**ANCIENT SCRIPT TRANSLATION using MATLAB**

by

P.ILLAVENIL 17BEC1046

AMAN SHAH JI 17BEC1164

ADITYA HARI 17BEC1085

***A project report submitted to***

**Dr. MOHANA PRASAD**

**Associate Professor, School of Electronics Engineering**

in partial fulfilment of the requirements for the course of

**ECE2006 – DIGITAL SIGNAL PROCESSING**

in

**B.TECH ELECTRONICS AND COMMUNICATION ENGINEERING**



**Vellore Institute of Technology, Chennai**

**Vandalur – Kelambakkam Road**

**Chennai – 600127**

**NOVEMBER 2019**



**Chennai**

**BONAFIDE CERTIFICATE**

Certified that this project report entitled “**ANCIENT SCRIPT TRANSLATOR”** is a bonafide work of **P.ILLAVENIL(17BEC1046)**, **AMAN SHAH JI(17BEC1164)** and **ADITYA HARI(17BEC1085)** who carried out the project work under my supervision and guidance.

***Guide Signature***

**ABSTRACT**

The project in which we are working on deals with translation of ancient scripts .We will be using OCR using MATLAB and we will also be adding text to speech engine.

The project is about Optical Character Recognition. It is a process of classifying optical patterns with respect to alphanumeric or other characters. Optical character recognition process includes segmentation, feature extraction and classification.

Text capture converts Analog text based resources to digital text resources. And then these converted resources can be used in several ways like searchable text in indexes so as to identify documents or images.

As the first stage of text capture a scanned image of a page is taken. And this scanned copy will form basis for all other stages. The very next stage involves implementation of technology Optical Character Recognition for converting text content into machine understandable or readable format.

OCR analysis takes the input as digital image which is printed or hand written and converts it to machine readable digital text format. Then OCR processes the digital image into small components for analysis of finding text or word or character blocks. And again the character blocks are further broken into components and are compared with dictionary of characters.

Matlab is an environment where problems and solutions can be denoted in terms of mathematical notations. A use of matlab includes analysis, algorithm development, computation and much more. matlab is a system where elements are placed in an array but are not required any dimensionless. It helps us to solve our problem in no time and provides an easy solution.

The OCR text is written into a pure text file that is then imported again to a search engine. The text is used as index searching of the information. Accuracy rates are measured in several ways and the ways they are measured impact the accuracy rate.

**ACKNOWLEDGEMENT**

We wish to express our sincere thanks and deep sense of gratitude to our project guide, **Dr. Mohana Prasad,** Associate Professor, School of Electronics Engineering, for his consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work.

We are extremely grateful to **Dr. Sivasubramanian A,** Dean, School of Electronics Engineering, VIT Chennai, for extending the facilities of the School towards our project and for her unstinting support.

We express our thanks to our Head of the Department **Dr. Vetrivelan. P / Dr. Thiripurasundari** for their support throughout the course of this project.

We also take this opportunity to thank all the faculty of the School for their support and their wisdom imparted to us throughout the course.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

**P.ILLAVENIL AMAN SHAH JI ADITYA HARI**

**TABLE OF CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **SUB- HEADING** | **NAME** | **PAGE NO.** |
|  |  | ABSTRACT | 3 |
|  |  | ACKNOWLEDGEMENT | 4 |
|  |  |  |  |
| 1 |  | INTRODUCTION | 6 |
|  |  |  |  |
|  | 1.1 | OBJECTIVES AND GOALS | 6 |
|  | 1.2 | BENEFITS | 6 |
|  | 1.3 | FEATURES | 6 |
|  |  |  |  |
| 2 |  | DESIGN | 7-18 |
|  | 2.1 | BLOCK DIAGRAM | 7-9 |
|  | 2.2 | HARDWARE ANALYSIS | 9-18 |
|  |  |  |  |
| 3 |  | SOFTWARE ANALYSIS | 19-20 |
|  |  |  |  |
| 4 |  | CONCLUSION AND FUTURE WORK | 21 |
|  | 4.1 | RESULT, CONCLUSION AND INTERFERANCE | 21 |
|  | 4.2 | FUTURE WORK | 21 |
| 5 |  | REFERENCES |  |

**1.INTODUCTION TO OCR**

Optical Character Recognition(OCR) is the mechanical or electrical conversion of images of typewritten or printed text into machine-encoded text. It is widely used as a form of data entry from printed paper data records, whether passport documents, invoices, bank statements, computerized receipts, business cards, mail, printouts of static-data, or any suitable documentation. It is a common method of digitizing printed texts so that it can be electronically edited, searched, stored more compactly, displayed on-line, and used in machine processes such as machine translation, text-to-speech, key data and text mining. OCR is a field of research in pattern recognition, artificial intelligence and computer vision.

