THE NEOTIA UNIVERSITY

ज्ञानम् आत्म प्रदीपाय

DEPARTMENT OF ROBOTICS & AUTOMATION

Internet of Things

LAB MANUAL

THE NEOTIA UNIVERSITY
DEPARTMENT OF ROBOTICS & AUTOMATION

EXPERIMENT NO.: 1

NAME OF THE EXPERIMENT: Study & Survey of various IoT Boards available in the market.

OBJECTIVE: To learn about various Boards available in Market for IoT.

THEORY:

What is the Internet of Things?

The Internet of Things connects the physical world to the Internet so that you can use data from devices to increase productivity and efficiency. Connecting things to the Internet is possible because different connectivity options are widely available, the cost of connecting is declining, and more devices are capturing data. All kinds of things are being used in IoT applications including consumer products such as refrigerators, security cameras, and cable set-top boxes; industrial systems such as conveyor belts and manufacturing equipment; and commercial devices such as traffic signals and smart meters. Any device that can be powered on could be part of an IoT application.

How does the Internet of Things work?

The Internet of Things was coined by Kevin Ashton, a British technology pioneer working on radio-frequency identification (RFID) who conceived a system of ubiquitous sensors connecting the physical world to the Internet. Today, the Internet of Things connects physical devices embedded with electronics, software, sensors, and actuators to the cloud and to each other. Devices communicate through different protocols and many, such as MQTT, were designed to tolerate intermittent connections and reduce network bandwidth requirements. All IoT communication must be secured using preventive security mechanisms and best practices, like device identity management, encryption, and access control as well as device auditing and monitoring. Although connected things, the Internet, and secure connectivity are required to create IoT applications, the value is in closing

the gap between the physical and digital world in self-reinforcing and self-improving systems.

Why is the Internet of Things important?

The Internet of Things is important because it makes previously unusable data available. IoT applications tap into device data and let you visualize, explore, and build sophisticated analytics such as machine learning in the cloud. IoT applications can also run on devices so they can respond in real-time as events unfold. For example, a predictive modelfor scene detection analysis could run on a security camera and, when the camera sees suspicious activity, it could send an alert.

IoT applications are deployed across a wide range of use cases including connected homes, connected vehicles, healthcare, industrial, retail, and many more. Connected homes are safer, cleaner, and more energy efficient. For example, Amway has successfully launched its first Internet-connected product—the Atmosphere Sky Air Treatment System— using AWS IoT to build policies and security throughout the entire architecture.

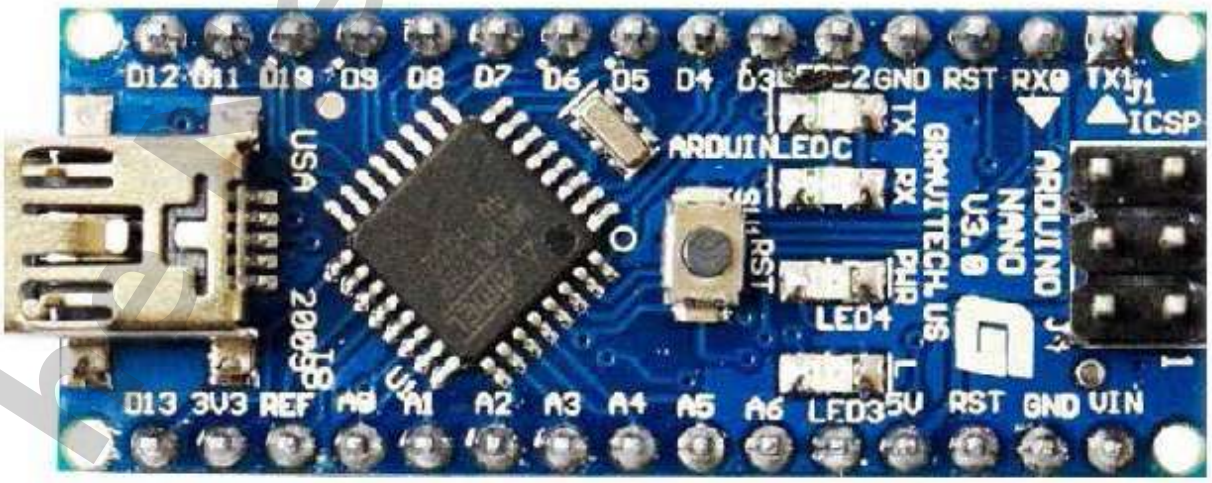
IoT applications are used for industrial use cases to improve processes, create new revenue streams, reduce costs, and increase safety by analyzing data from equipment, creating predictive maintenance models, and keeping devices up to date with over-the-air updates. For example, EMS uses AWS IoT to detect fuel leaks early to minimize environmental impact. EMS delivered a 500% ROI using AWS IoT.

a) Arduino Uno:

The Arduino Uno R3 is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which six can be used as PWM outputs), six analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. Simply connect it to a computer via a USB cable or power it with a AC-to-DC adapter or battery to get started.

b) Arduino Nano:

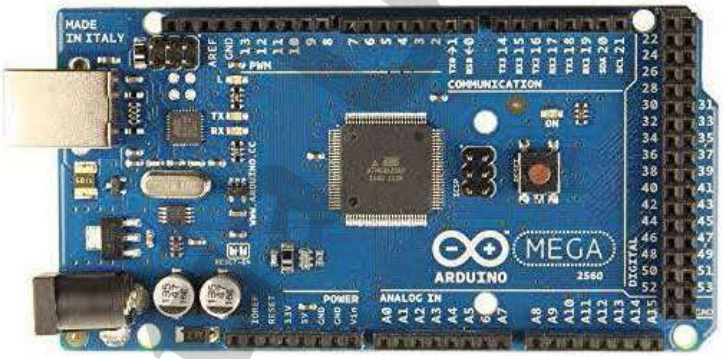
The Arduino Nano is a small, breadboard-friendly board based on the ATmega328. The microcontroller has more or less the same functionality of the Arduino Duemilanove, but in a different package. It lacks a DC power jack, and works with a Mini-B USB cable instead of a standard one.



c) Arduino Mega 2560 Rev. 3 Microcontroller Board:

It is based on the Atmel ATmega2560 8-bit microcontroller (MCU). Arduino Mega 2560 features 54 digital input/output pins (15 of which can be used as PWM outputs) and 16 analog inputs. This Arduino MCU board also includes 4 UARTs (hardware serial ports), a 16MHz crystal oscillator, a USB connection, a power jack, an In-Circuit Serial Programming (ICSP) header, and a reset button.

Mega 2560 includes everything the user needs to support the MCU. The user can get started by connecting the Mega 2560 to a computer with a USB cable or by powering it with an AC-to-DC adapter or battery. Arduino Mega 2560 board is compatible with most shields designed for the Uno and former boards Duemilanove or Diecimila. Mega 2560 is an update to the earlier Arduino Mega board.



d) ESP8266:

The low-cost microcontroller is one of the most popular prototyping solutions on the market.

Here are some of its crucial advantages:

- Wi-Fi support (which basically means IoT software development companies can connect the board to the Internet without an intermediary gateway that, for example, enables connectivity in Bluetooth solutions);
- Flexibility for rapid development with either native IDEs such as ESPlorer (provides an opportunity to code in a scripting language like LUA) or Arduino (C/C++ environment);
- Out-of-the-box support for the MQTT protocol (helps vendors integrate IoT hardware into complex device networks comprised of thousands of sensors).

A slightly limited RAM (160 Kb) is one of ESP8266's few drawbacks. It's more than enough to address typical IoT dev needs; however, the MCU might not handle heavy data processing.

e) Raspberry Pi:

Raspberry Pi is a credit card sized computing platform which runs on linux platform. This board features USB ports for keyboard and mouse, a HDMI port for display and an Ethernet port for wired internet connectivity. However for wireless connectivity we need to go for Wifi adapters which we can get it from the market without any trouble. The vast community makes it perfect to develop IOT projects since we get plenty of hands when we stumble.

CONCLUSION: Learn about Various Boards available in Market for IoT.

THE NEOTIA UNIVERSITY
DEPARTMENT OF ROBOTICS & AUTOMATION

EXPERIMENT NO.: 2

NAME OF THE EXPERIMENT: Study & Survey of various IoT Platforms available in the market.

OBJECTIVE: To learn about Various Platforms available in Market for IoT

THEORY:

What do you mean by IoT platforms?

The IoT platforms are suites of components those help to setup and manage the internet connected devices. A person can remotely collect data, monitor and manage all internet connected devices from a single system. There are a bunch of IoT platforms available online but building an IoT solution for a company is all depend on IoT platform host and support quality.

Here, we have compiled some best and well-known Open source Internet of things (IoT) platforms. Most of these platforms provide API to support the Linux hacker board such as Raspberry Pi, Arduino, and BeagleBone.

