International Journal of Technical Research and Applications e-ISSN: 2320-8163,
www.ijtra.com Volume 2, Issue 3 (May-June 2014), PP. 59-66

ULTRASONIC PERIPATETIC SCANNER FOR AUTONOMOUS TEST BENCH USING RASPBERRY Pi

D. Kole[1], S.Chakraborty[2], S. Bhunia[3]
[1]R&D Section, Deep Micro System, Hooghly, WB, India
[2]Dept. of Computer Science, Swami Vivekananda University, Madhya Pradesh, India
[3]Department of Electronics & Telecommunication, Indian Institute of Engineering

Abstract— **The autonomous Test Bench is the growing field of testing device. The Rapid production firms require rapid testing infrastructure. Many firms till now use legacy system. The Autonomous Test Bench is a Rapid Application Development tool to accelerate the device testing rapid. To implementation ATB (Autonomous Test Bench) the major requirement is to measure the target object distance from robotic arm to trigger the device (like push button, gripping something etc.) Not only Robotic system, many fields of industry are required to capture ultrasonic scan data. Sometimes it should be wireless system which may be positioned anywhere. This paper describes a novel way to capture surrounded ultrasonic scan data. The device is portable and wireless and as well as cost effective. To measure object distance, ultrasonic peripatetic scanner uses two servo motors to scan the object horizontally and vertically. The main controller is a credit card sized computer Raspberry pi with high processing capability and portability. The programs are written using Python which is an interpreted language.**

Key words— **Ultrasonic sensor, Distance measurement using sonar, Raspberry pi, Wi-Fi, Ultrasonic Scanner.**

I. INTRODUCTION

This paper addresses the problem of extracting the map of an unknown environment from echo signals received by ultrasonic transducers. A new map-building method is developed and demonstrated through simulations for ultrasonic sensors. Ultrasonic transducer HC SR-04 is used for scanning the environment. The method uses ultrasonic range and amplitude data to build a Three-dimensional map of an unknown environment. All of the amplitude information is taken into account in the simulation studies. The received echo amplitude where it exceeds a predefined threshold is used for drawing the map. The method can extract the surface profiles from the received echoes with knowledge of transducers' position. With this method, sonar signal returns from arbitrary indoor environments can be represented realistically and with relatively low computation time. Different types of environments composed of planar surfaces, corners, edges, cylinders, and arbitrarily curved surfaces have been considered. An error criterion is developed to assess the accuracy of the extracted maps. Horizontal and vertical movement of transducers gives the best estimate of the surface profiles. Most of the closed working space is a continuous combination of basic shapes. The method is fast and suitable for real-time map building applications.[45].

II. PRELIMINARIES / BACKGROUND

Doing something by any robotic arm, the main criteria is to measure distance between robot's end effector and the object. Distance value is used to move robot arm to capture, to grip or to trigger something. The most popular technique to measure the distance is Ultrasonic or Sonar sensor. It is easy to operate and cheap. Another method is to measure the distance by capturing image using still or video Camera. To extract the distance from captured image is too tedious. The captured Image is passes through various algorithms like object detection, segmentation, morphological processing etc. which require high processing power. To reduce the complexity, last of all we have accepted ultrasonic technique. It makes ATB (Autonomous Test Bench) viable by introducing some error correction algorithm.

III. PROPOSED TECHNIQUE

Fig.-1: System over View

Ultrasonic Module: Ultrasonic sensors (also known as transceivers when they both send and receive, but more generally called transducers) work on a principle similar to radar or sonar which evaluates attributes of a target by interpreting the echoes from radio or sound waves respectively. Ultrasonic sensors generate high frequency sound waves and evaluate the echo which is received back by the sensor. Sensors calculate the time interval between sending the signal and receiving the echo to determine the distance to an object.

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the aging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work of the system is as follows.

Using IO trigger for at least 10us high level signal the module automatically sends eight 40 kHz pulses and detect whether there is a reflected pulse signal from the object. In that case measuring the time of high output duration we can easily calculate the distance of the object from the sensor. Test distance = (high level time x velocity of sound (340M/S) /2

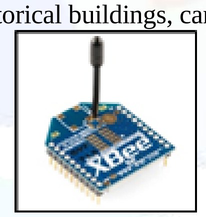
The timing diagram is shown below in Fig.2. We only need to supply a short 10uS pulse to the trigger input to start the ranging, and then the module will send out 8 cycle bursts of ultrasound at 40 kHz and raise its echo. The echo from the object is in pulsed nature whose width is proportional to the range of the object. We can calculate the range through the time interval between sending trigger signal and receiving echo signal using formula: uS / 58 = centimeters or uS / 148 =inch; or: the range = high level time * velocity (340M/S) / 2; we suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal.