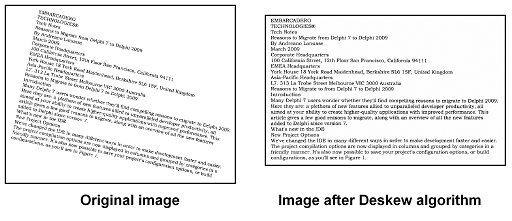
* 1. **Character Recognition and how it works**

The project is an offline recognition system developed to identify either printed characters or discrete run on handwritten characters .It is a part of pattern recognition that usually deals with the realization of the written texts in digital form is that ,It requires less space for storage and can be maintained for further references without referring to the actual script again and again.

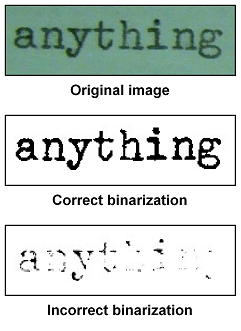
**How it works?**

The process of converting an image to an editable document is divided into several steps. Every step is a set of related algorithms that do a piece of the OCR job. The general steps in the OCR process are as follows:

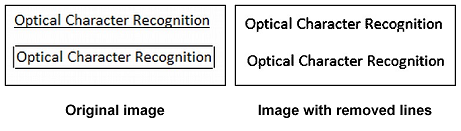
* Loading an image as bitmap from a given source. The source can be a file or a pointer to a memory block. Moreover, a good OCR system must understand a lot of image formats: BMP, TIFF (both one-page and multi-page images), JPEG, PNG, and so on. It must also support PDF files, because many documents are stored as images in the PDF format, and the only way to extract text from such files is to perform OCR.
* Detecting the most important image features, such as resolution and inversion. Many OCR algorithms expect a predefined range of font sizes and foreground/background colours, so the image must be rescaled and inverted before processing if necessary.
* Image can be skewed, or it can have a lot of noise, so de skewing and denoising algorithms are applied to improve the image quality.



* Many OCR algorithms can handle bi-tonal images only, so colour or grayscale images must be converted to bi-tonal. The process is called "binarization." This step is very important, because incorrect binarization will cause a lot of problems.



* Lines detection and removal. This step is required to improve page layout analysis, to achieve better recognition quality for underlined text, to detect tables, etc.



* Page layout analysis (also called "zoning"). The OCR system must detect the positions and types of all important areas in the image.
* Detection of text lines and words. Sometimes it is not an easy task because of different font sizes and small spaces between words.
* Combined-broken characters analysis. Oftentimes, some characters are broken into several parts, or some characters touch each other. It is necessary to detect such cases and find the correct position of every character.



* Recognition of characters. This is the main algorithm of OCR. An image of every character must be converted to the appropriate character code. Sometimes this algorithm produces several character codes for uncertain images. For example, recognition of the image of the "I" character can produce the codes for "I", "|", "1", "l"; the final character code will be selected later.
* Dictionary support. This step can improve the recognition quality. Some characters like "1" and "I", "C" and "G" can look very similar, and the dictionary can help to make the decision.
* Saving results to the selected output format, for example, searchable PDF, DOC, RTF, or TXT. It is important to save the original page layout: columns, fonts, colours, pictures, background, and so on.
  1. **Contribution of the authors**

1.Research on OCR and matlab functions

2.Downloaded dataset for English text

3.Integrated into code

4.Make necessary optimisations

5.Adding Image enhancements

6.Adding voice inputs

7.Text enhancement added with progression

8.Adding picture to voice

**2. ALGORITHM**

**2.1 Abstract of the algorithm used:**

Every optical image when converted into grey scale can be considered as a matrix with 1’s and 0’s as its elements. The theory behind this optical character recognition is division of the image into suitable number of pixels which represent the element of the matrix as stated above and comparing these pixels with those of pre-defined set of templates. For a single character , it can be directly compared with templates whereas for a word which consists of several characters it is primarily divided into group of clusters each consisting of a single character which is compared with the given templates.