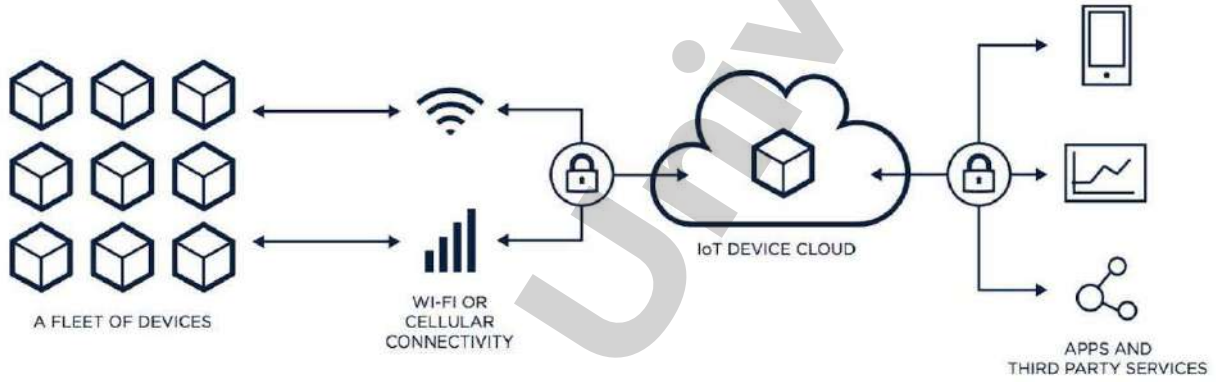
How to Choose Your IoT Platform — Should You Go Open-Source?

1. A complete IoT system needs hardware, such as sensors or devices. These sensors and devices collect data from the environment (e.g. a moisture sensor) or perform actions in the environment (e.g. watering crops).
2. A complete IoT system needs connectivity. The hardware needs a way to transmit all that data to the cloud (e.g. sending moisture data) or needs a way to receive commands from the cloud (e.g. water the crops now). For some IoT systems, there can be an intermediate step between hardware and connecting to the cloud, such as a gateway or router.
3. A complete IoT system needs software. This software is hosted in the cloud (what's the cloud?) and is responsible for analyzing the data it's collecting from the sensors and making decisions (e.g. knowing from moisture data that it just rained and then telling the irrigation system not to turn on today).
4. Finally, a complete IoT system needs a user interface. To make all of this useful, there needs to be a way for users to interact with the IoT system (e.g.

a web app with a dashboard that shows moisture trends and allows users to manually turn irrigation systems on or off).

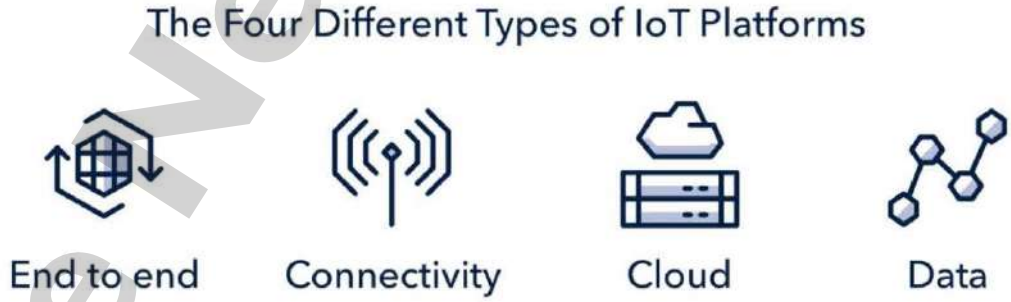
IoT platforms are the support software that connects everything in an IoT system. An IoT platform facilitates communication, data flow, device management, and the functionality of applications.

What is an IoT Platform?



In the simplest terms, an IoT platform is an integrated service that offers you the things you need to bring physical objects online. It needs to be capable of supporting millions of simultaneous device connections and easily allow you to configure your devices for machine-to-machine communication.

IoT Platform Types:



End-to-end IoT Platforms

Fundamentally, end-to-end IoT platforms provide the hardware, software, connectivity, security, and device management tools to handle millions of

concurrent device connections. It also provides all the managed integrations you need — OTA firmware updates, device management, cloud connection, cellular modem, etc — to connect and monitor a fleet of devices online.

Connectivity Management Platforms

These platforms offer low power and low cost connectivity management solutions through Wi-Fi and cellular technologies. This can range from connectivity hardware, cellular networks, and data routing features.

IoT Cloud Platforms

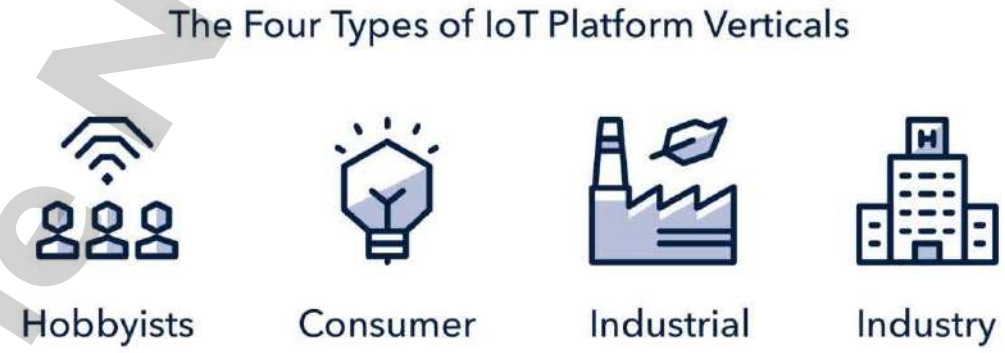
Cloud platforms aim to get rid of the complexity of building your own complex network stack and offer the backend services (plus other services) to monitor and tack millions of simultaneous device connections.

Data Platform

Of course, every type of IoT platform deals with data in some way. But these IoT data platforms combine many of the tools you need to route device data and manage / visualize data analytics.

IoT Platform Verticals:

Of course, categorizing these IoT platforms under a single category is probably presenting it to simply. All these platforms tend to offer more solutions and that can't be broken down into a single category. So you need to examine what they are offering, and who they are offering it to.



Hobbyists

- Prototyping Solutions
- Development Kits
- DIY solutions

Consumer Electronics

- Utilities
- Home automation
- Wearables
- Anything with Alexa-involved

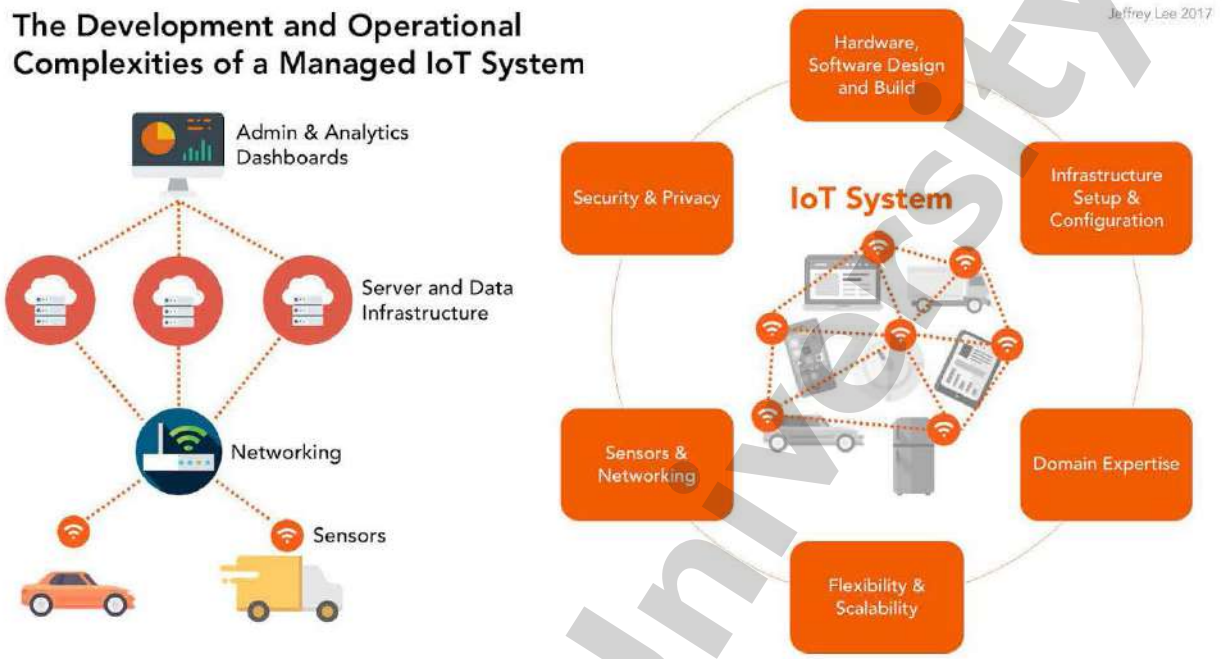
Industrial IoT (IIoT) Solutions

- Smart factory warehousing applications
- Predictive and remote maintenance.
- Industrial security systems
- Asset tracking and smart logistics
- Energy optimization
- Transportation monitoring
- Connected logistics

Industry-Driven

- Agriculture
- Healthcare
- Transportation
- Smart Cities
- Energy

What you should look for when examining platforms



When examining these platforms, these are the types of things you should be examining or looking for based on your solution:

1. Connectivity
How well does the vendor's network coverage fit your business's current and future initiatives?

2. Method of connectivity
What type of connectivity do you need? Do you need a Wi-Fi or cellular solution for your IoT product? You need to assess these needs and see how the vendor can address them.