Fig.-2: Signal Flow of Sonar

Wi-Fi : also spelled Wifi or WiFi, is a popular technology that allows an electronic device to exchange data or connect to the internet wirelessly using radio waves.

To connect to a Wi-Fi LAN, a computer has to be equipped with a wireless network interface controller. The combination of computer and interface controller is called a station. All stations share a single radio frequency communication channel. Transmissions on this channel are received by all stations within range. The hardware does not signal the user that the transmission was delivered and is therefore called a best-effort delivery mechanism. A carrier wave is used to transmit the data in packets, referred to as "Ethernet frames". Each station is constantly tuned in on the radio frequency communication channel to pick up available transmissions.

Wi-Fi allows cheaper deployment of local area networks (LANs). Also spaces where cables cannot be run, such as outdoor areas and historical buildings, can host wireless LANs.



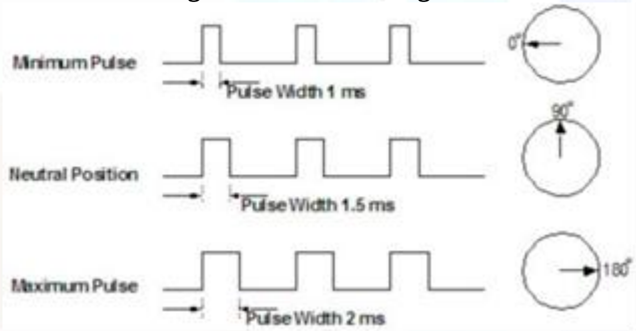
Servo Motor: Servo motors have been around for a long time and are utilized in many applications. They are small in size but pack a big punch and are very energy-efficient. Because of these features, they can be used to operate remote-controlled or radio-controlled toy cars, robots and airplanes. Servo motors are also used in industrial applications, robotics, in-line manufacturing.

Servos are controlled by sending an electrical pulse of variable width, or pulse width modulation (PWM), through the control wire. There is a minimum pulse, a maximum pulse, and a repetition rate. A servo motor can usually turn 90 degrees only in either direction for a total of 180 degree movement. The motor's neutral position is defined as the position where the servo has the same amount of potential rotation in the both the clockwise or counter-clockwise direction. The PWM sent to the motor determines position of the shaft, and based on the duration of the pulse sent via the control wire; the rotor will turn to the desired position. The servo motor expects to see a pulse every 20 milliseconds (ms) and the length of the pulse will determine how far the motor turns. For example, a 1.5ms pulse will make the motor turn to the 90-degree position.

If an external force pushes against the servo while the servo is holding a position, the servo will resist from moving out of that position. The maximum amount of force the servo can exert is called the torque rating of the servo. Servos will not hold their position forever though; the position pulse must be repeated to instruct the servo to stay in position.

Of course, we don't have to know how a servo works to use one, but as with most electronics, the more we understand, the more doors open for expanded research and its capabilities.

Fig.-3: Servo Pulse diagram



IV. RASPBERRY PI (MODEL B)

A. Introduction

The Raspberry Pi is a credit-card-sized single-board computer developed in the UK by the Raspberry Pi Foundation with the intention of promoting the teaching of basic computer science in schools. [47]

B. Raspberry Pi at a Glance

Hardware	Specification
CPU	700 MHz ARM1176JZF-S core
Instruction	ARMv6 instruction set
GPU	OpenGL ES 2.0 (24 GFLOPS)
SD RAM	512 MB

USB 2.0 ports	2
Video input	A CSI input connector
Video outputs	Composite RCA (PAL and NTSC)
Audio outputs	3.5 mm jack
Onboard storage	SD / MMC / SDIO
GPIO	8
UART	1
Ethernet	10/100Mbit/s
I2C BUS	1
SPI BUS	1 with two chip select
Power ratings	700 mA (3.5 W)
Power source	5 volt via MicroUSB or GPIO header
Size	85.60 mm × 53.98 mm
Weight	45 g
Operating systems	Arch Linux ARM, Debian GNU/Linux, Gentoo, Fedora, FreeBSD, NetBSD, Plan 9, Raspbian OS, RISC OS, Slackware Linux