**2.2 Pre processing:**

OCR software often "pre-processes" images to improve the chances of successful recognition. Techniques include:

* Deskew – If the document was not aligned properly when scanned, it may need to be tilted a few degrees clockwise or counter clockwise in order to make lines of text perfectly horizontal or vertical.
* Despeckle– remove positive and negative spots, smoothing edges
* Binarisation – Convert an image from colour or grey scale to black and white. The task of binarisation is performed as a simple way of separating the text (or any other desired image component) from the background. The task of binarisation itself is necessary since most commercial recognition algorithms work only on binary images since it proves to be simpler to do so. In addition, the effectiveness of the binarisation step influences to a significant extent the quality of the character recognition stage and the careful decisions are made in the choice of the binarisation employed for a given input image type; since the quality of the binarisation method employed to obtain the binary result depends on the type of the input image (scanned document, scene text image, historical degraded document etc.).
* Line removal – Cleans up non-glyph boxes and lines
* "zoning" – Identifies columns, paragraphs, captions, etc. as distinct blocks. Especially important in multi-customs layout and tables.
* Line and word detection – Establishes a baseline for word and character shapes, separates words if necessary.
* Script recognition – In multilingual documents, the script may change at the level of the words and hence, identification of the script is necessary, before the right OCR can be invoked to handle the specific script.
* Character isolation or "segmentation" – For per-character OCR, multiple characters that are connected due to image artifacts must be separated; single characters that are broken into multiple pieces due to artifacts must be connected.
* Normalize aspect ratio and scale

**2.3 Correlation matrix matching:**

Matrix matching involves comparing an image to a stored glyph on a pixel-by-pixel basis; it is also known as "pattern matching", "pattern recognition", or "image correlation". This relies on the input glyph being correctly isolated from the rest of the image, and on the stored glyph being in a similar font and at the same scale. This technique works best with typewritten text and does not work well when new fonts are encountered. This is the technique the early physical photocell-based OCR implemented, rather directly.

Feature extraction decomposes glyphs into "features" like lines, closed loops, line direction, and line intersections. The extraction features reduces the dimensionality of the representation and makes the recognition process computationally efficient. These features are compared with an abstract vector-like representation of a character, which might reduce to one or more glyph prototypes. General techniques of feature detection in computer vision are applicable to this type of OCR, which is commonly seen in "intelligent" handwriting recognition and indeed most modern OCR software.[[23]](https://en.wikipedia.org/wiki/Optical_character_recognition#cite_note-ocrwizard-23) Nearest neighbour classifier such as the k-nearest neighbour algorithm are used to compare image features with stored glyph features and choose the nearest match.

**2.4 Post processing:**

OCR accuracy can be increased if the output is constrained by a lexicon – a list of words that are allowed to occur in a document. This might be, for example, all the words in the English language, or a more technical lexicon for a specific field. This technique can be problematic if the document contains words not in the lexicon, like proper nouns. Tesseract uses its dictionary to influence the character segmentation step, for improved accuracy.

The output stream may be a plain text stream or file of characters, but more sophisticated OCR systems can preserve the original layout of the page and produce, for example, an annotated PDF that includes both the original image of the page and a searchable textual representation.

"Near-neighbour analysis" can make use of co-occurrence frequencies to correct errors, by noting that certain words are often seen together. For example, "Washington, D.C." is generally far more common in English than "Washington DOC".

Knowledge of the grammar of the language being scanned can also help determine if a word is likely to be a verb or a noun, for example, allowing greater accuracy.

