you need to exercise the full potentiometer range. You will see the L.E.D dim when the voltage is low and brighten when the voltage is high.




**Advanced TIP:** VBB is not an analog PSPICE simulator but instead models high level circuit behaviour. For example a PSPICE simulator will model the reduced L.E.D current and bandgap and deduce the L.E.D should dim but VBB models common circuit behaviours directly.

**Junctions**

Learn how to join links with junctions

**Junction Mode**

Links are drawn between component pins. However there is often the need to link multiple pins together on the same wire. You can draw multiple individual wires from pin to pin but you can also use junctions to simplify circuit layout and make it easier to understand what the connections are.

**Award**: Select Junction Mode from Wires ToolTab

**Place Junctions**

When in Junction Mode the cursor changes to a cross-hair and when the cursor is pressed a junction is placed at the cursor location. Junctions should be placed at the end point of a wire and a wire segment. This joins all the wires into a single wire.

**Award**: Place a junction at each of the 3 wire-t-junctions

**Exercise : Power Up and Verify**

When correctly placed the wire becomes a single wire leading from the DIP power source to the LED lights. All the LEDS should light up when the LED is in the 'on' position.

**Wire Links**

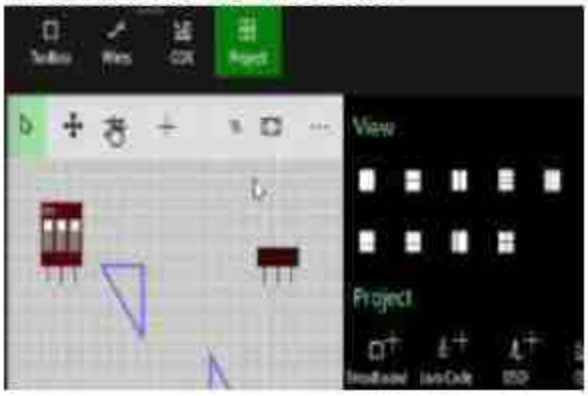
**Wires**

Wires link component pins to form circuits. Real breadboard 'wires' are usually coloured with plastic protection and are stripped back to wire only at the ends. Virtual Breadboard (VBB) wires are the also only active at the ends and snap to pins at either end. When successfully snapped the wire thickens slightly to give a visual clue the link has been successfully made.

**Enter Wire Mode**

**Wire mode** is a design mode and is set from the Wires tool tab. When in wire mode the cursor changes to a colored cross-hair and cursor actions draw wires.

**Award** : Select the Wires Tab and press a colored Wire Button to enter Wire Mode.



**Draw a Link**

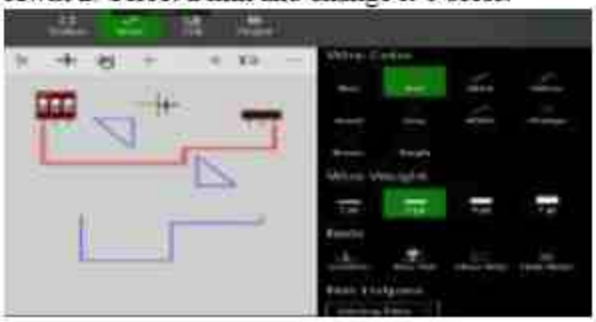
A link is active only at it's ends but it can have multiple 'joint's along it's length to enable better layout organisation than just have a wire running from end to end. Links are drawing by clicking the joints with the left button then either **clicking the right mouse button to end** the link at the last joint **or double clicking** to make the current joint the final link.

**Award**: Draw a link from DIP pin 1 to LED pin 3 avoiding the obstacles with pin joints. To get this award you need to link the correct pins (DIP Pin 1, LED Pin 3).

**Change the Link Color**

If you change your mind about the color of the link you can change it by first selecting it either in move or select mode. When selected the links properties will be shown in the properties pane. Select a new color from color property.

**Award**: Select a link and change it's color.



**Exercise : Practice makes perfect**

Repeat the previous step drawing a link from DIP Pin 2, to LED pin 2. This time use the alternative link ending. So if you used right click method, use double click this time. Or if you used double click use the right click method this time.