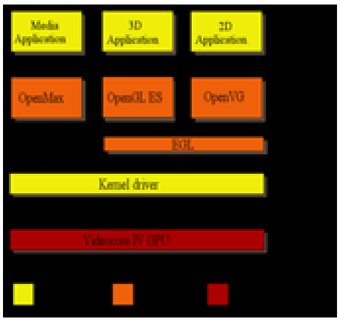
C. OS Install

Three different flavours of Linux are available officially: Pidora (based on Fedora); Archlinux (a DIY OS); and Raspian (Debian).

Raspian is the recommended OS for everyone new to the Raspberry Pi, but if we fancy trying out some of the others too, a new tool from the creators of Raspberry Pi has been released which allows you to choose from a selection of images during boot-up (including some of the others on this list).[48]

After cycling through several recommendations since just before the hardware was first made available, the Raspberry Pi Foundation created the New Out Of Box System (NOOBS) installer, and as of July 2013 suggests using it to install the Debian-derived Raspbian. The Foundation intends to create an application store website for people to exchange programs.

Raspbian is a Debian-based free operating system optimized for the Raspberry Pi hardware. It is the current recommended system, and was officially released in July 2012, although it is still in development. It is free software and maintained independently of the Raspberry Pi Foundation. It is based on ARM hard-float (armhf)-Debian 7 'Wheezy' architecture port with the LXDE desktop environment, but optimized for the ARMv6 instruction set of the Raspberry Pi. A minimum size of 2 GB SD card is required for Raspbian, but a 4 GB SD card or above is recommended. The downloaded Raspbian Wheezy image file has to be unzipped and then written to a suitable SD card, formatting it for use.



Procedure to system ready

OS install

Download latest Raspbian OS Raw Images 2014-01-07-wheezy-raspbian.zip(file name depends on release date) from http://www.raspberrypi.org/downloads. Download Win32DiskImager from http://sourceforge.net/projects/win32diskimager/files/latest/download. Insert SD card into your PC/Laptop. Run Win32DiskImager. Select Drive and locate .img file. Then press Write. Insert SD card into Raspberry pi and switch on. On setup Options screen, select option 3 "Enable Boot to Desktop/Scratch" .select "Desktop Log in as user 'pi' at the graphical desktop" and press enter. Press TAB key to select finish. Press enter. Select "yes" to reboot and press enter.

Default user name :pi
Password :raspberry

After OS installation, we need to install package. All installation carried out by opening LXTerminal(icon on Desktop).

Change KeyBaord Layout

After installation at first we should change the key board layout.

type on terminal

$ sudo leafpad /etc/default/keyboard

XKBLAYOUT="gb"
change to
XKBLAYOUT="us"

Change Password
For pi
$ sudo passwd pi

For root
$ sudo passwd root

Type old password, next type new password. And retype new password again for confirmation.

If we need to re-run the startup configuration options panel, use the following command:
$ sudo raspi-config

Clock Integration

There is no RTC in the Raspberry Pi. We have to attached a RTC with it. Pi recommended to use DS1307. System will update the time at the boot time. Entire time system will operate from virtual clock. Anytime we can read Hardware clock from any programming environment.

Follow the step to integrate RTC.

Step-01>Assembling the RTC DS1307

To enhance Rpi by RTC, we have to make the circuit and connect with Rpi through a 26 pin FRC cable. Care should be taken when assembling the circuit. No short-circuit should occurred.

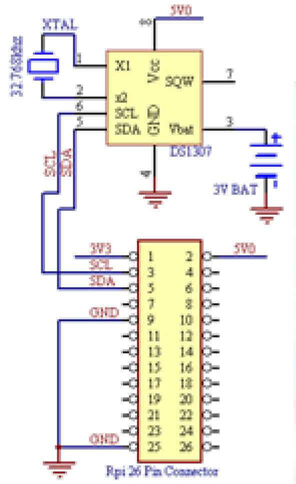


Fig.-4: RTC Connection

Step-02> Enable i2c bus

At terminal type

$ sudo leafpad /etc/modprobe.d/raspi-blacklist.conf

Fig.-4: RTC Connection

We will observe the following lines into the files :-

blacklist spi-bcm2708
blacklist i2c-bcm2708

put # at the beginning of the line. The lines should now look like :

blacklist spi-bcm2708
#blacklist i2c-bcm2708

Save and exit

Step-03>Load I2C module at boot time

Open terminal and type

$ sudo leafpad /etc/modules

Add the following lines at the end of the file:

i2c-dev
rtc-ds1307

Save and exit.

