ABSTRACT

 While the cost of living is going up, there is a growing focus to involve technology to lower those prices. With this in mind the Smart Home project allows the user to build and maintain a house that is smart enough to keep energy levels down while providing more automated applications.

A smart home will take advantage of its environment and allow seamless control whether the user is present or away. With a home that has this advantage, you can know that your home is performing at its best in energy performance.

It is to be implemented in existing home environments, without any changes in the infrastructure. Home Automation let the user to control the home from his or her computer or mobile and assign actions that should happen depending on time or other sensor readings such as light, temperature humidity from any device in the Home Automation system.

INTRODUCTION

Internet of Things (IoT) conceptualizes the idea of remotely connecting and monitoring real world objects (things) through the Internet. When it comes to our house, this concept can be aptly incorporated to make it smarter, safer and automated. This IoT project focuses on building a smart wireless home security system which sends alerts to the owner by using Internet who enters and at what time optionally. Besides, the same can also be utilized for home automation by making use of the same set of sensors.

The microcontroller used in the current prototype is the Arduino UNO which comes with an embedded micro-controller making use of which all the electrical appliances inside the home can be controlled and managed.

It involves the control and automation of lighting, heating, cooling, security etc. Wi-Fi is often used for remote monitoring and control. Home devices, when remotely monitored and controlled via the Internet, are an important constituent of the ***Internet of Things***.

**WHY ARDUINO?**

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. [Arduino boards](http://www.arduino.cc/en/Main/Products) are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux.

Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

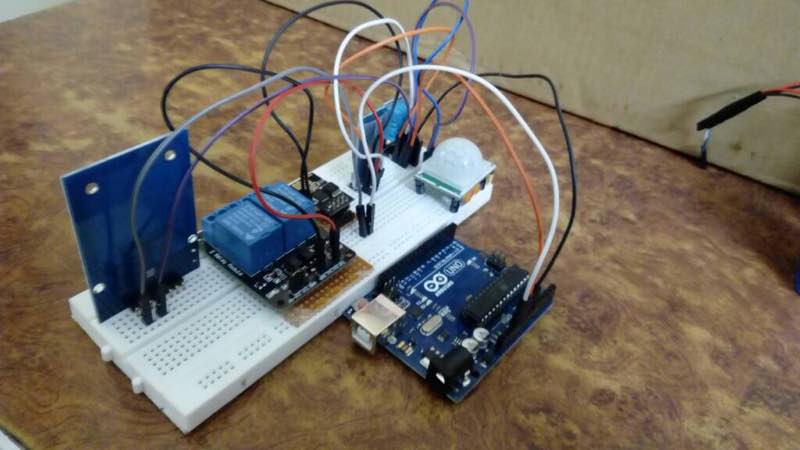
* **Inexpensive -** Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost cheap.
* **Cross-platform -** The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
* **Simple, clear programming environment**- The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
* **Open source and extensible software -** The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
* **Open source and extensible hardware -** The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the [breadboard version of the module](http://www.arduino.cc/en/Main/Standalone) in order to understand how it works and save money.

RELATED WORKS

Our project on Smart Home System has some similar operations which can be implemented in many other fields. In Smart Hotel System the RFID can be used as unique room number which can be used to enter the room and the PIR and DHT11 sensor is used to control the temperature of the room. Also an override option can be given to the guest living in a particular room using Bluetooth module. Similarly, this system can be used in seminar rooms of the college allocated only for the teachers for any meeting and also can be used in any IT sector office for any team allocated to a particular project. They can use the room for project meeting purposes and also for group discussions.