3. Market Longevity
How long has the IoT platform been in business? The IoT space is relatively new, but quickly building. Finding an IoT platform that has been offering services for 4+ years is usually ideal.

4. Type of service
How does the IoT platform describe and sell themselves? Some services are purely connectivity platforms, while others are end-to-end solutions that offer the hardware, software, and connectivity. You need to assess what your business needs. How will your needs change over time?

5. Geographic coverage
Do they provide embedded sim with global support? Does the IoT platform cover the regions your business needs?

6. Data plan

Does the vendor offer a fair data plan? You'll want the ability to pause or suspend your data services at any time and the ability to control how much data that is used.

7. Security / Privacy

Examine how they've deal with security and privacy issues in the past and review their security content. You need to assess how their platform combats security issues and how it abstracts the complexity away from you.

8. Managed Integrations / API Access

How does the vendor integrate all the complex stuff that you need for IoT — like cellular modems, carrier / sim cards, device diagnostics, firmware updates, cloud connections, security, application layer, RTOS — into a simple package that your engineering team won't have to worry about?

9. Data Access

How will you integrate the data acquired through the IoT platform with your enterprise back ends and current cloud service? What do you plan to do with this data? Does the service match those needs?

10. Domain Experts / Engineering Services / Partner Collaboration

IoT deployments are complex and you need a partner who can help you through the product development process. Does the IoT platform help you service those needs?

11. IoT Ecosystem

Take the time to learn and understand the relationships between the services the IoT platform offers. This will help you learn how their services can help you build your product. If you don't understand, talk to a sales representative.

12. IoT Roadmap

IoT platforms are only expanding from here on out. Does the IoT platform's roadmap match your organization's needs? Do they plan to expand into connectivity, data, hardware? How does that help you?

13. Hardware

Does the vendor offer any off-the-shelf applications, developer kits, or starter packages for the specific use case you are targeting? You will likely have to do some customization, but not having to start from scratch can save significant time and effort.

14. Hardware Agnosticism

Hardware Agnosticism is code for "we handle the software only, meaning you're totally on your own to build all the hard stuff." Which means, don't fall for it.

15. Device Management

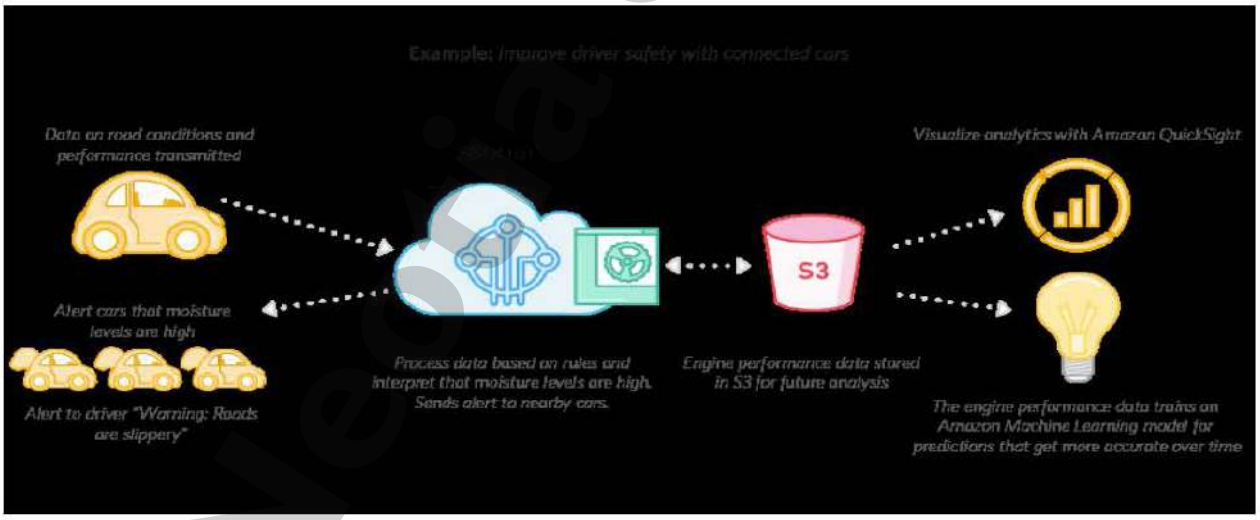
How does the vendor allow you to monitor, segment, and manage IoT devices that are out in the field?

16. OTA Firmware Updates

How does the vendor allow you to send updates and fix bugs on your devices remotely? It is a simple or complex process? (hint: you want the simple solution).

Here's a list of 10 of the best platforms available today.

1. **AMAZON WEB SERVICES IOT PLATFORM**



Amazon dominates the consumer cloud market. They were the first to really turn cloud computing into a commodity way back in 2004. Since then they've put a lot effort into innovation and building features, and probably have the most comprehensive set of tools available.

It's an extremely scalable platform, claiming to be able to support billions of devices, and trillions of interaction between them.

Pricing is based on messages sent and received by AWS IoT. Each IoT interaction can be thought of as a message between a device and a server. Amazon charges per

million messages sent or received. There are no minimum fees, and you won't get charged for messages to the following AWS services:

- Amazon S3
- Amazon DynamoDB
- AWS Lambda
- Amazon Kinesis
- Amazon SNS
- Amazon SQS

They also have a software development kit (SDK) to help developer build applications to run on AWS.
While Amazon probably has the most comprehensive service of all the cloud providers, it can also be quite expensive.

2. MICROSOFT AZURE IOT HUB

Microsoft is taking their Internet of Things cloud services very seriously. They have cloud storage, machine learning, and IoT services, and have even developed their own operating system for IoT devices. This means they intend to provide a complete IoT solution provider. The pricing is done in 4 tiers based on how much data your devices will generate. Below 8,000 messages per unit per day is free. It does get complicated when you start to integrate with other Microsoft services, but they have a great pricing calculator to help you out. Like Amazon, Google, Oracle and IBM, Microsoft also has some other cool services you can use on their could platform. These include things like machine learning a data analytics so you can build some really cool applications. check out our article here for more info.

3. IBM WATSON IOT PLATFORM

IBM is another IT giant trying to set itself up as an Internet of Things platform authority. They try to make their cloud services as accessible as possible to beginners with easy apps and interfaces. You can try out their sample apps to get a feel for how it all works. You can also store your data for a specified period, to get historical information from your connected devices.
Pricing works on three main metrics:
1. Data Exchanged
2. Data Analyzed
3. Edge Data Analyzed

You'll get 100 MB of each for free every month, so you can try it out.
IBM Watson also offers some cool security possibilities based on machine learning and data science. I wrote about some of them here.

4. GOOGLE CLOUD PLATFORM

Search giant Google is also taking the Internet of Things very seriously. They claim that "Cloud Platform is the best place to build IoT initiatives, taking

advantage of Google's heritage of web-scale processing, analytics, and machine intelligence".
Their focus is on making things easy and fast for your business, where instant information is expected. And, offer "Google grade" security. Using this platform also lets you take advantage of Google's private global fiber network.
Pricing on Google Cloud is done on a per-minute basis. It is usually cheaper than Amazon Web Services and even has a price comparison tool to show you how much you'll save. But doesn't have same extensive tools and documentation.
Like Microsoft, Google also has its own IoT operating system (based on Android).

5. ORACLE

Oracle is a platform as a service provider that seems to be focusing on manufacturing and logistics operations. They want to help you get your products to market faster. Pricing for Oracle is calculated per device. There is a set number of messages per device, per month, with an additional cost if you go over this number.

WHICH ONE SHOULD YOU GO WITH?

This is a tough question. There is no best IoT cloud platform, and ultimately it will depend on the specific needs of your business. At the moment Amazon is the most established in this field but could be expensive. If you just want to test out some ideas, go with a provider that offers a free tier. You'll be able to get a feel for how it works, the pros and cons, and what features you might need in the future. If you still aren't sure, contact a development team that has experience in building the type of systems you need. They will know the ins-and-outs of every platform, and will easily be able to recommend the perfect IoT cloud platform to take your business to the next level.

CONCLUSION:

Learn about popular Platforms available in the market which are used for IoT.

THE NEOTIA UNIVERSITY
DEPARTMENT OF ROBOTICS & AUTOMATION

EXPERIMENT NO.: 3
NAME OF THE EXPERIMENT: Interfacing sensors and actuators with Arduino.

