ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 3, March 2016

Design and Implementation of Real Time Video Image Edge Detection System Using MyRIO

Smruti Ranjan Nayak[1], Shanaz Aman[2], Harish Chandra Mohanta[3], Dr Satyasish Mishra[4]

Lecturer, Dept. of ECE, SoET, CUTM, Bhubaneshwar, India[1]

Assistant Professor, Dept. of ECE, SoET, CUTM, Bhubaneshwar, India[2]

Assistant Professor, Dept. of ECE, SoET, CUTM, Bhubaneshwar, India[3]

Professor & HOD, Dept. of ECE, SoET, CUTM, Bhubaneshwar, India[4]

ABSTRACT: In this paper the design had been aimed for to describe the real time video edge detection based on full use of hardware resources within the FPGA ZYNQ 7010 and ARM Cortex9 within the MyRIO. According to the video signal theory and edge detection algorithm, the author designs the video image edge detection system on the MyRIO custom FPGA template The hardware circuit of the algorithm is achieved on MyRIO and simulated in Labview2014. When the system is validated, it indicates that the real time video image edge detection system can detect high precision edge[1].This approach that can be effectively leverage the opportunities and take on the challenges of modern academic research and development[2].

KEYWORDS: Video Signal, Canny Edge Detection, Labview2014, Custom FPGA template, MyRIO.

I.INTRODUCTION

Edge detection is a fundamental tool used in most image processing applications to obtain information from the video frames in which feature extraction and object segmentation is done In the ideal case, the result of applying an edge detector to an video may lead to a set of connected curves that indicate the boundaries of objects, the boundaries of surface markings as well as curves that correspond to discontinuities in surface orientation [2]. In Xilinx's FPGA hardware resources there are rich internal Multipliers. The design process can directly call these resources to operate, so it is easy to implement complex convolution [1].

In this paper, it proposes real time video image canny edge detection using NI myRIO embedded design device was created to do real-world engineering. It features a 667 MHz dual-core ARM Cortex-A9 programmable processor and a customizable Xilinx field-programmable gate array (FPGA ZYNQ7010)[6]. Rather than spending copious amounts of time debugging code syntax or developing user interfaces, in this paper uses the LabVIEW graphical programming paradigm to focus on constructing their systems and solving their design problems without the added pressure of a burdensome tool.

II.MODELLING OF REAL TIME VIDEO IMAGE EDGE DETECTION

The Canny edge detection algorithm is known as the optimal edge detector [5]. The first and most obvious criterion is low error rate. It is important that edges occurring in images should not be missed and that there be no responses to non-edges. The second criterion is that the edge points be well localized. In other words, the distance between the edge pixels as found by the detector and the actual edge is to be at a minimum. A third criterion is to have only one response to a single edge. This was implemented because the sobel were not substantial enough to completely eliminate the possibility of multiple responses to an edge[5].

Based on these criteria, the canny edge detector first smoothes the image to eliminate and noise. It then finds the image gradient to highlight regions with high spatial derivatives. The algorithm then tracks along these regions and

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

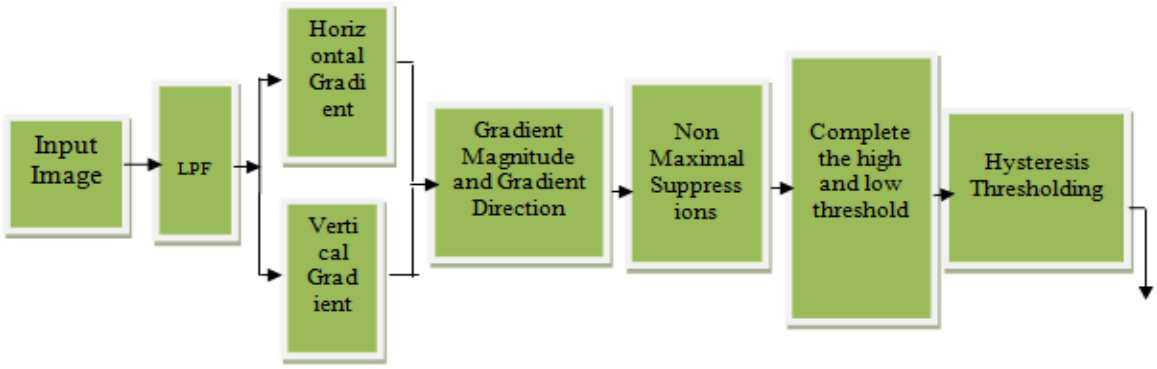
International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 3, March 2016