Step-04>Software Installation

Now, we required internet connection. Connect your Rpi with a router or internet sharing windows pc.

At Terminal, type -
$ sudo apt-get install python-smbus
press 'y' to continue installation.

Step-05>To make DS1307 clock loadable at startup

At Terminal, type -
$ sudo leafpad /etc/rc.local
Add the following lines befor exit0
echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-1/new_device
hwclock–s

Save, exit and system reboot.

Step-06>Update system clock: at Terminal Type

$ sudodate --set "5 Aug 2012 12:54"
The above date will be changed on current date, time.

Step-07>synchronize DS1307 with System clock

at Terminal Type
$ sudo hwclock --systohc--localtime
Or,
$ sudo hwclock --systohc--utc

Step-08>Read/Writeclock

at Terminal Type

Read system Date and Time:
$ date

Read DS1307 Date and Time:
$ sudo hwclock–show
Or,
$ sudo hwclock-r

Write Date and Time to DS1307 :
$ sudo hwclock -w

N.B. Square wave output (pin no 7 of DS1307) is disabled.

SPI BUS INSTALL

Step-01> Enable i2c bus

At terminal type
$ sudo leafpad /etc/modprobe.d/raspi-blacklist.conf
put # at the beginning of the line, The lines should now look like :
#blacklist spi-bcm2708
Save and exit

Step-02> Install Driver for SPI BUS

At terminal
$ sudo apt-get install python-dev
$ mkdir python-spi
$ cd python-spi

$ wget https://raw.github.com/doceme/ py-spidev/ master/setup.py

$ wget https://raw.github.com/doceme/py-spidev/master/spidev_module.c

$ sudo python setup.py install

Step-03> Test SPI BUS by Program

Open a python editor, either through IDLE, or just a text file, and type the following:

```
Import spidev
import time
spi = spidev.SpiDev()
spi.open(0,0)
while True:
resp = spi.xfer2([0x00])
print resp[0]
time.sleep(1)
```

SERIAL PORT(RS232) INSTALL

At terminal, type :

sudo apt-get install python-serial

Sample program for RS232

```
import serial
ser = serial.Serial("/dev/ttyAMA0")
ser.write("UART the Font")
read = ser.read()
print read
ser.close()
```

GPIO Access

```
# The required module to access
Import RPi.GPIO as GPIO

#set up GPIO using BCM numbering
GPIO.setmode(GPIO.BCM)

#setup GPIO using Board numbering
GPIO.setmode(GPIO.BOARD)

# set up GPIO output channel
GPIO.setup(11, GPIO.OUT)

# To pin Logic hi
GPIO.output(11,GPIO.HIGH)

# To pin Logic low
GPIO.output(11,GPIO.LOW)

# setup GPIO as Input
GPIO.setup(11,GPIO.IN,pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(11, GPIO.IN, pull_up_down = GPIO.PUD_UP)

# read Imput example
if(GPIO.input(23) ==1):
        print(“Button 1 pressed”)

# Edge detection
GPIO.wait_for_edge(11, GPIO.RISING)
        print(“Button 1 Pressed”)
GPIO.wait_for_edge(11, GPIO.FALLING)
        print(“Button 1 Released”)
```

sample program to access GPIO, Open IDLE or IDLE3 and type :

```
Import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setup(11, GPIO.OUT)
while True:
GPIO.output(11,GPIO.HIGH)
time.sleep(1)
GPIO.output(11,GPIO.LOW)
time.sleep(1)
```

Glade Installation

Glade is Rapid Application Development (RAD) tool. It is used to Design GUI easily and rapidly. To install Glade, command-

$ sudo apt-get install glade

Glade Icon is available in Program menu

For GUI programming we need gtk library. On terminal type -

$ sudo apt-get install python-gtk2

$ sudo apt-get install libgtk2.0-dev

MySQL install

MySQL is a database software from Oracle.