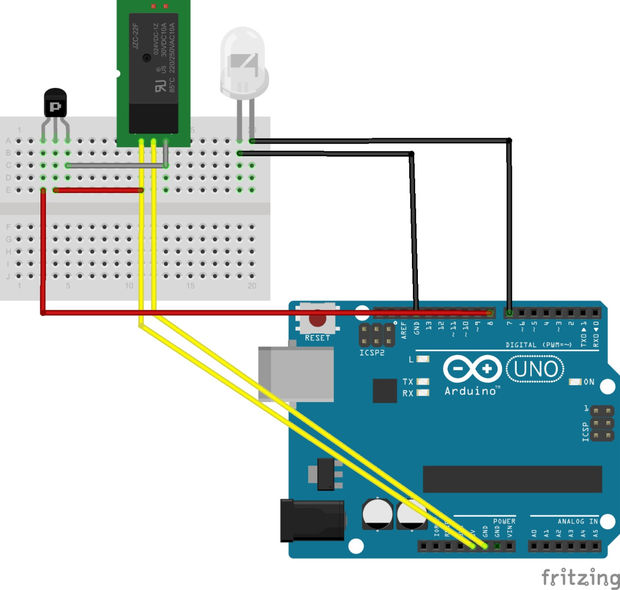
PROCEDURE

* **CIRCUIT DIAGRAM:**



Basic connection of Arduino UNO R3 with RFID module, PIR sensor, Wi-Fi module, Bluetooth module, humidity and temperature sensor.

* ***CONNECTION OF ARDUINO WITH RFID MODULE:***

[](http://www.instructables.com/file/F66R2DEIGHOHNAP/)

**Connection Description:**

**MOSI** ---> PIN 11 **MISO** ---> PIN 12

**SCK** ---> PIN 13 **SS/SDA** ---> PIN 10

**RST** ---> PIN 9 **LED** ---> PIN 7 & GND

**Output pow**er ---> VCC connects to 5Volt pins and collector pin of TIP31.

**GND** connect to GND pins; **IN** connect to **emitter** pin of TIP31

**Pin 8** is connected to the **base** pin of the TIP31

**Code:**

#include "SPI.h"

#include "MFRC522.h"

#define SS\_PIN 10

#define RST\_PIN 9

#define SP\_PIN 8

MFRC522 rfid(SS\_PIN, RST\_PIN);

MFRC522::MIFARE\_Key key;

void setup() {

Serial.begin(9600);

SPI.begin();

rfid.PCD\_Init();

pinMode(A1,OUTPUT);

pinMode(A2,OUTPUT);

pinMode(A3,OUTPUT);

pinMode(A4,OUTPUT);

}

void loop() {

if (!rfid.PICC\_IsNewCardPresent() || !rfid.PICC\_ReadCardSerial())

return;

// Serial.print(F("PICC type: "));

MFRC522::PICC\_TypepiccType = rfid.PICC\_GetType(rfid.uid.sak);

// Serial.println(rfid.PICC\_GetTypeName(piccType));

// Check is the PICC of Classic MIFARE type

if (piccType != MFRC522::PICC\_TYPE\_MIFARE\_MINI &&

piccType != MFRC522::PICC\_TYPE\_MIFARE\_1K &&

piccType != MFRC522::PICC\_TYPE\_MIFARE\_4K) {

Serial.println(F("Your tag is not of type MIFARE Classic."));

return;

}

String strID = "";

for (byte i = 0; i< 4; i++) {

strID +=

(rfid.uid.uidByte[i] < 0x10 ? "0" : "") +

String(rfid.uid.uidByte[i], HEX) +

(i!=3 ? ":" : "");

}

strID.toUpperCase();

//Serial.print("Tap card key: ");

//Serial.println(strID);

if(strID=="EB:FE:EF:66")

{

Serial.println("Urmisha Das");

digitalWrite(A4,HIGH);

delay(1000);

digitalWrite(A4,LOW);

rfid.PICC\_HaltA();

rfid.PCD\_StopCrypto1();

return;

}

if(strID=="66:69:04:07")

{

Serial.println("Meghna Kundu");

digitalWrite(A1,HIGH);

delay(1000);

digitalWrite(A1,LOW);

rfid.PICC\_HaltA();

rfid.PCD\_StopCrypto1();

return;

}

if(strID=="DB:B4:EC:66")

{

Serial.println("Arpita Mandal");

digitalWrite(A2,HIGH);

delay(1000);

digitalWrite(A2,LOW);

rfid.PICC\_HaltA();

rfid.PCD\_StopCrypto1();

return;

}

if(strID=="DB:43:02:99")

{

Serial.println("Aditya Biwas");

digitalWrite(A3,HIGH);

delay(1000);

digitalWrite(A3,LOW);

rfid.PICC\_HaltA();

rfid.PCD\_StopCrypto1();

return;