Sensors

- IR sensor
- PIR sensor
- Ultrasonic sensor
- Temperature and humidity (DHT11) sensor
- Temperature (LM35) (analog) sensor

Actuators

- DC motor
- Stepper motor

OBJECTIVE:

1. To know about various sensors available in the market.

2. To know about various actuators available in the market.

3. To Know how to work with arduino. (EXPERIMENT NO.-3)

THEORY:

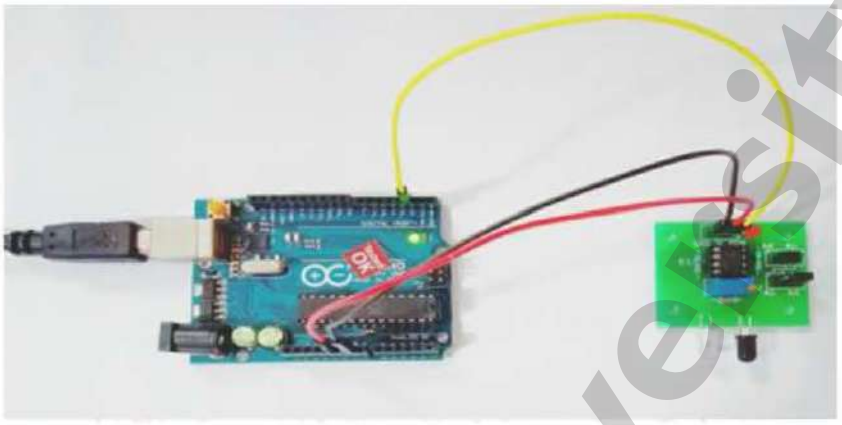
HARDWARE REQUIREMENT:

Arduino Uno, 9v adaptor, USB cable, IR sensor, PIR sensor, Ultrasonic sensor, Temperature and humidity (DHT11) sensor, Temperature (LM35) (analog) sensor, DC motor, Stepper Motor, Breadboard.

SOFTWARE REQUIREMENT:

Windows XP OS, Arduino 1.8.8 IDE

a) Interfacing IR sensor with Arduino



CONNECTION:

- IR Sensor has three pins(GND,V,OUT).
- It works under 5v. Connect V pin of sensor with Arduino 5v
- GND pin of IR connected to GND of Arduino
- Interface OUT pin of sensor with 2 nd pin of Arduino

b) Interfacing of PIR Sensor with Arduino

Operating Principle of PIR sensor:

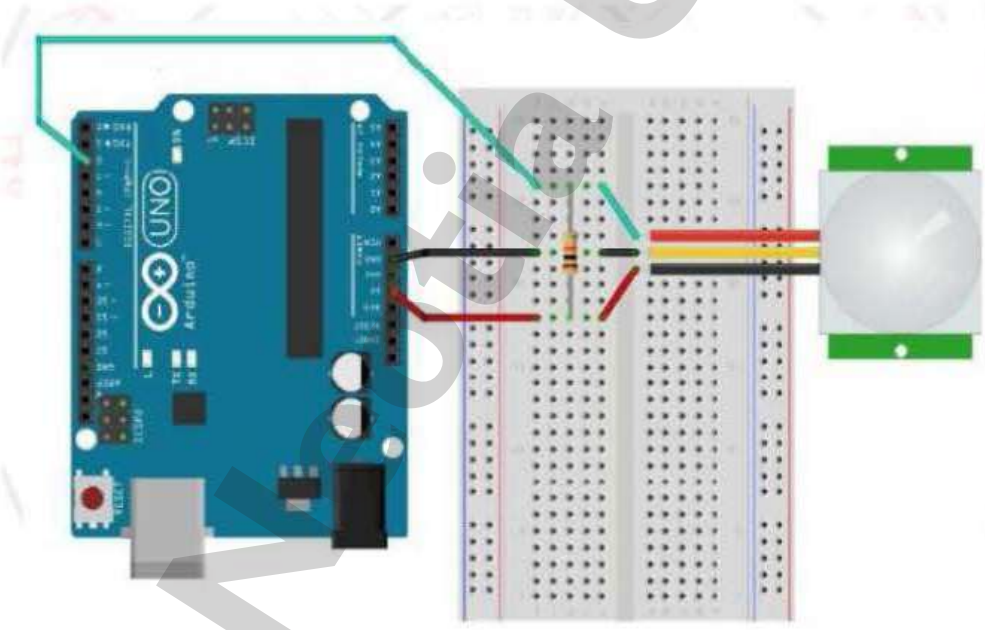
The modern studies in the field of quantum physics tells us the fact each and every object when it is placed at a temperature above absolute zero, emits some energy in the form of heat and this heat energy is in fact the form of infrared radiations. So an other question comes into mind that why our eyes can't see these waves? It is because that these waves have infrared wavelengths and his wavelength is invisible to human eyes. if you want to detect these waves then, you have to design a proper electronic circuit.

If you see closely the name of PIR sensor which is Passive Infrared Sensor. Passive elements are those elements that don't generate their own voltages or energy. They just only measures things. So we can say that this sensor is a passive infrared sensor and it doesn't generate anything by itself. It is only capable to measure the rediations emitted by other objects around it. It measures those raditions and do some desired calculations on them.

PIR sensor have total 3 pins. The configuration of each pin is shown in the image given below:

1. Pin#1 is of supply pin and it is used to connect +5 DC voltages.
2. Pin#2 is of output pin and this pin is used to collect the output signal which is collected by PIR sensor.
3. Pin#3 is marked as GND pin. This pin is used to provide ground to internal circuit of PIR sensor.
This whole configuration is also shown in the image given below:

The pin configuration of a PIR sensor is shown in the image given above. Since we have to interface the PIR sensor with Arduino micro controller. The image showing the interfacing of PIR sensor with Arduino is shown below as:



C) Interfacing Ultrasonic Sensor with Arduino:

An Ultrasonic Sensor is a device that measures distance to an object using Sound Waves. It works by sending out a sound wave at ultrasonic frequency and waits for it to bounce back from the object. Then, the time delay between transmission of sound and receiving of the sound is used to calculate the distance.
It is done using the formula Distance = (Speed of sound * Time delay) / 2

We divide the distance formula by 2 because the sound waves travel a round trip i.e from the sensor and back to the sensor which doubles the actual distance.

The HC-SR04 is a typical ultrasonic sensor which is used in many projects such as obstacle detector and electronic distance measurement tapes. In this Instructable I'll teach you how to interface the HC-SC04 with an Arduino Uno.

To interface an Ultrasonic Sensor with an Arduino and view the distance on the serial monitor you'll need:

- Arduino Uno
- HC-SR04 Module
- BreadBorad
- Jumper wires

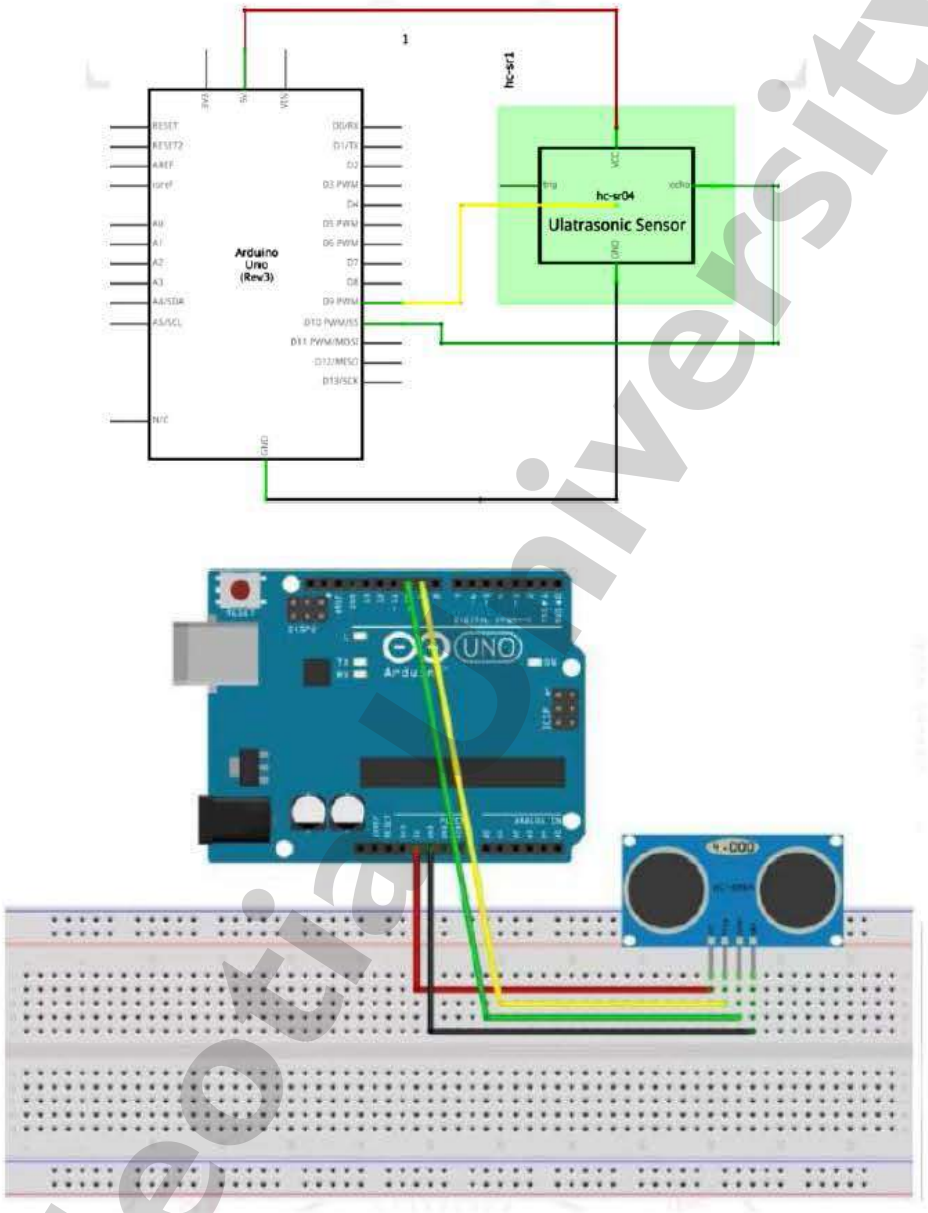
The HC-SR04 is an ultrasonic ranging module. This economical sensor provides 2cm to 400cm of non-contact measurement functionality with a ranging accuracy that can reach up to 3mm. Each HC-SR04 module includes an ultrasonic transmitter, a receiver and a control circuit.