suppresses any pixel that is not at the maximum (non maximum suppression). The gradient array is now further reduced by hysteresis. Hysteresis is used to track along the remaining pixels that have not been suppressed. Hysteresis uses two thresholds and if the magnitude is below the first threshold, it is set to zero (made a non edge). If the magnitude is above the high threshold, it is made an edge. And if the magnitude is between the 2 thresholds, then it is set to zero unless there is a path from. This paper design and implemented a algorithm for real time video image canny edge detection using NI myRIO this embedded hardware is to simplify hardware setup. To accomplish this, NI myRIO software provides a custom setup and configuration utility separate from the NI Measurement & Automation Explorer (MAX) configuration utility.

III.ALGORITHIM FOR CANNY EDGE DETECTION



Figure(1):Block Diagram Of Canny Edge Detection

Step 1

In order to implement the canny edge detector algorithm, a series of steps must be followed. The first step is to filter out any noise in the original image before trying to locate and detect any edges. And because the Gaussian filter can be computed using a simple mask, it is used exclusively in the Canny algorithm. Once a suitable mask has been calculated, the Gaussian smoothing can be performed using standard convolution methods. A convolution mask is usually much smaller than the actual image. As a result, the mask is slid over the image, manipulating a square of pixels at a time. **The larger the width of the Gaussian mask, the lower is the detector's sensitivity to noise**. The localization error in the detected edges also increases slightly as the Gaussian width is increased. The Gaussian mask used in my implementation is shown below.

$$\frac{1}{115}\begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

Figure(1):Discrete approximation to Gaussian function with δ=1.4

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

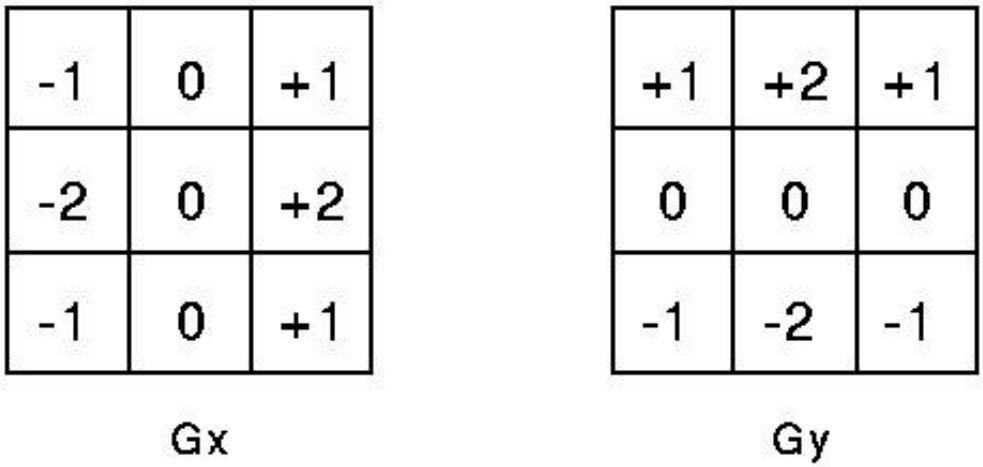
International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 3, March 2016

Step 2

After smoothing the image and eliminating the noise, the next step is to find the edge strength by taking the gradient of the image. The Sobel operator performs a 2-D spatial gradient measurement on an image. Then, the approximate absolute gradient magnitude (edge strength) at each point can be found. The Sobel operator uses a pair of 3x3 convolution masks, one estimating the gradient in the x-direction (columns) and the other estimating the gradient in the y-direction (rows). They are shown below:



Figure(2): Sobel Operator

The magnitude, or EDGE STRENGTH, of the gradient is then approximated using the formula:

$$|G| = |Gx| + |Gy|$$

Step 3

Finding the edge direction is trivial once the gradient in the x and y directions are known. However, you will generate an error whenever sumX is equal to zero. So in the code there has to be a restriction set whenever this takes place. Whenever the gradient in the x direction is equal to zero, the edge direction has to be equal to 90 degrees or 0 degrees, depending on what the value of the gradient in the y-direction is equal to. If GY has a value of zero, the edge direction will equal 0 degrees. Otherwise the edge direction will equal 90 degrees. The formula for finding the edge direction is just:

$$\text{theta} = \text{invtan}\ (Gy / Gx)$$

Step 4

Once the edge direction is known, the next step is to relate the edge direction to a direction that can be traced in an image. So if the pixels of a 5x5 image are aligned as follows:

x x x x x

x x x x x

x x a x x

x x x x x

x x x x x

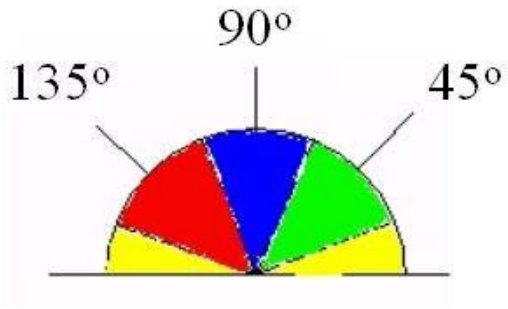
ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 3, March 2016

Then, it can be seen by looking at pixel "**a**", there are only four possible directions when describing the surrounding pixels - **0 degrees** (in the horizontal direction), **45 degrees** (along the positive diagonal), **90 degrees** (in the vertical direction), or **135 degrees** (along the negative diagonal). So now the edge orientation has to be resolved into one of these four directions depending on which direction it is closest to (e.g. if the orientation angle is found to be 3 degrees, make it zero degrees). Think of this as taking a semicircle and dividing it into 5 regions.



Figure(3): Edge Direction

Therefore, any edge direction falling within the yellow range (0 to 22.5 & 157.5 to 180 degrees) is set to 0 degrees. Any edge direction falling in the green range (22.5 to 67.5 degrees) is set to 45 degrees. Any edge direction falling in the blue range (67.5 to 112.5 degrees) is set to 90 degrees. And finally, any edge direction falling within the red range (112.5 to 157.5 degrees) is set to 135 degrees.

Step 5

After the edge directions are known, non maximum suppression now has to be applied. Non maximum suppression is used to trace along the edge in the edge direction and suppress any pixel value (sets it equal to 0) that is not considered to be an edge. This will give a thin line in the output image.

Step 6

Finally, hysteresis is used as a means of eliminating streaking. Streaking is the breaking up of an edge contour caused by the operator output fluctuating above and below the threshold. If a single threshold, T1 is applied to an image, and an edge has an average strength equal to T1, then due to noise, there will be instances where the edge dips below the threshold. Equally it will also extend above the threshold making an edge look like a dashed line. To avoid this, hysteresis uses 2 thresholds, a high and a low. Any pixel in the image that has a value greater than T1 is presumed to be an edge pixel, and is marked as such immediately. Then, any pixels that are connected to this edge pixel and that have a value greater than T2 are also selected as edge pixels. If we think of following an edge, we need a gradient of T2 to start but we don't stop till we hit a gradient below T1.