**3.SIMULATION:**

**3.1 Description about Software component**

MATLAB (*matrix laboratory*) is a [multi-paradigm](https://en.wikipedia.org/wiki/Multi-paradigm_programming_language) [numerical computing](https://en.wikipedia.org/wiki/Numerical_analysis) environment and [proprietary programming language](https://en.wikipedia.org/wiki/Proprietary_programming_language) developed by [MathWorks](https://en.wikipedia.org/wiki/MathWorks). Adopting the philosophy that *everything is a matrix*, MATLAB allows [matrix](https://en.wikipedia.org/wiki/Matrix_(mathematics)) manipulations, plotting of [functions](https://en.wikipedia.org/wiki/Function_(mathematics)) and data, implementation of [algorithms](https://en.wikipedia.org/wiki/Algorithm), creation of [user interfaces](https://en.wikipedia.org/wiki/User_interface), and interfacing with programs written in other languages, including [C](https://en.wikipedia.org/wiki/C_(programming_language)), [C++](https://en.wikipedia.org/wiki/C%2B%2B), [C#](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)), [Java](https://en.wikipedia.org/wiki/Java_(programming_language)), [Fortran](https://en.wikipedia.org/wiki/Fortran) and [Python](https://en.wikipedia.org/wiki/Python_(programming_language)).

Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the [MuPAD](https://en.wikipedia.org/wiki/MuPAD) [symbolic engine](https://en.wikipedia.org/wiki/Computer_algebra_system), allowing access to [symbolic computing](https://en.wikipedia.org/wiki/Symbolic_computing) abilities. An additional package, [Simulink](https://en.wikipedia.org/wiki/Simulink), adds graphical multi-domain simulation and [model-based design](https://en.wikipedia.org/wiki/Model-based_design) for [dynamic](https://en.wikipedia.org/wiki/Dynamical_system) and [embedded systems](https://en.wikipedia.org/wiki/Embedded_system).

**3.2 CODE:**

**OCR :**

function str=OCR(imagen)

% OCR (Optical Character Recognition).

warning off %#ok<WNOFF>

close all

% READ THE IMAGE

%imagen=imread('sentence.bmp');

% SHOW IMAGE

imagen1 = imagen;

figure,imshow(imagen1);

title('INPUT IMAGE WITH NOISE')

% Convert to gray scale

%if size(imagen,3)==3 %RGB image

% imagen=rgb2gray(imagen);

%end

% CONVERT THE IMAGE TO GRAYSCALE FOR EASIER READING

threshold = graythresh(imagen);

imagen =~im2bw(imagen,threshold);

imagen2 = imagen;

figure,imshow(imagen2);

word=[ ];

str=string(100);

i=1;

re=imagen;

fid = fopen('text.txt', 'wt'); %OPEN TEXT FILE FOR THE OUTPUT

load templates %LOAD THE TEMPLATES OF THE LETTERS

global templates

num\_letras=size(templates,2); %Compute the number of letters in template file

while 1

%'lines\_crop':SPLITS THE WHOLE PIC INTO LINES

[fl re]=lines\_crop(re); %fl= first line, re= remaining image

imgn=fl;

n=0;

%figure,imshow(fl);pause(0.5)%SHOWS EACH LINE

spacevector = []; %TOTAL NMBER OF SPACES

rc = fl;

while 1

%'letter\_crop' SEPARATES EACH LETTER FROM THE IMAGE

[fc rc space]=letter\_crop(rc); %fc = first letter in the line

%rc = remaining cropped line

%space = space between the letter

% cropped and the next letter

%figure,imshow(fc);pause(0.5) %SHOW LETTERS ONE BY ONE

img\_r = imresize(fc,[42 24]); %RESIZE FOR CORRELATION

n = n + 1;

spacevector(n)=space;

%'read\_letter' CORRELATION IS DONE TO FIND THE CORRECT LETTER

%given in the folder 'letters\_numbers'

letter = read\_letter(img\_r,num\_letras);

word = [word letter]; %LETTER CONCATENATION

if isempty(rc) %breaks loop when there are no more characters

break;

end

end

max\_space = max(spacevector);

no\_spaces = 0; %NUMBER OF SPACES

for x= 1:n %LOOP TO FIND SPACES BETWEEN CHARACTERS

if spacevector(x+no\_spaces)> (0.75 \* max\_space)

no\_spaces = no\_spaces + 1;

for m = x:n

word(n+x-m+no\_spaces)=word(n+x-m+no\_spaces-1);

end

word(x+no\_spaces) = ' ';

spacevector = [0 spacevector];

end

end

fprintf(fid,'%s\n',word); %WRITE WORD READ INTO TEXT FILE

str(i)=word;

i=i+1;

word=[ ];

%BREAK LOOP WHEN SENTENCE ENDS

if isempty(re) %See variable 're' in Fcn 'lines'

break

end

end

fclose(fid);

% winopen('text.txt') %OPEN THE TEXT FILE TO DISPLAY

end

**DATA SET:**

%function read\_letter

function letter=read\_letter(imagn,num\_letras)

% Computes the correlation between template and input image

% and its output is a string containing the letter.