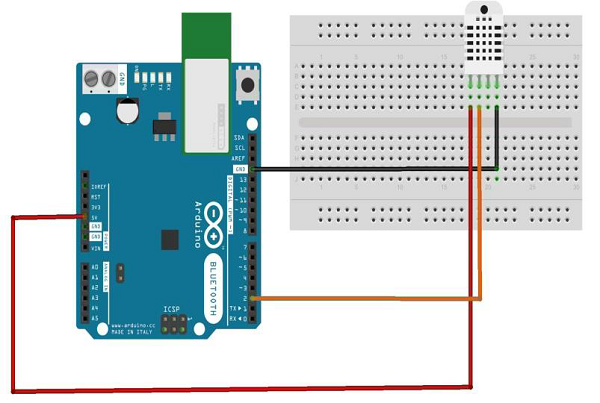
}

rfid.PICC\_HaltA();

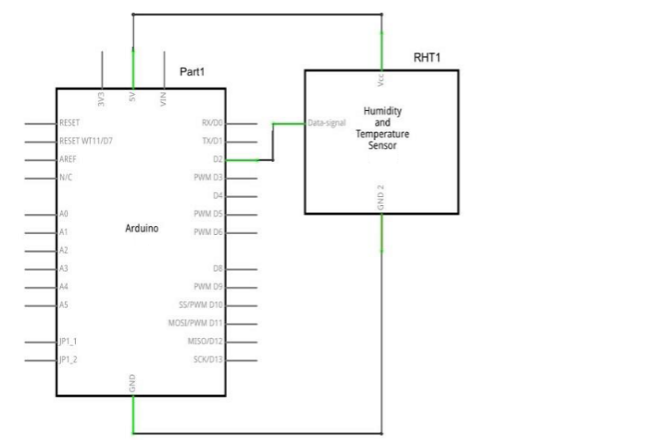
rfid.PCD\_StopCrypto1();

}

* **ARDUINO CONNECTION WITH TEMPERATURE AND HUMIDITY SENSOR**

****

* + - **Power** − 3-5V
    - **Max Current** − 2.5mA
    - **Humidity** − 0-100%, 2-5% accuracy
    - **Temperature** − 40 to 80°C, ±0.5°C accuracy



**Connection Description:**

* + - DATA pin to Arduino pin number 2.
    - Vcc pin to 5 volt of Arduino board.
    - GND pin to the ground of Arduino board.
    - We need to connect 10k ohm resistor (pull up resistor) between the DATA and the Vcc pin.

**Code:**

#include<dht.h>

dht DHT;

**// if you require to change the pin number, Edit the pin with your arduino pin.**

#define DHT11\_PIN A0

void setup() {

Serial.begin(9600);

Serial.println("welcome to TechPonder Humidity and temperature Detector"); }

void loop() { // READ DATA

intchk = DHT.read11(DHT11\_PIN);

Serial.println(" Humidity " );

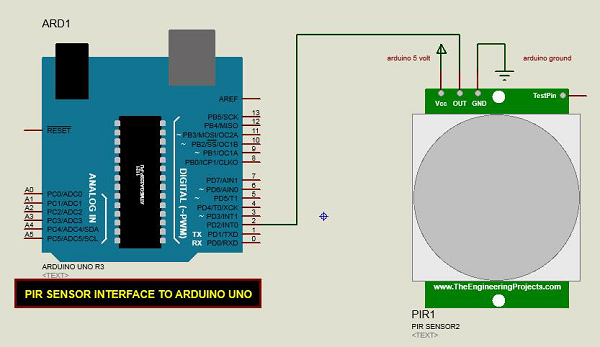
Serial.println(DHT.humidity, 1);

Serial.println(" Temparature ");

Serial.println(DHT.temperature, 1);

delay(2000);

* **ARDUINO CONNECTION WITH PIR SENSOR:**



**Connection Description:**

* + - Connect the +Vcc to +5v on Arduino board.
    - Connect OUT to digital pin 2 on Arduino board.
    - Connect GND with GND on Arduino.