IV. ARCHITECTURE OF NI MYRIO

The NI myRIO-1900 provides analog input (AI), analog output (AO), digital input and output (DIO), audio, and power output in a compact embedded device. The NI myRIO-1900 connects to a host computer over USB and wireless 802.11b,g,n[7]. NI myRIO is a reconfigurable and reusable tool that helps to learn a wide variety of engineering concepts as well as complete design projects. The RT and FPGA capabilities along with onboard memory and built-in WiFi allow as to deploy applications remotely and run them “headlessly” (without a remote computer connection). Three connectors (two NI myRIO expansion ports [MXP] and one NI miniSystems port [MSP] that is identical to the NI myDAQ connector) send and receive signals from sensors and circuitry that students need in their systems. Forty digital I/O lines overall with support for SPI, PWM out, quadrature encoder input, UART, and I2C; eight single-ended analog inputs; two differential analog inputs; four single-ended analog outputs; and two ground-referenced analog outputs allow for connectivity to countless sensors and devices and programmatic control of systems[6]. All of this

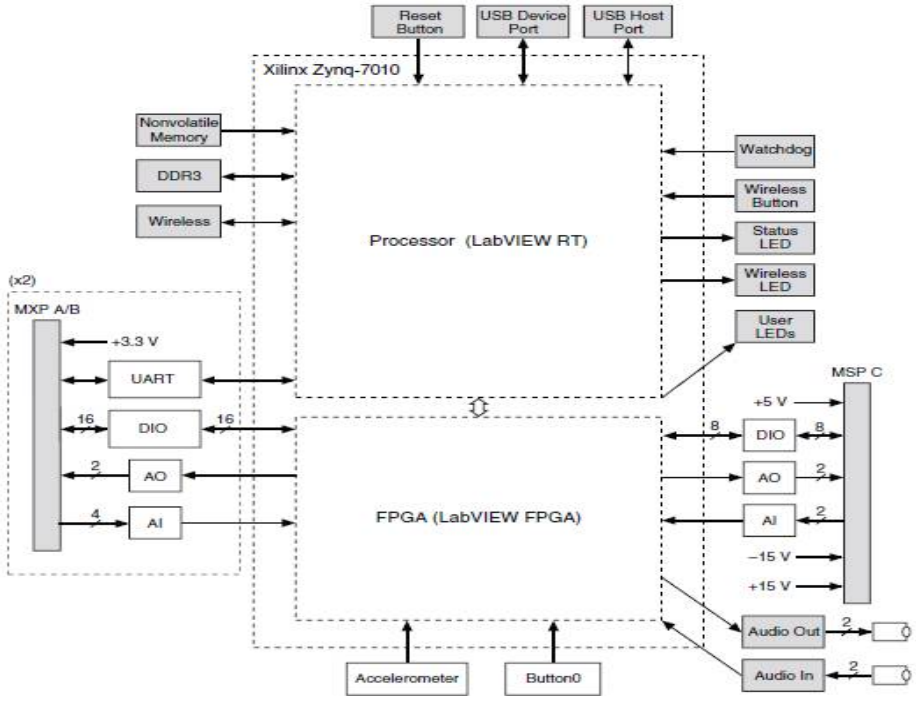
ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 3, March 2016

functionality is built-in and preconfigured in the default FPGA functionality, which eliminates the need for expansion boards or “shields” to add utility. Ultimately, these features allows as to do real-world engineering .



Figure(4)[7]:NI MYRIO Hardware Block Diagram

V. HARDWARE IMPLEMENTATION OF CANNY EDGE DETECTION USING MYRIO

Here we explained video edge detection by using LabVIEW MyRIO 2014. At the beginning video signal is read from camera and is divided into a number of frames. These frames are applied to edge detection block one by one and displayed on the screen.