There are Four Pins on the HC-SR04. They are :

- Vcc (5V supply)
- Gnd (Ground)
- Trig (Trigger)
- Echo (Receive)

The key features to be noted are:

- Operating Voltage: 5V DC
- Operating Current: 15mA
- Measure Angle: 15°
- Ranging Distance: 2cm - 4m



The connections are as follows:

- Vcc to 5V Pin of the Arduino.
- Gnd to Gnd Pin of the Arduino.
- Trig to Digital Pin 9 .
- Echo to Digital Pin 10.

d) DHT11 interfacing with arduino:

DHT11 sensor is used to measure the temperature and humidity. It has a resistive humidity sensing component and a negative temperature coefficient (NTC). An 8

bit MCU is also connected in it which is responsible for its fast response. It is very inexpensive but it gives values of both temperature and humidity at a time.

Specification of DHT11

- It has humidity range from 20 to 90% RH
- It has temperature range from 0 – 50 C
- It has signal transmission range of 20 m
- It is inexpensive
- It has fast response and it is also durable

DHT11 Pin out:

- The first pin of the DHT11 is vcc pin.
- The second pin of the DHT is Data pin.
- The third pin is not used.
- The fourth pin of the DHT sensor is ground.

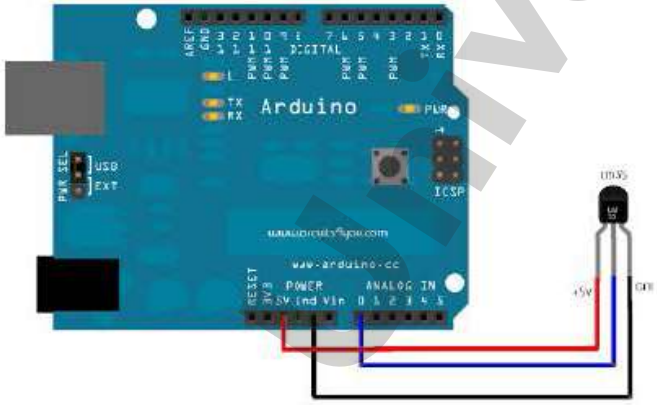
First of all connect the ground and the VCC of the DHT11 temperature and humidity sensor to the ground and 5v of the Arduino Then connect the data pin of the DHT11 sensor to the pin 2 of the Arduino.
Installing the DHT11 Library:

To run the following code in Arduino IDE you will first have to install the DHT library in you Arduino directory.

e) LM35 Temperature Sensor (Analog sensor) Interfacing with Arduino:

LM35 is a precision temperature sensor with its output proportional to the temperature (in oC). With LM35, temperature can be measured more accurately than with a thermistor. It also possess low self heating and does not cause more than 0.1 oC temperature rise in still air. The operating temperature range is from -55°C to 150°C. The output voltage varies by 10mV in response to ambient temperature, its scale factor is 0.01V/ oC.

Connection Diagram:



temp = (1.1 * analogRead(tempPin) * 100.0) / 1024;
To change aRef to 1.1V, you use the command "analogReference(INTERNAL);"

f) Interfacing of DC motors with Arduino:

A DC motor (Direct Current motor) is the most common type of motor. DC motors normally have just two leads, one positive and one negative. If you connect these two leads directly to a battery, the motor will rotate. If you switch the leads, the motor will rotate in the opposite direction.

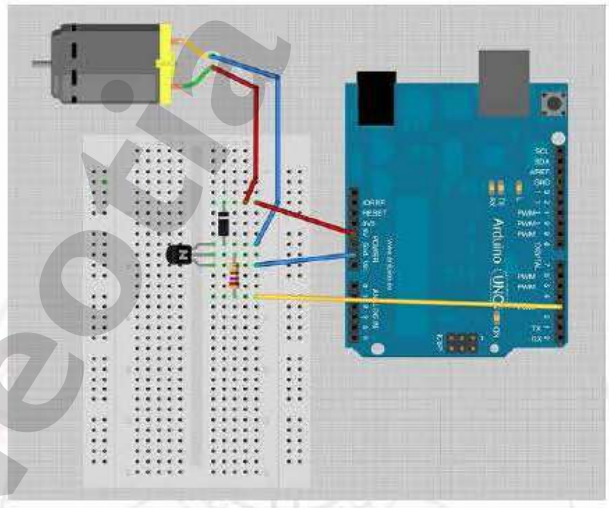
Warning − Do not drive the motor directly from Arduino board pins. This may damage the board. Use a driver Circuit or an IC.
We will divide this chapter into three parts −

- Just make your motor spin
- Control motor speed
- Control the direction of the spin of DC motor

Components Required:
You will need the following components −

- 1x Arduino UNO board
- 1x PN2222 Transistor
- 1x Small 6V DC Motor
- 1x 1N4001 diode
- 1x 270 Ω Resistor

Procedure:
Follow the circuit diagram and make the connections as shown in the image given below.



Precautions:

Take the following precautions while making the connections.

- First, make sure that the transistor is connected in the right way. The flat side of the transistor should face the Arduino board as shown in the arrangement.
- Second, the striped end of the diode should be towards the +5V power line according to the arrangement shown in the image.

Code to Note:

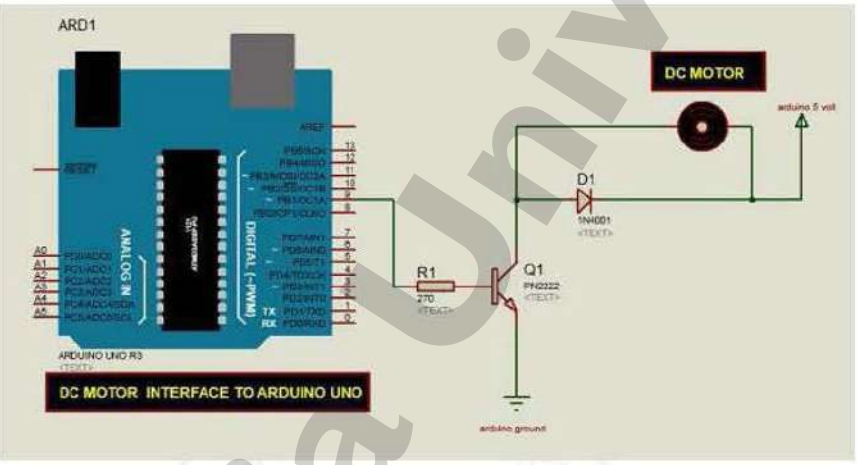
The transistor acts like a switch, controlling the power to the motor. Arduino pin 3 is used to turn the transistor on and off and is given the name 'motorPin' in the sketch.

Result

Motor will spin in full speed when the Arduino pin number 3 goes high.

Motor Speed Control

Following is the schematic diagram of a DC motor, connected to the Arduino board.



g) Interfacing of Stepper motor with Arduino:

The A4988 is a microstepping driver for controlling bipolar stepper motors which has built-in translator for easy operation. This means that we can control the stepper motor with just 2 pins from our controller, or one for controlling the rotation direction and the other for controlling the steps.

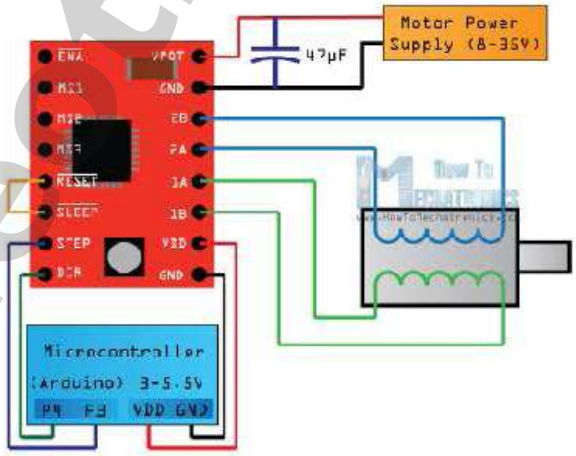
The Driver provides five different step resolutions: full-step, haft-step, quarter-step, eight-step and sixteenth-step. Also, it has a potentiometer for adjusting the current output, over-temperature thermal shutdown and crossover-current protection.

Its logic voltage is from 3 to 5.5 V and the maximum current per phase is 2A if good addition cooling is provided or 1A continuous current per phase without heat sink or cooling.