**Code:**

int input=A0;

void setup() {

Serial.begin(9600);

// put your setup code here, to run once:

pinMode(input,OUTPUT);

}

void loop() {

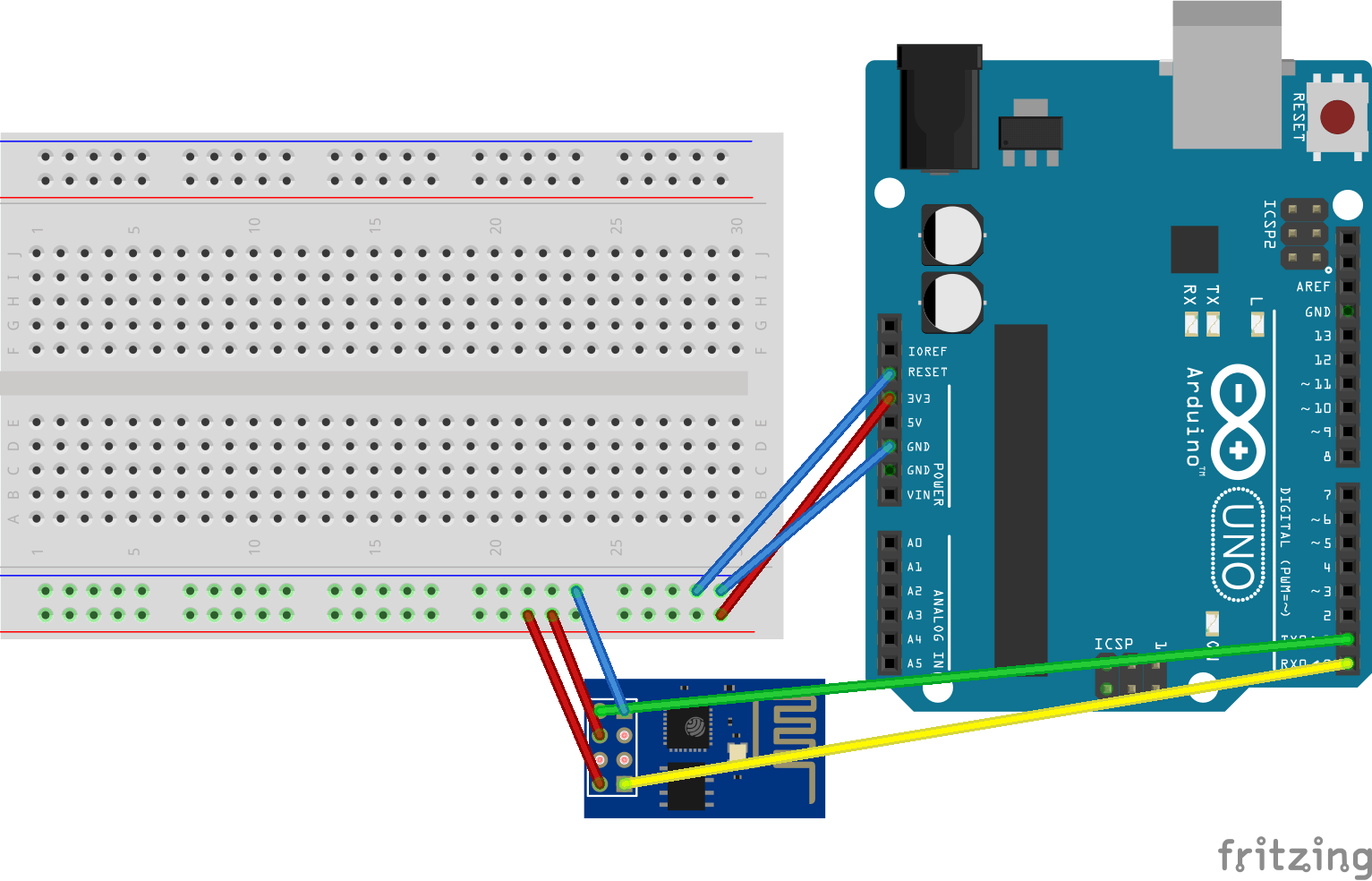
// put your main code here, to run repeatedly:

Serial.println(analogRead(A0));

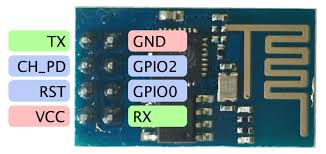
delay(1000);

}

* **CONNECTION OF ESP8266 WITH ARDUINO**

****

**DETAILED DIAGRAM OF ESP 8266:**

****

**Connection Description**

Connect the Arduino’s **3v3** (3.3V) output to the red line on a breadboard. The ESP8266 works with 3.3V and not 5V, so this is necessary. If you want to connect other components that use 5V, you can connect the 5V output to the other red line of the breadboard, just make sure you don’t connect the two.