**Reposition Link Corner**

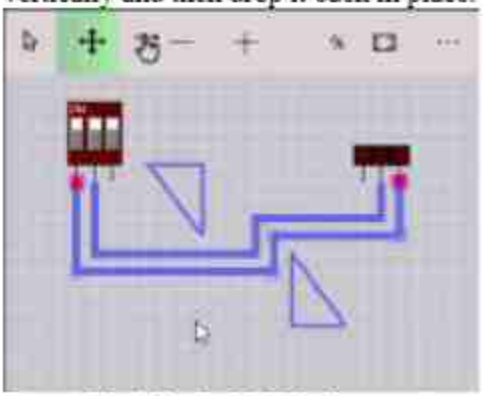
When in Move Mode links can be moved around in several different ways. You can move joints, sections or move the link as a whole.

**Award**: Enter move mode and drag a corner of a link, move it around and then drop it back in place.

**Reposition Horizontal Link Segment**

If you drag a link section from the middle of the section instead of the corners the whole section will move. If the section is horizontal it can be moved vertically.

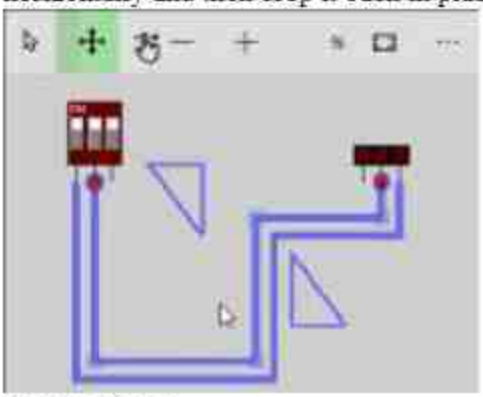
**Award**: Drag a horizontal link section up and down vertically and then drop it back in place.



**Reposition Vertical Link Segment**

If the link section is vertical it can be moved horizontally.

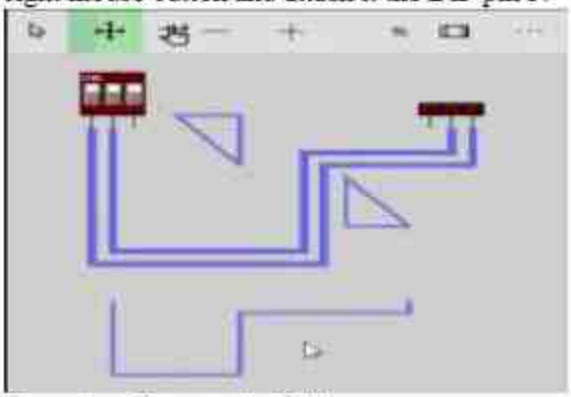
**Award**: Drag a vertical link section left and right horizontally and then drop it back in place.



**Moving Links**

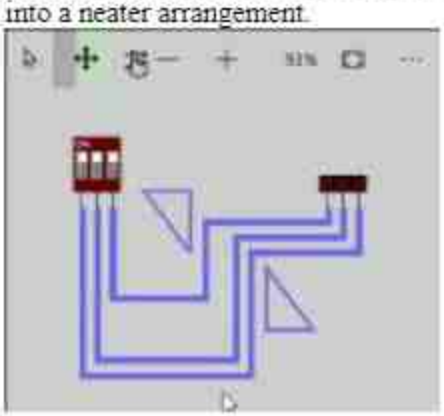
You can also move a link either as part of a group or as a single link by dragging with the **right mouse** button

**Award**: Drag the whole unattached link using the right mouse button and attach it the DIP pin 3.



**Exercise : Reorganise Links**

A neatly organized layout is easier to understand. Practice your new skills to re-arrange the layout into a neater arrangement.



**Moving Components**

**Moving Components**

Virtual Breadboard (VBB) components can be moved around a design by dragging and dropping them while in **Move Mode** .

**Select Move Mode**

Move mode **( ShortCut V)** is one of the Editor Design Modes which can be selected from the Edit mode toolbar. Move Mode is used when you want to move components around.