% Size of 'imagn' must be 42 x 24 pixels

% Example:

% imagn=imread('D.bmp');

% letter=read\_letter(imagn)

%load templates

global templates

comp=[ ];

for n=1:num\_letras

sem=corr2(templates{1,n},imagn);

comp=[comp sem];

%pause(1)

end

vd=find(comp==max(comp));

%\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-

if vd==1

letter='A';

elseif vd==2

letter='B';

elseif vd==3

letter='C';

elseif vd==4

letter='D';

elseif vd==5

letter='E';

elseif vd==6

letter='F';

elseif vd==7

letter='G';

elseif vd==8

letter='H';

elseif vd==9

letter='I';

elseif vd==10

letter='J';

elseif vd==11

letter='K';

elseif vd==12

letter='L';

elseif vd==13

letter='M';

elseif vd==14

letter='N';

elseif vd==15

letter='O';

elseif vd==16

letter='P';

elseif vd==17

letter='Q';

elseif vd==18

letter='R';

elseif vd==19

letter='S';

elseif vd==20

letter='T';

elseif vd==21

letter='U';

elseif vd==22

letter='V';

elseif vd==23

letter='W';

elseif vd==24

letter='X';

elseif vd==25

letter='Y';

elseif vd==26

letter='Z';

%\*-\*-\*-\*-\*

elseif vd==27

letter='1';

elseif vd==28

letter='2';

elseif vd==29

letter='3';

elseif vd==30

letter='4';

elseif vd==31

letter='5';

elseif vd==32

letter='6';

elseif vd==33

letter='7';

elseif vd==34

letter='8';

elseif vd==35

letter='9';

elseif vd==36

letter='0';

%\*\*\*\*\*\*\*\*

elseif vd==37

letter='a';

elseif vd==38

letter='b';

elseif vd==39

letter='c';

elseif vd==40

letter='d';

elseif vd==41

letter='e';

elseif vd==42

letter='f';

elseif vd==43

letter='g';

elseif vd==44

letter='h';

elseif vd==45

letter='i';

elseif vd==46

letter='j';

elseif vd==47

letter='k';

elseif vd==48

letter='l';

elseif vd==49

letter='m';

elseif vd==50

letter='n';

elseif vd==51

letter='o';

elseif vd==52

letter='p';

elseif vd==53

letter='q';

elseif vd==54

letter='r';

elseif vd==55

letter='s';

elseif vd==56

letter='t';

elseif vd==57

letter='u';

elseif vd==58

letter='v';

elseif vd==59

letter='w';

elseif vd==60

letter='x';

elseif vd==61

letter='y';

elseif vd==62

letter='z';

elseif vd==63

letter='#';

elseif vd==64

letter='$';

else

letter='l';

%\*-\*-\*-\*-\*

end

**MAIN CODE:**

clc

clear all

% TEXT TO SPEECH

NET.addAssembly('System.Speech');

obj = System.Speech.Synthesis.SpeechSynthesizer;

obj.Volume = 100;

Speak(obj,"Optical character recognition.");

pause(0.5);

Speak(obj,"SELECT IMAGE");

[file,path] = uigetfile({'\*.bmp;\*.png'});

if isequal(file,0)

disp('User selected Cancel')

else

imagen=imread(file);

%imagen = imflatfield(imagen,30);

imagen=imsharpen(imagen);

unstr=OCR(imagen);

str=lower(unstr);

pause(1);

%SPEAK OUT TEXT

for i=1:length(str)

disp(str(i));

Speak(obj,str(i));

end

close all

end

**3.5 Other Applications of OCR:**

Use of OCR engines:

* [Data entry](https://en.wikipedia.org/wiki/Data_entry_clerk) for business documents, e.g. [check](https://en.wikipedia.org/wiki/Check_clearing), passport, invoice, bank statement and receipt
* [Automatic number plate recognition](https://en.wikipedia.org/wiki/Automatic_number_plate_recognition)
* In airports, for passport recognition and [information extraction](https://en.wikipedia.org/wiki/Information_extraction)
* Automatic insurance documents key information extraction
* Extracting business card information into a contact list
* More quickly make textual versions of printed documents, e.g. [book scanning](https://en.wikipedia.org/wiki/Book_scanning) for [Project Gutenberg](https://en.wikipedia.org/wiki/Project_Gutenberg)
* Make electronic images of printed documents searchable, e.g. [Google Books](https://en.wikipedia.org/wiki/Google_Books)
* Converting handwriting in real time to control a computer ([pen computing](https://en.wikipedia.org/wiki/Pen_computing))
* Defeating [CAPTCHA](https://en.wikipedia.org/wiki/CAPTCHA) anti-bot systems, though these are specifically designed to prevent OCR. The purpose can also be to test the robustness of CAPTCHA anti-bot systems.
* Assistive technology for blind and visually impaired user

**RESULT AND FUTURE WORKS:**

**References**

1. **SOFTWARE IMPLEMENTATION**

The software used is code compose studio

**Program code**

#include <msp430.h>

longtempInDeg;

longprevTemp = 0;

longcurrTemp = 0;

/\*

\* main.c

\*/

int main(void) {

WDTCTL = WDTPW | WDTHOLD; // Stop watchdog timer

//select the P1.6 as the PWM output

P1DIR |= BIT6 ;

P1SEL |= BIT6 + BIT1 ;

P2DIR |= BIT2 ;

P2SEL |= BIT2 ;

P1OUT = 0x00 ;

P2OUT = 0x00 ;

// ADC initialization

ADC10CTL0 = ADC10SHT\_3 + SREF\_1 + REFON + ADC10ON + MSC;

ADC10CTL1 = INCH\_10 + INCH\_1 + ADC10DIV\_3 +CONSEQ\_3;

ADC10AE0|= 0x02;

\_delay\_cycles(30);

// PWM initialization

TACTL = TASSEL\_1 + MC\_1 ;

TACCR0 = 1000-1;

TACCTL1 = OUTMOD\_7;

TACCR1 = 500-1;

/\*TBCTL = TBSSEL\_1 + MC\_1 ;

TBCCR0 = 1000-1;

TBCCTL1 = OUTMOD\_7;

TBCCR1 = 500-1;\*/

while(1)

{

ADC10CTL0 |= ENC + ADC10SC; // Sampling and conversion start

while (ADC10CTL1 & ADC10BUSY); // check for ADC conversion is completed

tempInDeg = (((ADC10MEM - 673) \* 423) / 1024) - 10;

currTemp = tempInDeg;

if(currTemp>prevTemp + 2)

{

TACCR1 = tempInDeg\*20 -1;

prevTemp = currTemp;

}

if(currTemp<prevTemp + 2)

{

TACCR1 = tempInDeg\*20 -1;

prevTemp = currTemp;

}

\_delay\_cycles(5000);

if(ADC10MEM>0x03D8)

{

TACCR1=1000;

}

else

{

TACCR1=200;

}

\_delay\_cycles(1000);

}

}

**4. CONCLUSION AND FUTURE WORK**

**4.1 CONCLUSION**

* The home automation system was built and implemented.
* The system is targeted at elderly and disabled people.
* The prototype developed can control electrical devices in a home or office.
* The system implements the wireless network using RF modules for their efficiency and low power consumption.
* The preliminary test results are promising.

**4.2 FUTURE WORK**

* The MSP430 and IR sensor can also be used to track the movements of the person.
* Voice operated system can be implemented
* Adding confirmation commands to the voice recognition system.
* Integrating variable control functions to improve the system versatility such as providing control commands other than ON/OFF commands. For example “Increase Temperature”, “Dim Lights” etc.
* Integration of GSM or mobile server to operate from a distance.
* Design and integration of an online home control panel.

**5. REFERENCES**

1. www.google.com
2. [www.ti.com](http://www.ti.com)
3. [www.circuitstoday.com](http://www.circuitstoday.com)
4. [www.fadooengineers.com](http://www.fadooengineers.com)
5. [www.electroschematics.com](http://www.electroschematics.com)
6. [www.engineersgarage.com](http://www.engineersgarage.com)
7. [www.wikipedia.com](http://www.wikipedia.com)
8. [www.electrosome.com](http://www.electrosome.com)