Minimum Logic Voltage:	3V
Maximum Logic Voltage:	5.5 V
Continuous current per phase:	1 A
Maximum current per phase:	2 A
Minimum Operating Voltage:	8 V
Maximum Operating Voltage:	35 V

A4988 Stepper Driver Pinout:

Now let's close look at the pinout of the driver and hook it up with the stepper motor and the controller. So we will start with the 2 pins on the button right side for powering the driver, the VDD and Ground pins that we need to connect them to a power supply of 3 to 5.5 V and in our case that will be our controller, the Arduino Board which will provide 5 V. The following 4 pins are for connecting the motor. The 1A and 1B pins will be connected to one coil of the motor and the 2A and 2B pins to the other coil of the motor. For powering the motor we use the next 2 pins, Ground and VMOT that we need to connect them to Power Supply from 8 to 35 V and also we need to use decoupling capacitor with at least 47 µF for protecting the driver board from voltage spikes.



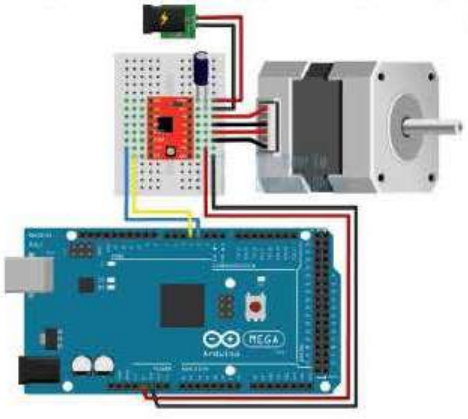
The next two 2 pins, Step and Direction are the pins that we actually use for controlling the motor movements. The Direction pin controls the rotation direction of the motor and we need to connect it to one of the digital pins on our microcontroller, or in our case I will connect it to the pin number 4 of my Arduino Board.

With the Step pin we control the mirosteps of the motor and with each pulse sent to this pin the motor moves one step. So that means that we don't need any complex programming, phase sequence tables, frequency control lines and so on, because the built-in translator of the A4988 Driver takes care of everything. Here we also need to mention that these 2 pins are not pulled to any voltage internally, so we should not leave them floating in our program.

Next is the SLEEP Pin and a logic low puts the board in sleep mode for minimizing power consumption when the motor is not in use. Next, the RESET pin sets the translator to a predefined Home state. This Home state or Home Microstep Position can be seen from these Figures from the A4988 Datasheet. So these are the initial positions from where the motor starts and they are different depending on the microstep resolution. If the input state to this pin is a logic low all the STEP inputs will be ignored. The Reset pin is a floating pin so if we don't have intention of controlling it with in our program we need to connect it to the SLEEP pin in order to bring it high and enable the board.

MS1	MS2	MS3	Resolution
LOW	LOW	LOW	Full Step
HIGH	LOW	LOW	Halft Step
LOW	HIGH	LOW	Quarter Step
HIGH	HIGH	LOW	Eighth step
HIGH	HIGH	HIGH	Sixteenth Step

Circuit Schematic:



CONCLUSION:

Learn to interface various sensors with Arduino, to interface various actuators with Arduino and to program in Arduino.

THE NEOTIA UNIVERSITY
DEPARTMENT OF ROBOTICS & AUTOMATION

EXPERIMENT NO.: 4

NAME OF THE EXPERIMENT: Interfacing sensors and actuators with Raspberry Pi 3.

Sensors

- IR sensor
- PIR sensor
- Ultrasonic sensor
- Temperature and humidity (DHT11) sensor

Actuators

- DC motor

OBJECTIVE:

1. To know about various sensors available in the market.

2. To know about various actuators available in the market.

3. To know how to work with Raspberry Pi 3.

THEORY:

Hardware Requirement:
Raspberry Pi 3, USB to VGA Converter, Monitor, Keyboard, Mouse, 9v adaptor, USB cable, IR sensor, PIR sensor, Ultrasonic sensor, Temperature and humidity (DHT11) sensor, Temperature (LM35) (analog) sensor, DC motor, Stepper Motor, Breadboard.

Software Requirement:
Raspbian OS, Python 3 IDE.

THEORY:
IR (Infrared) Sensor IR (Infrared) Sensor works by emitting infrared signal/radiation and receiving of the signal when the signal bounces back from any obstacle. In other words, the IR Sensor works by continuously sending signal (in a direction) and continuously receive signal, if comes back by bouncing on any obstacle in the way. Components: IR Sensor Emitter: This component continuously

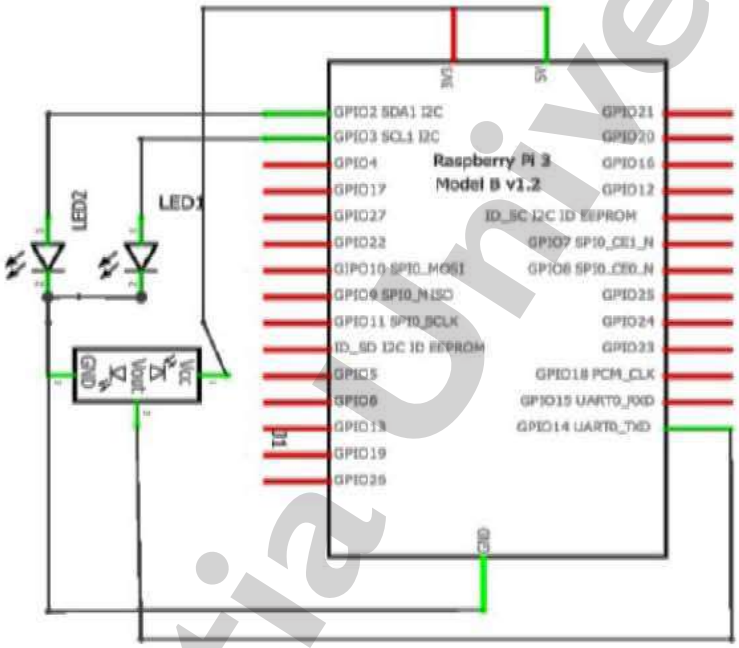
emits the infrared signal Receiver: It waits for the signal which is bounced back by obstacle Indicator: On board LED to signal if obstacle is deducted by the sensor Output: Could be used as Input for further processing of the signal Ground: Ground/Negative point of the circuit Voltage: Input 3.3V In this tutorial we will learn how we can Interface an IR sensor with Raspberry pi. These sensors are most commonly use in small robots like line follower robot, Edge avoiding robot etc.. Simply putting, it can detect the presence of objects before it and also differentiate between white and black colour.

So lets learn how to interface this sensor with Raspberry Pi. In this project, when there is no object in front of IR sensor then the Red LED remains turned on and soon as we put something in front of IR sensor then red LED turns off and Green LED turn on. This circuit can also serve as Security Alarm Circuit. IR Sensor Module: IR sensors (Infrared sensor) are modules which detect the presence of objects before them. If the object is present it give 3.3V as output and if it is not present it gives 0 volt. This is made possible by using a pair of IR pair (transmitter and receiver), the transmitter (IR LED) will emit an IR ray which will get reflected if there is a object present before it. This IR ray will be received back by the receiver (Photodiode) and the output will be made high after amplified using an op-amp link LM358 The IR Sensor used in this assignment is like all IR sensor it has three pins which are 5V, Gnd and Out respectively. The module is powered by the 5V pin from Raspberry Pi and the out pin is connected to GPIO14 of Raspberry Pi. The potentiometer on top of the module can be used to adjust the range of the IR sensor. Circuit Diagram and Explanation The circuit diagram for connecting Raspberry Pi with IR sensor is shown below. As you can see the circuit diagram is very simple. We have directly powered the IR module from the 5V and Ground Pin of Raspberry Pi. The output pin of the IR module is connected to the GPIO14. We have also used two LED (Green and Red) to indicate the status of the object. These two LEDs are connected to GPIO3 and GPIO2 respectively. Since the GPIO pins of Raspberry Pi are 3.3V, a current limiting resistor is not mandatory. However if desired a resistor of value 470 ohms can be added between the ground pin of LEDs and Raspberry Pi. The whole circuit is powered by a 5V mobile charger through the micro USB port of the Raspberry pi.

Note: When connecting any sensor, make sure the ground of the sensor is connected to ground of the MCU or MPU (here Raspberry Pi). Only then they will be able to communicate.

Code: Here we are using Python Programming language for programming RPi. There are many ways to program your Raspberry Pi. The complete Python program is given at the end of this tutorial. Working: Once you have created your

python code, execute it using the run command and see the result. You should also see the red colour LED going high when there is no object in front of the sensor. Now, bring something close to the IR led and you should notice the red LED turning off and the Green turning on. Applications: As mentioned in the earlier sections, Proximity Sensor or Obstacle Detection is the main application of interfacing IR Sensor with Raspberry Pi.



b) Interfacing PIR sensor with Raspberry pi:

All living beings radiate energy to the surroundings in the form of infrared radiations which are invisible to human eyes. A PIR (Passive infrared) sensor can be used to detect these passive radiations. When an object (human or animal) emitting infrared radiations passes through the field of view of the sensor, it detects the change in temperature and therefore can be used to detect motion.
HC-SR501 uses differential detection with two pyroelectric infrared sensors. By taking a difference of the values, the average temperature from the field of view of a sensor is removed and thereby reducing false positives.
Interfacing HC-SR501 with Raspberry Pi is easy because the output of a sensor is Pi friendly ie. 3.3V and it can be powered from the 5V rail of Pi.