Command-

$ apt-get install mysql-workbench

AUTO START PROGRAM

Auto start program starts automatically after power up. The procedure as follows –

create a folder named “autostart” on desktop.

Create a blank file, named “test.desktop”. File name maybe changed, but extension should not.

Add the following line in the test.desktop file.

```
[Desktop Entry]
Encoding=UTF-8
Type=Application
Name=my_auto_test_program
Comment=
Exec= sudo python /home/pi/Desktop/my_program.py
StartupNotify=false
Terminal=false
Hidden=false
```

Save the file and put into autostart folder.

Open file manager. Go to /home/pi/.config. if .config not visible, select “show hidden file or floder” .Place the autostart folder into .config folder. Create a python program, named my_program.py and save to the desktop. Reboot. Now every time power up, our program will run.

V. METHODOLOGY

After the powering up, system boot from attached SD cards. Without SD cards system will not boot. Various linux distros are available for Raspberry pi. We select debian wheezy port. It is native os and comparatively easy to operate. After completion of system boot, our registered auto run program starts.

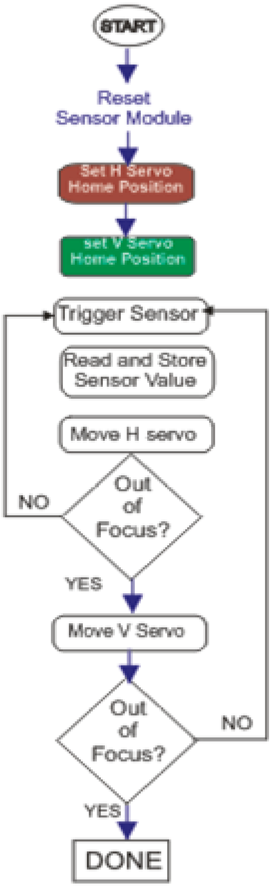
Step-1: send reset signal to the microcontroller which is attached with two servo motor, through Wi-Fi. The command controller send pulse so that, ultrasonic sensor can move to the Home position. The home position is predetermined. Servo is a mechanical device, so it takes few seconds to move the desired position. After completion of desired position, it sends an acknowledgement signal to the main unit

Step-2: Now system waits for user interaction. According to command system send command to move certain degree. The commands are pursed by the servo controller and drive the horizontal and vertical servo to get the desired position. Now trigger the Ultrasonic module. Module send eight pulse train and gets echo signal. The duration between send and receive pulse is send to the main system. The main system stores this raw data to database.

Step-3 on the occasion of customized search, system retrieved the raw data from database. Then, according to operation, system transform the data into desired format. The data can be used to simulate the model or to trigger the object.

VI. EXPERIMENTAL RESULT

Various experiments were conducted at different situation and environment. One of them is included here. The environment to be scanned was as follows



The following table shows a scan data

No.	Vertical Servo movement (degree)	Horizontal Servo Movement (Degree)	16 bit Timer value
1	0.0	10	32712
2	0.0	11	32712
3	0.0	12	32712
4	0.0	13	32690
5	0.0	14	32691
6	0.0	15	32690
7	0.0	16	32690
8	0.0	17	32691
9	0.0	18	32690

The resolution of horizontal servo movement is kept one degree. We can increase it, but takes more time to scan the object and to take the decision. If we observe carefully, there is no vertical movement because all values are same i.e. 0.0 degree. It means the table is showing one line scan data. Timer value is not to scale with time. It depends on crystal value and pre scalar value.

VII. CONCLUSIONS

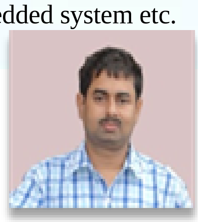
We have introduced a new approach to capture ultrasonic image by moving sonar sensor horizontally and vertically. This scan data is saved into a database. The stored data is analyzed by many algorithms and retrieved various parameter like object distance, type of object. Gripping position etc. Raw captured data is also fed to the 3D engine OPENGL and creates a 3D image on Monitor. It helps projection and movement plan of Robot. During many weeks of testing in repeatedly modified environments the robot did not collide even once, except for those collisions that were caused by software bugs or hardware problems.