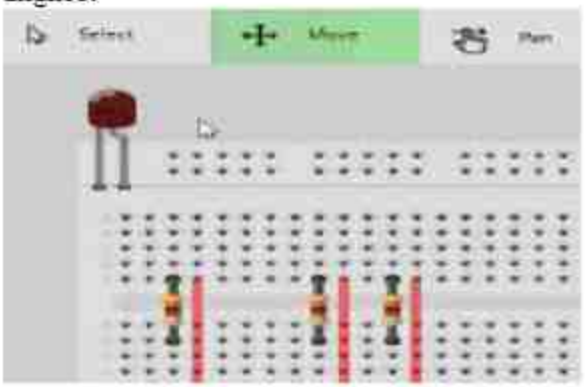
**Award**: Enter move mode by selecting the move icon from the mode toolbar

**Move Component**

When in move mode components can be dragged and dropped into position by pressing and holding and moving to the new location. VBB dynamically calculates the connections the component is making with other components and highlights the shared pin contacts. This is a useful visual guide for making sure components are correctly connected.

**Award**: Drag and Drop the L.E.D to the left most wire/resistor connection using the contact point

highlighting to make sure the L.E.D is correctly aligned.




**Copy and Paste**

Use **Copy** and **Paste** to duplicate the currently selected breadboard elements. Copy copies to the Clipboard and Paste creates new versions of the copied components with a small offset to the originally selected elements and selects the newly created elements.

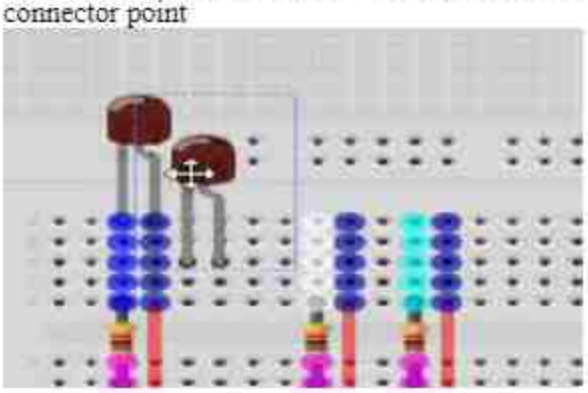
The Select toolbar contains functions that work with the currently selected elements including Copy and Paste.

**Award**: From the Selected toolbar first press Copy and then press Paste to duplicate the L.E.D


**Advanced Tip**: Copy stores the currently selection to the Clipboard and can be pasted to other Breadboards in the project or even Breadboards in other Project.

**Move Duplicate**

Move the duplicate L.E.D to the next resistor/link connector point



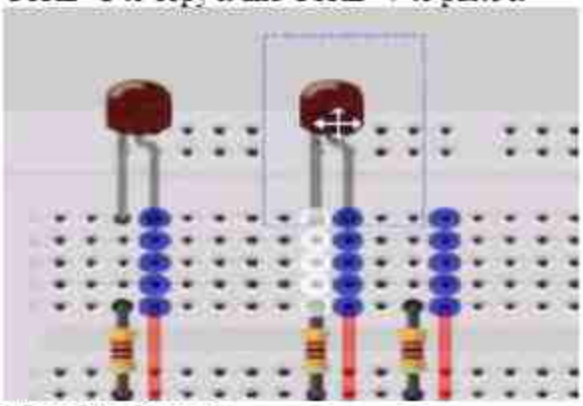
**Copy and Paste with ShortCut keys**

In the previous award the Copy and Paste functions were activated from the toolbar. The VBB App is designed with touchscreens in mind so all functions are available with touchable sized buttons.

However for keyboard users Copy and Paste have well defined shortcut keys in windows || Function || ShortCut | Copy | CTRL+C | Paste | CTRL+V | Cut | CTRL+X

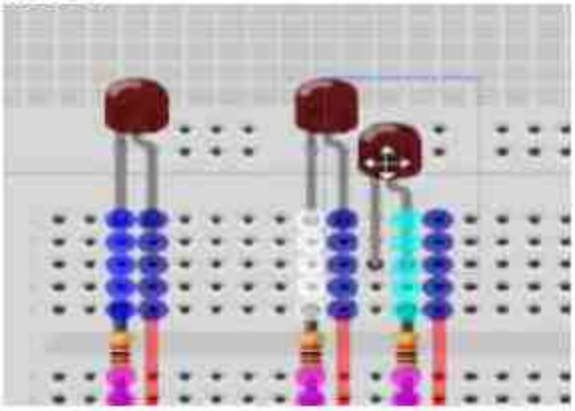
These shortcuts can often be faster to use when working in desktop environment.