The PIR sensor consists of 3 pins:
1. Vcc – 4.5V to 20V, Input power

2. OUTPUT – TTL output of sensor 0V, 3.3V
3. GND – Ground

The module has a rectangular window with two sub-probes 1 and 2 located at two ends of the rectangle. When a body emitting infrared radiation moves from side to side, the time for each probe for detection varies. Larger the time difference, more sensitive the device. It also uses a Fresnel lens to improve sensing aperture and filter in infrared waves.

Adjustment:
For adjusting the detection delay (0.3 seconds to 600 seconds): Turn the potentiometer clockwise to increase and anticlockwise to decrease.
For adjusting the sensing distance (3 meters to 7 meters): Turn the potentiometer clockwise to increase and anticlockwise to decrease.

Hardware Requirement

1. Raspberry Pi setup (screen,mouse,keypad,raspberry pi,AC to 5V DC converter).
2. 3 female to female connector wires
3. A PIR sensor.

c) Interfacing of Ultrasonic sensor with Raspberry Pi 3:

Ultrasonic Sensors, particularly HC-SR04 Ultrasonic Sensor, are very popular among electronic hobbyists and are frequently used in a variety of projects like Obstacle Avoiding Robot, Distance Measurement, Proximity Detection and so forth. In this project, we will learn about HC-SR04 Ultrasonic and see how to interface one with Raspberry Pi.

The range of this Ultrasonic Sensor seems very less but it is sufficient for the applications it is implemented in i.e. Proximity Detection and Obstacle Avoiding, for example. The Raspberry Pi Ultrasonic Sensor Interface is different from interfacing LED, Button, LCD, Motors, etc. with Raspberry Pi. This is because the output of the HC-SR04 Ultrasonic Sensor is at a 5V logic level whereas the Raspberry Pi works on a 3.3V logic level.

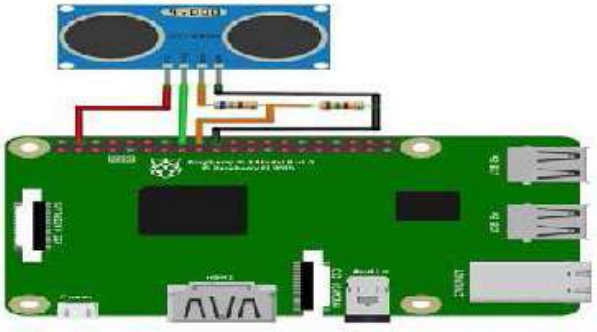
It basically consists of three parts: an ultrasonic transmitter, a control circuit and an ultrasonic receiver. Coming to the pins of the HC-SR04 Sensor, it has only four pins namely VCC, TRIG (Trigger), ECHO (Echo) and GND.

The following image shows the connections between the Raspberry Pi and the HC-SR04 Ultrasonic Sensor. This circuit diagram is made with Fritzing Software.

Components Required

- Raspberry Pi 3 Model B
- HC-SR04 Ultrasonic Sensor
- 680 Ω Resistor (1/4 Watt)
- 1.5 KΩ Resistor (1/4 Watt)
- Connecting Wires
- Mini Breadboard
- Power Supply
- Computer

Circuit Diagram:



d) Interfacing of DHT11 with Raspberry Pi 3 :

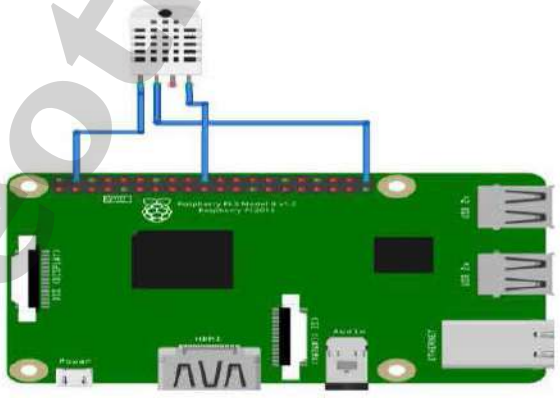
Hardware used:

- Raspberry-Pi Board
- DHT11 Temperature and Humidity Sensor
- Connecting Wires
- Software used:
- Python Interpreter

THEORY:

In this project, we will learn about DHT11 Humidity and Temperature Sensor and how the Raspberry Pi DHT11 Humidity Sensor interface works Overview DHT11 is a Digital Sensor consisting of two different sensors in a single package. The sensor contains an NTC (Negative Temperature Coefficient) Temperature Sensor, a Resistive-type Humidity Sensor and an 8-bit Microcontroller to convert the analog signals from these sensors and produce a Digital Output. Circuit Diagram The following is the circuit diagram of the DHT11 and Raspberry Pi Interface.

Circuit Design:



If you observe the circuit diagram, there is not a lot of stuff going on with respect to the connections. All you need to do is to connect the VCC and GND pins of the DHT11 Sensor to +5V and GND of Raspberry Pi and then connect the Data OUT of the Sensor to the GPIO4 i.e. Physical Pin 7 of the Raspberry Pi.

Installing DTH11 Library:

Since we are using a library called Adafruit_DHT provided by Adafruit for this project, we need to first install this library into Raspberry Pi.

- First step is to download the library from GitHub
- Now, enter the following command to download the files related to the Adafruit_DHT library.

git clone http://github.com/adafruit/Adafruit_Python_DHT.git

- All the contents will be downloaded to a folder called 'Adafruit_Python_DHT'. Open this directory using cd Adafruit_Python_DHT. To see the contents of this folder, use 'ls' command.
- In that folder, there is file called 'setup.py'. We need to install this file using the following command.

sudo python setup.py install

e) Interfacing of DC motors with Raspberry Pi 3:

Principle of Operation:

The main principle in controlling a DC Motor with Raspberry Pi lies with the Motor Driver. A Motor Driver is a special circuit or IC that provides the necessary power (or rather the current) to the motor for smooth and safe operation.Even a small 5V DC Motor draws a high initial current of around 300 – 400 mA. This current will then fall down 150 – 200 mA as the motor gains speed to around. This is a huge current for devices like Microcontrollers, Arduino, Raspberry Pi etc. Hence, we should never connect a motor directly to Raspberry Pi (or any other microcontroller). Motor Driver play an important role in this situation. They take the control signals from Raspberry Pi and provide the necessary drive current to the motor from the power supply.

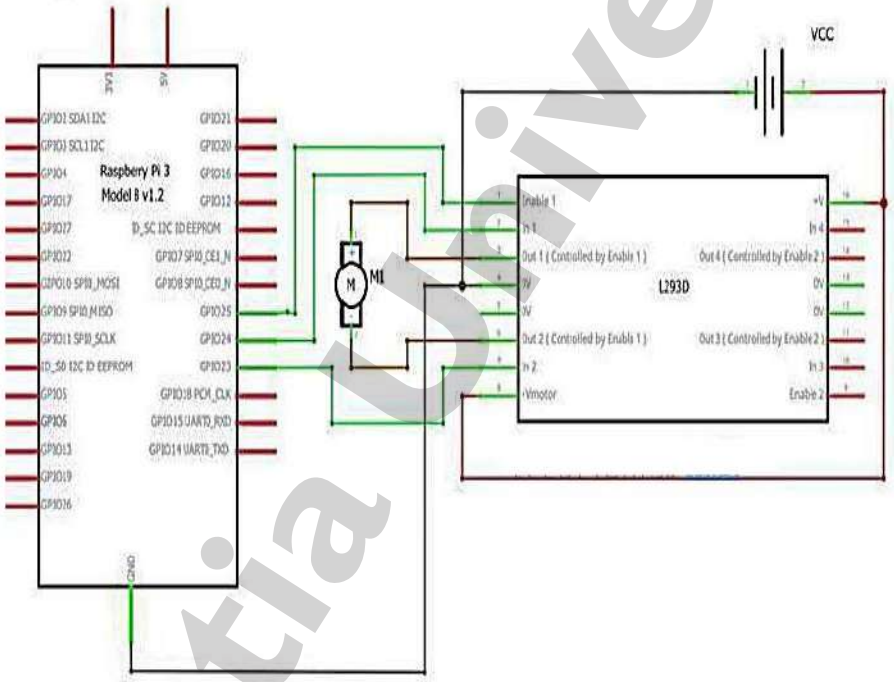
In this project, the motor driver (L293D) is given with two control signals from Raspberry Pi through GPIO Pins. As per the Python Program, the motor will rotate in either forward or reverse direction.

Components Required

- Raspberry Pi 3 Model B
- L293D Motor Driver IC or Module

- Small DC Motor (5V)
- Connecting wires (Jumper Wires)
- 5V – 2A Power Supply for Raspberry Pi
- 5V Supply for Motor
- Miscellaneous (Computer, Ethernet Cable, etc.)