Figure(5): MyRIO Interfacing with camera

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

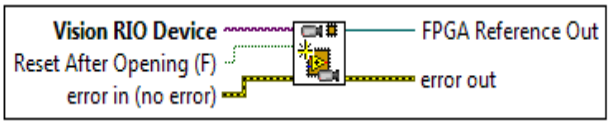
International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 3, March 2016

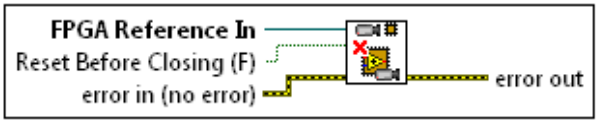
Vision RIO Open FPGA Reference VI:

 Opens a reference to the specified FPGA device. You must open a reference to the FPGA device before you can use any Vision RIO VI that requires an FPGA Reference



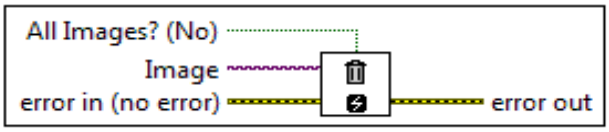
Vision RIO Close FPGA Reference VI:

 Closes the reference to the FPGA device and, optionally, resets execution of the FPGA on the device.



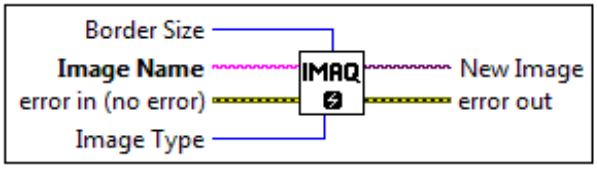
IMAQ Dispose VI:

 it destroy an image and frees the space it occupied in memory. This VI is required for each image created in an application to free the memory allocated to the IMAQ create.



IMAQ Create VI:

 it create a temporary memory location for an image ,IMAQ Dispose VI to create or dispose NI Vision images in Labview.



IMAQdx Open Camera VI:

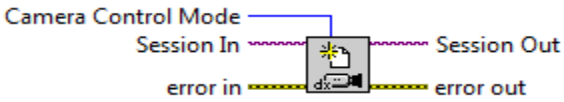
 NI-IMAQdx driver software gives you the ability to acquire images with IEEE 1394 and Gig E vision camera.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

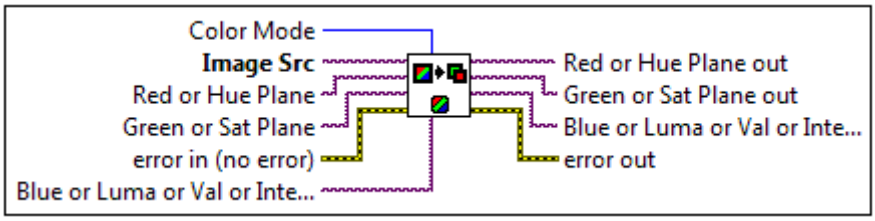
(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 3, March 2016

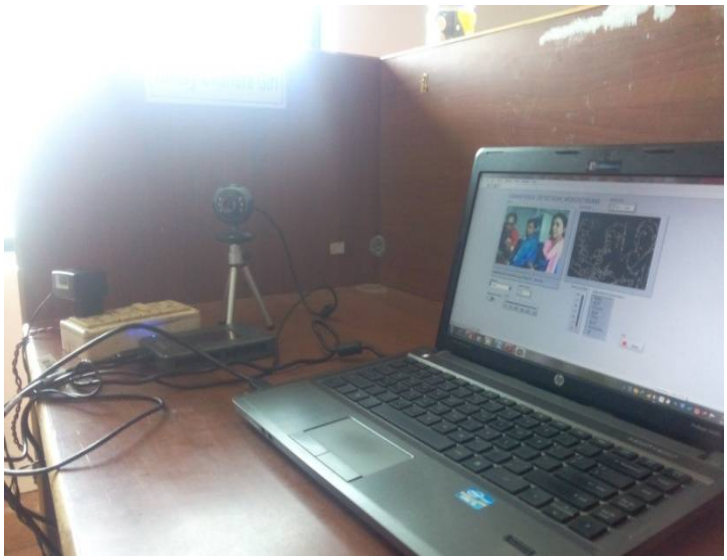


IMAQ Extract Color Planes VI:


it extract the three planes(RGB,HSL,HSI) from an image.