**Award**: Copy and Paste the duplicate L.E.D using the short cut keys by first selecting it then using the CTRL+C to copy it and CTRL+V to paste it



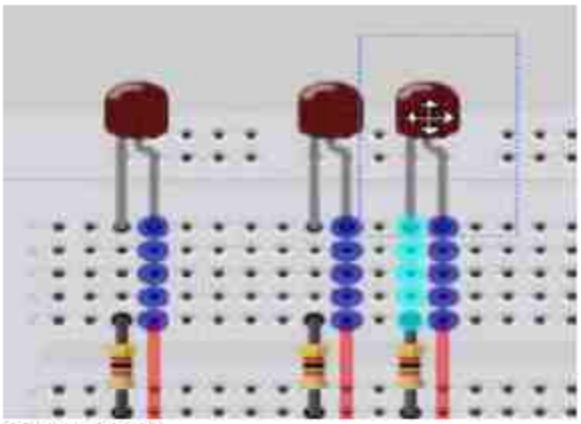
**Move Duplicate 2**

Move the second duplicate L.E.D to the next resistor/link connector point inline with the first two L.E.D's



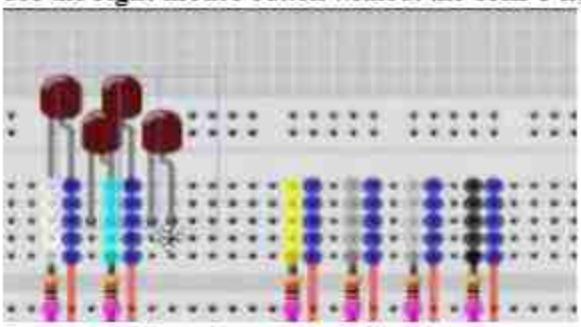
**Group Selection Append**

In an earlier award a group of components were selected by drawing a selection window around the whole group. You can also create groups by appending new components to the group by holding down the **SHIFT** key and then selecting other components to add to the selection group.



**Group Move**

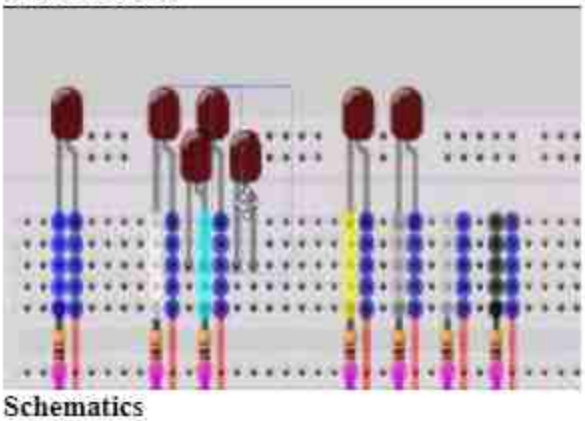
Moving a group of selected elements can be much faster than moving components one at a time. To move the whole group hold down the **SHIFT** key and press and hold one of the selected components. You can then drag and drop the group. You can also use the **right mouse** button without the SHIFT key.



**Power On First Customised Circuit**

Repeat the previous steps 2 steps to practice the copy and moving components.

When ready you can then Power On your circuit to test it



**Schematics**

**Schematics**

Schematics are abstract representations of physical components. It can be easier to understand how a circuit functions when represented in schematic format. Often schematics are used to describe a circuits netlist in a separate design step. In Virtual Breadboard (VBB) Schematics and Breadboard components are merged into one design sheet creating hybrid designs where all the information about the design is in one place.

- Components with the same ID are linked
- Components have schematic/breadboard representations
- When a circuit has a schematic and breadboard component with the same ID the schematic component creates the netlist for the breadboard component.

**Edit 'ID' property**

When a component shares the same 'ID' property the pin the pins of both components are linked with virtual links.

**Award** : Edit the property of the LED to be D1

**Common 'ID' Common Netlist**

Components with the same ID share a netlist between them. To see this copy and paste a component with the same ID and Show the Nets to see the hidden links between the components.

**Exercise**: Duplicate LED D1 and Show Nets

**Edit 'Layout' property**

The Layout property determines which type of view of the component is rendered. So far you have only seen the Breadboard layout. Many components also support a Schematic layout.

**Award**: Select the duplicate LED and set it's layout property to Schematic

**Exercise : Repeat the previous steps**

Repeat the above steps for the Resistor * Edit the Resister property to be R1 Copy and Paste the Resistor Set the duplicate Resistor layout property to be Schematic

**Wire Up the Schematic**

When the schematic twins are wired to form valid circuits the matching Breadboard component twin is also wired with the same netlist.