1. Connect **GND** (ground) to the blue line.
2. Connect the **RES** or **RESET** pin to the blue line. When you ground the reset pin, the Arduino works as a dumb USB to serial connector, which is what we want to talk to the ESP8266.
3. Connect the **RXD** pin of the Arduino to the **RX** pin of the ESP8266 (yellow color in the picture).
4. Connect the **TXD** pin of the Arduino to the **TX** pin of the ESP (green color in the picture). Usually, when we want two things to talk to each other over serial, we connect the **TX** pin of one to the **RX** of the other (send goes to receive and the opposite). Here we do not have the Arduino talk to the ESP8266 though, our computer is talking to it *via* the Arduino.
5. Connect the **GND** pin of the ESP to the blue line and the **VCC** pin to the red line.
6. Finally **CH\_PD** goes to the red line, supposedly it will not work if you do not connect this.

**Code:**

#include <SoftwareSerial.h>

SoftwareSerial esp8266(4,5); // make RX Arduino line is pin 2, make TX Arduino line is pin 3.

// This means that you need to connect the TX line from the esp to the Arduino's pin 2

// and the RX line from the esp to the Arduino's pin 3

void setup()

{

Serial.begin(9600);

esp8266.begin(115200); // your esp's baud rate might be different

}

void loop()

{

if(esp8266.available()) // check if the esp is sending a message

{

while(esp8266.available())

{

// The esp has data so display its output to the serial window

char c = esp8266.read(); // read the next character.

Serial.write(c);

}

}

if(Serial.available())

{

// the following delay is required because otherwise the arduino will read the first letter of the command but not the rest

// In other words without the delay if you use AT+RST, for example, the Arduino will read the letter A send it, then read the rest and send it

// but we want to send everything at the same time.

delay(1000);

String command="Wifi connected";

while(Serial.available()) // read the command character by character

{

// read one character

command+=(char)Serial.read();