V. RESULT AND DISCUSSION



Figure(6):Real Time Video Signal Captured By Camera

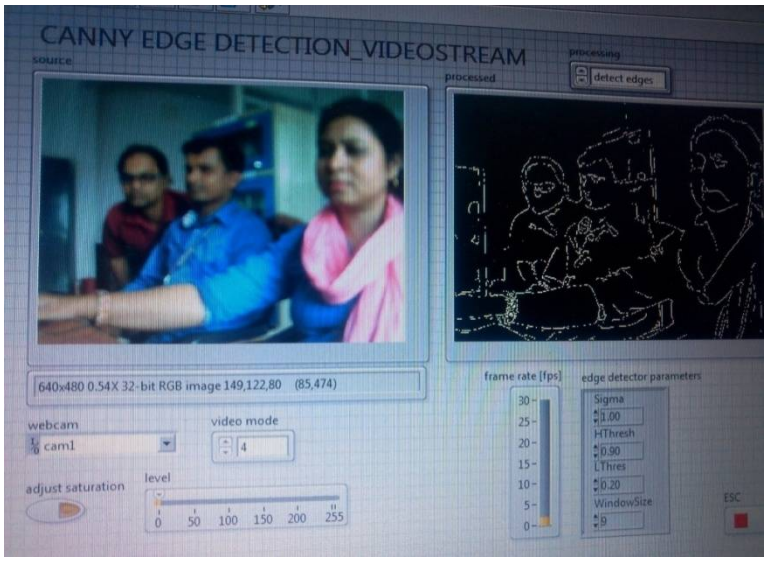
In the figure 6 , it shows the camera interfacing with MyRIO ,where real time video is being captured by camera and send it to MyRIO hardware(FPGA ZYNQ 7010 & ARM Cortex9) for further process.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 3, March 2016



Figure(7): Real Time Video stream Canny Edge Detection Using MyRIO

In Figure 7, it shows that real time video stream is being captured by camera and shown in source display and simultaneously MyRIO process take the video stream and process the algorithim by FPGA and for real time operation sends the process signal to ARM Cortex 9 processor, which is shown in processed display. Here we have used Video 4 mode sigma value 1.00,H Threshold 0.90,LThreshold 0.20 and window size 9,for high resolution of real time canny edge detection.

VI.CONCLUSION

This paper realizes a hardware-based video edge detection system on MyRIO which uses FPGA ZYNQ 7010 and ARM Cortex9 Processor, mainly by the completion of the data flow programming using Labview 2014 in FPGA custom MyRIO template. After verification it indicates that the MyRIO can detect the real time video edge with high precision satisfying the requirements of the real-time video image edge detection.

REFERENCES

[1]Jincheng Wu, Jingrui Sun, Wenying LiuCollege of Physics & Electronic Information Science Tianjin Normal University,P.R.China." Design and Implementation of Video Image edge Detection System Based on FPGA)" Date of Conference 16-18 oct -2010,page No-472-476,ISBN Code-978-1-4244-6513,Inspection Accession No-11676142.
[2]Ravi Kumar A.V, Dr. Nataraj K.R, Dr.Rekha K.R Electronics Communication Engineering Department, V T U SJBIT Bangalore, Karnataka, India"Morphological Real Time Video Edge Detection in Labview"Vol-3(2),Page NO-3808-3811,ISSN-0975-9646.
[3]Tariq M. Khana,∗, D.G. Baileyb, Mohammad A.U. Khanc, Yinan Kongaa "Real-time edge detection and range finding using FPGAs ,Optic 126 (2015),Page No-1545-1550.
[4]R .Ponneela Viignesh, R Rajendram"Performance And analysis of Edge detection using FPGA Implementation .IJMER ,Vol-2 Issue-2,March-April-2012,PP-552-554,ISSN-2249-6645
[5]John Canny, IEEE Transactions on pattern analysis and machine intelligence, vol. pami-8, No. 6, November 1986
[6] National Instrument,"Design real sytem fast manual",http://www.ni.com/white-paper/52481/en/
[7] National Instrument,"User Guide & Specification",www.ni.com/pdf/manuals/376047a.pdf