**Award**: Snap the schematic elements together to form a powered LED circuit

**Exercise : 'Power Up' and verify**

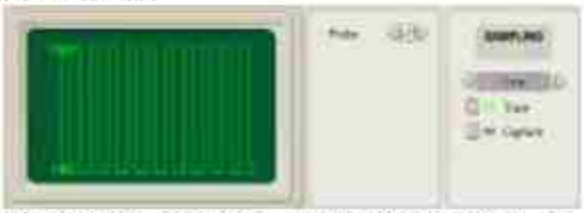
At runtime only the Breadboard component is activated. The schematic component can be used to create an understandable netlist but it does not participate in the runtime circuit.

**Note**: One advantage of this approach is the physical Breadboard components can be placed anywhere without the confusion of connecting wires running all over the place.

**Exercise**: 'Power Up' the circuit and verify the LED brightens as powered by the virtual links.

**DSO Basics**

**DSO Basics**



The **Digital Storage Oscilloscope** (DSO) is an essential tool for visualising and analysing circuits. In this training you will learn the basic steps of adding a DSO to your project and linking it to a probe.

**The DSO Design Sheet**

The DSO has two elements, the DSO design sheet and logic probes. The DSO design sheet is used to visualise the signals and configure the triggers and timebase view parameters. Probes are Breadboard components and are inserted into Breadboard circuits to capture signals and are linked to the DSO to send and visualise the captured signals.

**Award** : Add a DSO Design Sheet to the project

**Viewing the DSO**

Like all Design Sheets the DSO needs to be dragged and dropped into a view pane to actually view it. This allows different configurations of circuits and instrument views to be created and easily switched between.

**Award** : Drag and Drop the DSO Design sheet into the bottom view panel

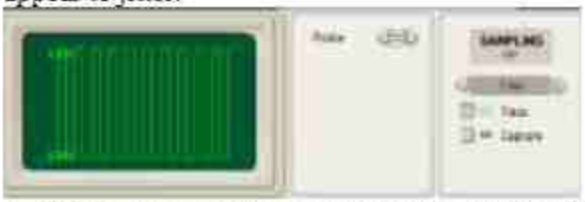
**Attaching a probe**

To capture signal data to display in the DSO you add one or more probes to the circuit.

**Award** :Add a probe and link it to the frequency generator

**Power On to start tracing**

The default mode is Trace Mode which continuously captures signals with the current timebase settings. The signal sampled will be the signal shown. Without a trigger there is no specific event that starts the sampling and the signal will appear to jitter.

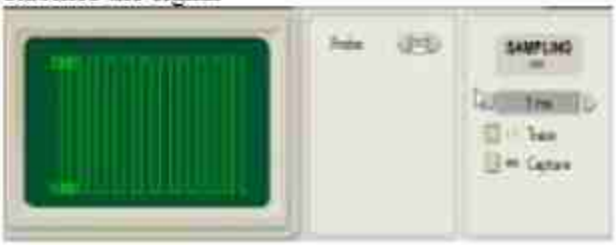


A Trigger is a specific event that the sampler waits for before it begins to sample the signal. The effect of this is to place the signal in known location on the screen allowing visual comparison with previous signals better suited for detecting signal changes.

Triggers are a property of the probes. When powered up the probes in the Breadboard are scanned and added to the DSO Triggers panel. These triggers are logically AND'd together to create a single filter trigger event

| Trigger | Event | Description |
|---|---|---|
| | Don't care | Not used in filter |
| | Is LOW | True if this signal is LOW |
| | Is RISING | True if this signal is Rising Edge |
| | Is HIGH | True if this signal is HIGH |
| | Is FALLING | True if this signal is Falling Edge |

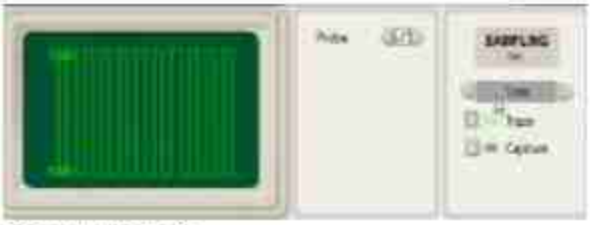
**Award** :Change the Trigger to Rising Edge to stabilise the signal



**Setting the Timebase**

The Time base is the amount of time shown by each grid unit in the DSO display. Changing the Time base has the effect of zooming in and out of the signal.