Our system can easily work in real time, making it possible to digitize process and recognize each echo several times per second. Clearly, there are many ways in which we could try to improve our results, for instance by adjusting the pre-processing scheme, using a different classification method, or collecting larger data sets. We could also increase efficiency by using dedicated hardware for some of the pre-processing. However, we feel that the simplicity and robustness of this system are part of its appeal.

VIII. ACKNOWLEDGEMENTS

Funding is acknowledged from the P.C. Electronics & System with the project Batch Weighing System. Honorable Sir, Mr. Pranab Chandra seth was fully cooperated with us not only finance, giving technical supervision also.

Dr. Dhritinandan Kole was born in Hooghly district of West Bengal, India on 11th December, 1977. He had carried out B.Tech degree in Instrumentation Engg. From University of Calcutta, Kolkata, India in the year of 2000. After that he had completed his M. Tech degree from the same university in 2002. He is awarded with Ph.D degree in the year of 2012 in the field of Applied Physics (Instrumentation Engineering) from University of Calcutta. He has more than five years of teaching experience in the post of Senior Lecturer in MCKV Institute of Engineering, Howrah, and West Bengal, India. He had also engaged with research and development work for more than 7 years in Industrial sector like Simplex Infrastructure Limited, R&D section of IIMC, Kolkata, Pervcom Consulting Pvt Ltd etc. Presently he is working as R&D Executive in Eltron Wireless, Kolkata and also as consultant of Deep Micro System, Hooghly. His research interests include wireless sensor network, control and instrumentation, embedded system etc.



Sudip Chakraborty was born at Chhinamore in Hooghly district, West Bengal, India on 7th January 1977. He received the master degree in Computer science from Sikkim Manipal University in 2013. He is pursuing his Ph.D. in Artificial Intelligent in Robotics, Department of Computer science, Swami Vivekananda University, Sagar, (M.P.), India. His research interests include modeling and virtual simulation, autonomous Test Bench, autonomous security system etc. He is currently working as a Senior Engineer in the R&D Department at P.C. Electronics & system, Kolkata, since 2012, Mr. Chakraborty published / presented a research paper in the International Journals of high repute.

Susmita Bhunia was born at Dara in Paschim Medinipur district, West Bengal, India on 23rd January 1988. She completed her Diploma in Electronics &Telecommunication from Women's Polytechnic Chandannagar, India in the year 2012. Now, she is pursuing her B. Tech. Degree in Electronics & Telecommunication from Indian Institute of Engineering, India. She is currently working as an Assembling and testing Engineer at P.C. Electronics & system, Kolkata, since 2013. Her Interest area is embedded electronics.

REFERENCES

[1] Mittal, R.K &Nagrath, I.J "Robotics and Control", Tata McGraw-Hill Education Private Limited, ISBN-10: 0-07-048293-4. 2012, Page no- 71-111

[2] Saha, S.K "Introduction to Robotics", Tata McGraw-Hill Education Private Limited, ISBN-10: 0-07-066900-0. 2011, Page no- 113-121

[3] Niku, Saeed B. "Introduction to Robotics, Analysis, Systems, Applications", PHI Learning Private Limited, ISBN-978-81-203-2379-7, 2009, Page no- 113-121

[4] Klafter , Richard D. , Chmielewski, Thomas A., Negin, Michael "Robotic Engineering: An Integrated Approach", PHI Learning Private Limited, ISBN-978-81-203-0842-8,2009, Page no- 610-619

[5] Blankenship, J and Mishal, S "Robot Programmer's Bonanza",BPB Publications, ISBN-10:81-8333-290-0, 2008, Page no-65-72

[6] Appin Knowledge Solutions "Robotics",BPB Publications, ISBN-10:81-8333-212-9, 2007, Page no-189-215

[7] http://colinfahey.com/csharp_wrapper_for_opengl/csharp_wrapper_for_opengl.html

[8] https://sites.google.com/site/leniel/blog/RobotArmOpenGLCSharp.zip?attredirects=0

[9] http://www.glprogramming.com/blue/ch03.html

[10] http://www.glprogramming.com/blue/ch03.html#id46859

[11] http://www.movesinstitute.org/~mcdowell/mv4202/notes/lect9.pdf

[12] https://www.cs.duke.edu/courses/cps124/fall09/notes/05_pipeline/cox_04transformations.pdf