}

esp8266.println(command); // send the read character to the esp8266

}

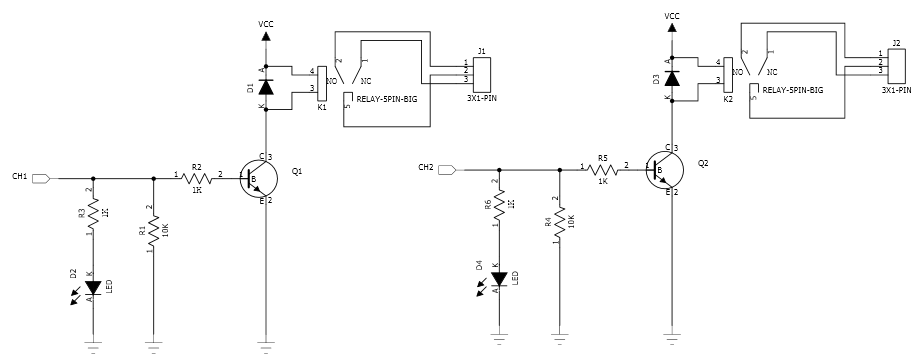
}

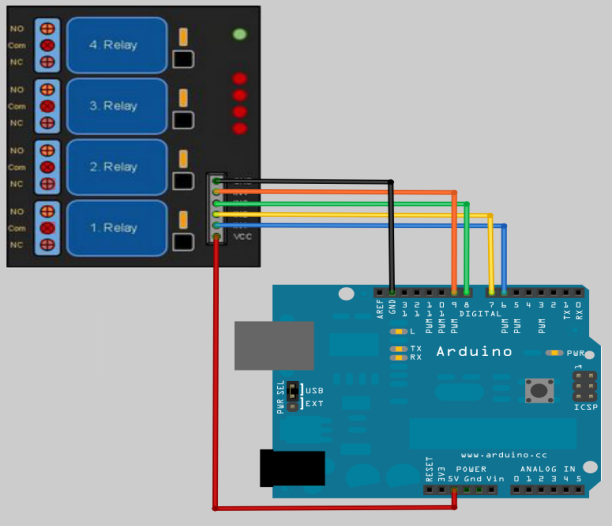
* **CONNECTION OF ARDUINO WITH RELAY:**

Relays work on electromagnetism, When the Relay coil is energized it acts like a magnet and changes the position of a switch. The circuit which powers the coil is completely isolated from the part which switches ON/OFF

The default state of the relay when the power is off for COMM (power) to be connected to NC (normally closed), this is the equivalent of setting the 4 Relay boards IN pin to HIGH (has +5v) sent to it. It is a safety feature to notuse the NC connector in-case you Arduino loses power it will automatically turns off all the devices connected to the relay.  When you have something connected to the relays NO (Normally Open) connector and you set the corresponding IN pin to LOW (0v), power will flow in from the COMM connector and out of the NO connector powering your device.

**Circuit Diagram**





**Code:**

int relay=7;

int relay2=8;

intpir=6;

void setup() {

// put your setup code here, to run once:

pinMode(relay,OUTPUT);

pinMode(relay2,OUTPUT);

pinMode(pir,INPUT);

Serial.begin(9600);

}

void loop() {

// put your main code here, to run repeatedly:

if(digitalRead(pir)==HIGH){

digitalWrite(relay,LOW);

digitalWrite(relay2,HIGH);

Serial.println("pir");

delay(40);}

else{

digitalWrite(relay,HIGH);

digitalWrite(relay2,LOW);

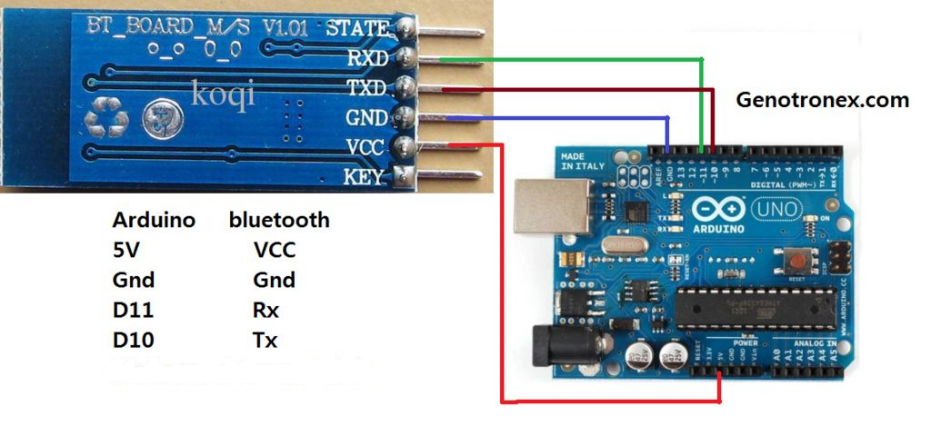
Serial.println("no pir");

delay(40);

}

}

* **ARDUINO CONNECTION WITH BLUETOOTH MODULE:**



For the Arduino or other micro controllers that can't use a USB Bluetooth Dongle you need a module or shield. With the Arduino we would connect such a module to either the hardware serial port or if we use the SoftwareSerial.h Arduino Library then you can choose which pins to connect the module to. We show a simple Arduino sketch below where the Bluetooth module pins are connected to pins 10 & 11 on the Arduino. These always cross over so the TX pin from the module will connect to RX (here it is pin 10) on the Arduino. Also there is a software library called SoftwareSerial.h which enables you to use any pins on the Arduino to connect up our Bluetooth module and this is important if we want to use both serial ports at the same time, the **soft Serial** port to send and receive commands over the Bluetooth link to BTInterface and the hardware **Serial** port to send and receive commands to the Serial Monitor program. So we need to setup your Bluetooth module with the Transmit pin on the module going to the Receive pin (RX) on the Arduino (pin 10) and the Receive pin on the module going to the Transmit pin (TX) on the Arduino (pin 11), then when you connect to your Bluetooth module with BTInterface you can send commands directly from the serial monitor and see which commands are being sent from BTInterface. We should be able to see from this that the **Serial** port is connected to the serial monitor and the **soft Serial** port is connected to the Bluetooth module. When we have this set up correctly then anything you type in the Send line text box of the serial monitor will be sent to BTInterface when you hit the return key (or press the send button) and it will also be displayed in the main part and anything that BTInterface sends will also be displayed.WhenBTInterface first connects to a Bluetooth device it sends the word **btinterface** which your program can use to know that it has connected so here is a simple example of how you would write code to detect this word and do stuff.