**Award** :Change the Time base zooming the display in and out



**Signal Analysis**

The DSO is used to sample and display signals to assist in circuit analysis and troubleshooting. For example you can use it to visualise and measure the frequency generated by a frequency generator.

**Exercise** : Change frequency by sliding the slider of the frequency generator and visualise the signal changes in the DSO

PRINCIPAL
Sri Indu College of Engineering and Technology

Estd.2001

# Sri Indu

College of Engineering &

UGC Autonomous Institu

Recognized under 2(f) & 12(B) of U 19

NAAC, Approved by AICTE &

Permanently Affiliated to JNTUH

NAAC

NATIONAL ASSESSMENT AND ACCREDITATION COUNCIL

## CERTIFICATE OF APPRECIATION

*This is to certify that*

**MASADI AJAY of THIRD YEAR ECE**

*has successfully completed the Six Week Value Added Course on* **HANDS ON TRAINING COURSE ON IMPLEMENTATION OF IMAGE PROCESSING CONCEPTS FOR REALTIME APPLICATIONS USING MATLAB** ***organized by the department of Electronics and Communication Engineering*** *of Sri Indu College of Engineering and Technology started on 19.09.2021.*

Prof.K.Ashok Babu
Convener

Dr.G.Suresh
Principal

SRI INDU COLLEGE OF ENGG & TECH

PRINCIPAL
Sri Indu College of Engineering and Technology

Estd.2001

# Sri Indu

College of Engineering &

UGC Autonomous Institu

Recognized under 2(f) & 12(B) of U 19

NAAC, Approved by AICTE &

Permanently Affiliated to JNTUH

NAAC

NATIONAL ASSESSMENT AND ACCREDITATION COUNCIL

## CERTIFICATE OF APPRECIATION

*This is to certify that*

**M. NISHITHA of THIRD YEAR ECE**

*has successfully completed the Six Week Value Added Course on* **HANDS ON TRAINING COURSE ON IMPLEMENTATION OF IMAGE PROCESSING CONCEPTS FOR REALTIME APPLICATIONS USING MATLAB** ***organized by the department of Electronics and Communication Engineering*** *of Sri Indu College of Engineering and Technology started on 19.09.2021.*

Prof.K.Ashok Babu
Convener

Dr.G.Suresh
Principal

SRI INDU COLLEGE OF ENGG. & TECH.

PRINCIPAL
Sri Indu College of Engineering and Technology

Estd.2001

# Sri Indu

College of Engineering &

UGC Autonomous Institu

Recognized under 2(f) & 12(B) of U 19

NAAC, Approved by AICTE &

Permanently Affiliated to JNTUH

NAAC

NATIONAL ASSESSMENT AND ACCREDITATION COUNCIL

## CERTIFICATE OF APPRECIATION

*This is to certify that*

**V.LASYA of THIRD YEAR ECE**

*has successfully completed the Three Week Value Added Course on* Image Retrieval Process using MATLAB ***organized by the department of Electronics and Communication Engineering*** *of Sri Indu College of Engineering and Technology started on 13.02.2021.*

Prof.K.Ashok Babu
Convener

Dr.G.Suresh
Principal

SRI INDU COLLEGE OF ENGG. & TECH.

PRINCIPAL
Sri Indu College of Engineering and Technology

Estd.2001

# Sri Indu

College of Engineering &

UGC Autonomous Institu

Recognized under 2(f) & 12(B) of U 19

NAAC, Approved by AICTE &

Permanently Affiliated to JNTUH

NAAC

NATIONAL ASSESSMENT AND ACCREDITATION COUNCIL

## CERTIFICATE OF APPRECIATION

*This is to certify that*

**GELLA HARINI of THIRD YEAR ECE**

*has successfully completed the Three Week Value Added Course on* Image Retrieval Process using MATLAB ***organized by the department of Electronics and Communication Engineering*** *of Sri Indu College of Engineering and Technology started on 13.02.2021.*

Prof.K.Ashok Babu
Convener

Dr.G.Suresh
Principal

SRI INDU COLLEGE OF ENGG. & TECH.

PRINCIPAL
Sri Indu College of Engineering and Technology
(Vill): Sheriguda-501 510,
Ibrahimpatnam(M), R.R.Dist