[13] http://en.wikipedia.org/wiki/Denavit–Hartenberg_parameters

[14] http://www.dbgidoon.com/academic/resultmay13/synopsis-format.pdf

[15] http://www.oocities.org/raj_kamat/rkamath.pdf

[16] http://vkghaz.blogspot.in/2013/07/blog-post_29.html

[17] http://robots.freehostia.com/Circuits/BatteryMonitor/BatteryMonitor.html

[18] http://www.bakerhughes.com/capabilities/carbon-capture-and-storage-ccs/co2-monitoring event monitoring globe> app2.nea.gov.sg
[19] www.sciencedaily.com
[20] www. robotics.youngester.com
[21] http://gtri.gatech.edu/img/facilities-high-power-2.jpg
[22] http://www.enterpriseconnect.gov.au/ecservices/tper/PublishingImages/Images_large/Testing-and-Analytical/high-voltage-testing-equipment_LARGE.jpg
[23] http://www.tauscher.com/assets/images/news2011_013.jpg
[24] http://dx.com/p/ultrasonic-sensor-distance-measuring-module-138563
[25] http://www.blogcdn.com/www.engadget.com/media/2012/05/05-31-2012dsc0213300-1338512994.jpg
[26] https://www.sparkfun.com/products/10414
[27] http://theseus.fi/bitstream/handle/10024/32108/Valkeapaa_Tommi.pdf?sequence=1
[28] http://en.wikipedia.org/wiki/Automation
[29] http://www.intechopen.com/download/get/type/pdfs/id/6966
[30] http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1356187/
[31] http://www.haefely.com/pdf/scientific/e1-37.pdf
[32] http://projectsera.eu/publications/papers/EM10_L_Ganster02.pdf
[33] http://en.wikipedia.org/wiki/Analog_to_Digital
[34] http://en.wikipedia.org/wiki/Digital_to_analog_converter
[35] http://en.wikipedia.org/wiki/Microcontroller
[36] http://upload.wikimedia.org/wikipedia/commons/1/18/PIC18F8720.jpg
[37] http://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=1045&context=honors
[38] http://crypto.stanford.edu/ portia/papers/pgbgp++.pdf
[39] http://searchnetworking.techtarget.com/definition/autonomous-system
[40] https://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6504448
[41] http://www.cs.unm.edu/~karlinjf/papers/dimacs08.pdf
[42] http://www.jameco.com/Jameco/workshop/howitworks/how-servo-motors-work.html
[43] http://en.wikipedia.org/wiki/Raspberry_Pi
[44] https://www.google.co.in/search?biw=1366&bih=684&tbm=isch&sa=1&q=radio+wave+pix&oq=radio+wave+pix&gs_l=img.3...11285.21434.0.21677.15.14.0.0.0.0.361.1848.2-2j4.6.0....0...1c.1.36.img..9.6.1847.3uVybKcXos0
[45] http://pdf.aminer.org/000/361/096/a_method_of_modelling_application_to_an_ultrasonic_sensor.pdf
[46] http://in.mouser.com/new/Digi-International/digixbeeWIFI/
[47] http://en.wikipedia.org/wiki/Raspberry_Pi
[48] http://www.makeuseof.com/tag/7-operating-systems-you-can-run-with-raspberry-pi/

For I2C bus :-

http://quick2wire.com/articles/physical-python-part-1/

http://www.bristolwatch.com/rpi/rpi_i2c_ds1307.htm

http://learn.adafruit.com/adding-a-real-time-clock-to-raspberry-pi/wiring-the-rtc

http://www.raspberrypi.org/phpBB3/viewtopic.php?t=16817&p=317041

http://www.raspberrypi.org/forum/viewtopic.php?f=44&t=31420

http://www.raspberrypi.org/phpBB3/viewtopic.php?f=44&t=16218&p=269238&hilit=DS1307#p269238

http://binerry.de/post/26685647322/raspberry-pi-and-i2c

http://rayshobby.net/phpBB3/viewtopic.php?f=28&t=65

http://docs.slackware.com/howtos:hardware:syncing_hardware_clock_and_system_local_time?do=export_pdf

for Serial Port

http://www.elinux.org/Serial_port_programming

http://www.hobbytronics.co.uk/raspberry-pi-serial-port

for GPIO

http://www.rpiblog.com/2012/09/using-gpio-of-raspberry-pi-to-blink-led.html

http://makezine.com/projects/tutorial-raspberry-pi-gpio-pins-and-python/