**Code:**

#include <SoftwareSerial.h>

SoftwareSerialbluetooth(2,3); // RX, TX

char bt;

void setup() {

// put your setup code here, to run once:

bluetooth.begin(9600);

Serial.begin(9600);

pinMode(7,OUTPUT);

pinMode(8,OUTPUT);

}

void loop() {

// put your main code here, to run repeatedly:

digitalWrite(8,HIGH);

digitalWrite(7,HIGH);

while (bluetooth.available()>0)

{

bt=bluetooth.read();

Serial.println(bt);

if(bt=='a')

{

ll:

//Serial.println("hello");

digitalWrite(7,LOW);

digitalWrite(8,HIGH);

delay(100);

while(1)

if(bluetooth.read()=='b')

gotojk;

}

if(bt=='b')

{

jk:

//Serial.println("hi");

digitalWrite(8,LOW);

digitalWrite(7,HIGH);

delay(100);

while(1)

if(bluetooth.read()=='a')

gotoll;

}

}

**Result and Analysis**

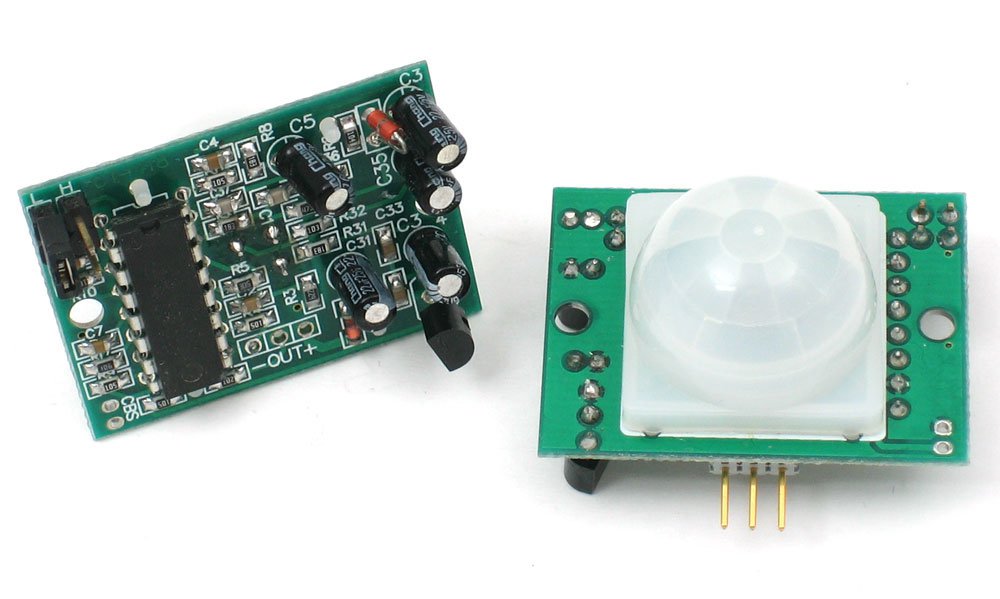
**Our smart home comprises three main features:**

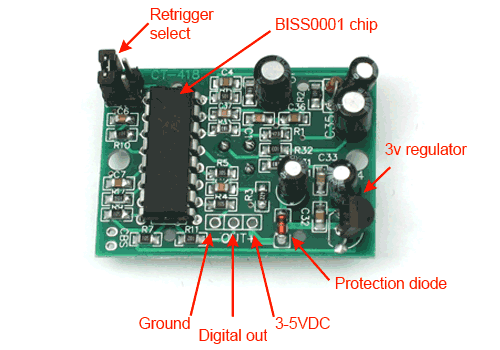
1. **Smart Key:** Smart key in our project is implemented with RFID module (RC 522). This uses electromagnetic fields to automatically identify and track tags attached to objects.