Circuit Diagram:



- The design of the circuit for controlling a DC Motor with Raspberry Pi is very simple. First, connect the pins 8 and 16 (VCC2 and VCC1) of L293D to external 5V supply (assuming you are using a 5V Motor).
- There are four ground pins on L293D. Connect pin 4 to the GND of supply. Also, connect the ground pin of L293D to GND pin of the Raspberry Pi.
- Finally, we have the enable and control input pins. Connect the pin 1 of L293D (1,2EN) to GPIO25 (Physical Pin 22) of Raspberry Pi. Then connect control input pins 2 and 7 (1A and 2A) to GPIO24 (Physical Pin 18) and GPIO23 (Physical Pin 16) respectively.

CONCLUSION:

Learn to interface various sensors with Raspberry Pi 3, learn to interface various actuators with Raspberry Pi 3, learn to Install Operating System Raspberry Pi 3 and learn to program Raspberry Pi 3.

THE NEOTIA UNIVERSITY
DEPARTMENT OF ROBOTICS & AUTOMATION

EXPERIMENT NO.: 5

NAME OF THE EXPERIMENT: WAP for LED blink using Raspberry Pi.

OBJECTIVE: Student should get the knowledge of LED blinking using Raspberry Pi.

HARDWARE REQUIREMENTS:

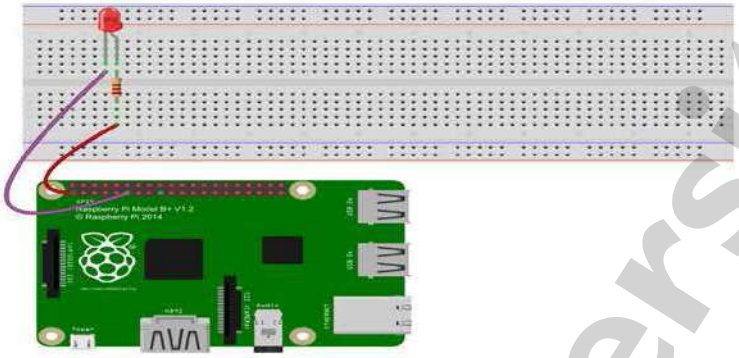
- 1x Breadboard
- 1x Raspberry Pi
- 1x RGB LED
- 1x 330Ω Resistor
- 2x Jumper Wires

THEORY:

Semiconductor light-emitting diode is a type of component which can turn electric energy into light energy via PN junctions. By wavelength, it can be categorized into laser diode, infrared light-emitting diode and visible light-emitting diode which is usually known as light-emitting diode (LED).

When 2V-3V forward voltage is supplied to an LED, it will blink only if forward currents flow through the LED. Usually there are red, yellow, green, blue and color-changing LEDs which change color with different voltages. LEDs are widely used due to their low operating voltage, low current, luminescent stability and small size. LEDs are diodes too. Hence they have a voltage drop which usually varies from 1V to 3V depending on their types. Generally they brighten if supplied with a 5mA-30mA current and we usually use 10mA-20mA.Thus when an LED is used ,it is necessary to connect a current-limiting resistor to protect it from being burnt.

In this experiment, connect a 220Ω resistor to the anode of the LED, then the resistor to 3.3 V and connect the cathode of the LED to GPIO0 (See Raspberry Pi Pin Number Introduction). Write 1 to GPIO0, and the LED will stay off; write 0 to GPIO0, and then the LED will blink, just as indicated by the principle above.



Step 1: Build the circuit given above
Step 2: Change directory
cd /home/pi/Sunfounder_SuperKit_ Python_code_for_RaspberryPi/

Step 3: Run
sudo python 01_led.py
Now, you should see the LED blink.

Python Code:

```
#!/usr/bin/env python
import RPi.GPIO as GPIO
import time

LedPin = 11    # pin11

def setup():
        GPIO.setmode(GPIO.BOARD)       # Numbers GPIOs by physical location
        GPIO.setup(LedPin, GPIO.OUT)   # Set LedPin's mode is output
        GPIO.output(LedPin, GPIO.HIGH) # Set LedPin high(+3.3V) to off led

def loop():
        while True:
                print '...led on'
                GPIO.output(LedPin, GPIO.LOW)  # led on
                time.sleep(0.5)
                print 'led off...'
                GPIO.output(LedPin, GPIO.HIGH) # led off
                time.sleep(0.5)

def destroy():
        GPIO.output(LedPin, GPIO.HIGH)     # led off
        GPIO.cleanup()                     # Release resource
```

```
if   name   == '  main  ':     # Program start from here
        setup()
        try:
                loop()
        except KeyboardInterrupt:  # When 'Ctrl+C' is pressed, the child program destroy() will be
executed.
                destroy()
```

CONCLUSION: Student will be developed program of LED bilking using Raspberry Pi.

THE NEOTIA UNIVERSITY
DEPARTMENT OF ROBOTICS & AUTOMATION

EXPERIMENT NO.: 6

NAME OF THE EXPERIMENT: Study and Implement Zigbee Protocol using Arduino / Raspberry Pi.

OBJECTIVE: Student should get the knowledge of Zigbee Protocol using Raspberry Pi.

HARDWARE REQUIREMENTS:

- Raspberry Pi2
- XBee 1mW Wire Antenna- Series 1 (2 No:)
- XBee Explorer Dongle (2 No :)

THEORY:

ZigBee Communication Using Raspberry Pi:

ZigBee is a communication device used for the data transfer between the controllers, computers, systems, really anything with a serial port. As it works with low power consumption, the transmission distances is limited to 10–100 meters line-of-sight. ZigBee devices can transmit data over long distances by passing data through a mesh network of intermediate devices to reach more distant ones. ZigBee is typically used in low data rate applications that require long battery life and secure networking. Its main applications are in the field of wireless sensor network based on industries as it requires short-range low-rate wireless data transfer. The technology defined by the ZigBee specification is intended to be simpler and less expensive than other wireless networks.

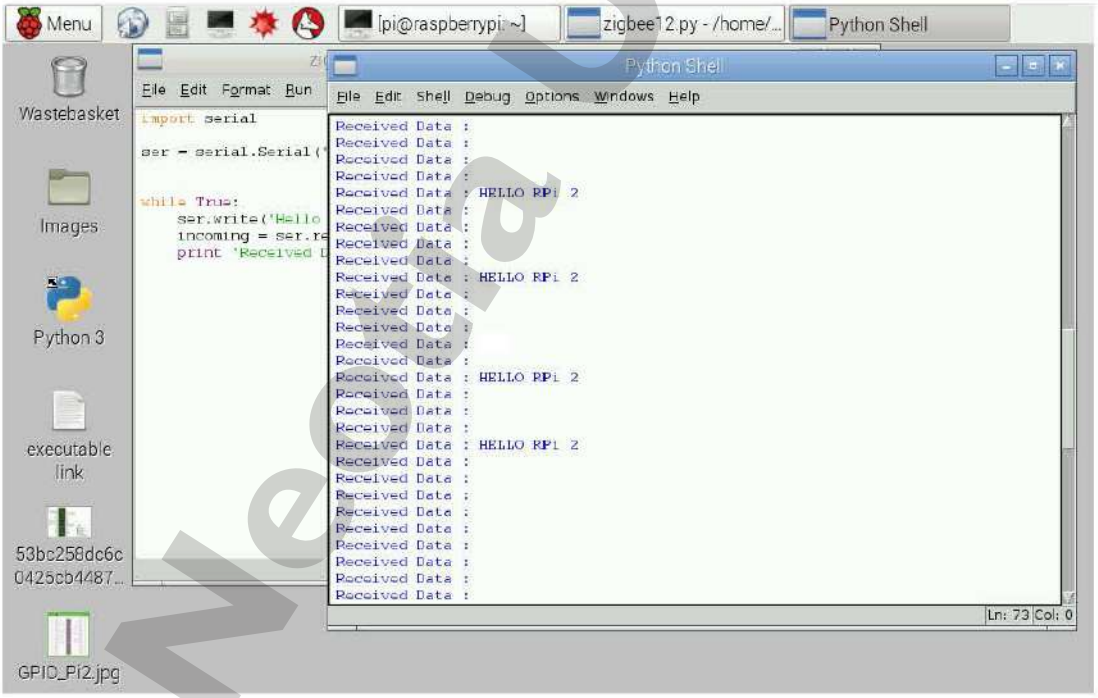
Here we make use of an interface of Zigbee with Raspberry Pi2 for a proper wireless communication. Raspberry Pi2 has got four USB ports, so it is better to use a Zigbee Dongle for this interface. Now we want to check the communication between the two paired ZigBee modules.

The response showed inside a red box indicates the presence of a usb device in the module. Write a python script to perform Zigbee communication which is given below.

```
import serial

# Enable USB Communication
ser = serial.Serial('/dev/ttyUSB0', 9600,timeout=.5)
while True:
ser.write('Hello User \r\n') # write a Data
incoming = ser.readline().strip()
print 'Received Data : '+ incoming
```

The two zigbee must be in a line of sight and check the results in the Python shell and in the hyperterminal of the computer.



CONCLUSION: Student will be developed program of Zigbee Protocol using Raspberry Pi.

Courtesy: Google source