***Fig: RC522 (RFID MODULE)***

The RC522 use of advanced modulation and demodulation concept completely integrated in the 13.56MHz all kinds of passive contactless communication methods and protocols.14443A compatible transponder signal. The digital part handles the ISO14443A frames and error detection. In addition, support Quick CRYPTO1 encryption algorithm, the term verification MIFARE series. MFRC522 support MIFARE series of high-speed non-contact communication, two-way data transfer rates up to 424kbit/s. With the use of RFID module only the registered card user can unlock the door of the house, and the person entering the home his or her name and time of entrance it will be uploaded in cloud and the owner will receive an email.

1. **Automated electric appliances:** PIR sensors allows to sense motion, almost always used to detect whether a human has moved in or out of the sensors range. For that reason they are commonly found in appliances and gadgets used in homes or businesses. They are often referred to as PIR, "Passive Infrared", "Pyro electric", or "IR motion" sensors.

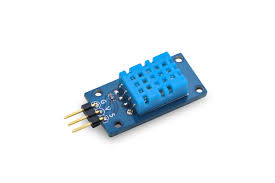


****

**Fig: *PIR (passive electrical sensor)***

PIR sensors are infrared-sensitive light detectors. By monitoring light in the infrared spectrum, PIR sensors can sense subtle changes in temperature across the area they’re viewing. When a human or some other object comes into the PIR’s field-of-view, the radiation pattern changes, and the PIR interprets that change as movement i.e., in our work it will switch of fan & light if it can sense any person within the room(specified range)

1. ***Humidity and temperature sensor:***The RHT03 (also known by DHT-22) is a low-cost humidity and temperature sensor with a single wire digital interface. The sensor is calibrated and doesn’t require extra components so you can get right to measuring relative humidity and temperature.



***FIG:DHT11 temperature and Humidity sensor***

The DHT11 is a basic, ultra low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and spits out a digital signal on the data pin (no analog input pins needed). It’s simple to use, but requires careful timing to grab data. The only real downside of this sensor is you can only get new data from it once every 2 seconds, so when using our library, sensor readings can be up to 2 seconds old. With this sensor, it will sense the surrounding temperature and humidity, according to user defined or customised threshold value it will automatically switch the cooler on.

CONCLUSION

We have done IOT (Internet of Things) project on Smart Home System using Smart Key(RFID), electricity control(PIR) and Temperature control(DHT11) using relay(2-pin). In necessary time given with the above resources.

There is also a scope of improvement in our project which we have discussed earlier.

Through this project, we tried to contribute to a better future of the smart city which is the ultimate target of our project.

FUTURE SCOPE

Predicting the future of just about anything is very risky business. Smart Home System is an industry that largely started with X10 devices in 1980. Today, we believe the future of home automation will very much ride the digital age and develop along with the computer and networking systems in the years to come.

Initially it appeared, companies such as Microsoft and Exceptional Innovation with their Life/ware software were positioning the Windows Media Centre PC as the heart and soul of a complete solution for Smart Home System relying on web services to seamlessly interface with Automated Kitchen Appliances, Automated Coffee Makers, Dog Feeders, Automated Garage Doors, Automated Water Pumping System, switches to compliment the digital media management capabilities of Windows Media Centre. However, Exceptional Innovation stopped selling their systems for residential installations, and Microsoft Media Centre capabilities have disappeared in the Windows 8 OS.

REFERENCES

* <http://playground.arduino.cc/Code/PIRsense>
* <http://www.eng.utah.edu/~cs3992/archive/2012/SmartHomeFinalReport.pdf>
* <http://www.instructables.com/id/Arduino-RFID-Reader-MFRC522-Turorial/>
* <http://btinterface.com/BTInterface/BTInterfaceHelp/stepbysteparduino.html>
* <https://www.elprocus.com/wireless-home-automation-using-internet-of-things/>