

Smart Home

Assignment-1

By
C. Gnanaruby
952319104013

Smart Home

IOT or internet of things is an upcoming technology that allows us to control hardware devices through the internet. Here we propose to use IOT in order to control home appliances, thus automating modern homes through the internet. This system uses three loads to demonstrate as house lighting and a fan. Our user friendly interface allows a user to easily control these home appliances through the internet. For this system we use an AVR family microcontroller. This microcontroller is interfaced with a wifi modem to get user commands over the internet. Also we have an LCD display to display system status. Relays are used to switch loads. The entire system is powered by a 12 V transformer. After receiving user commands over the internet, microcontroller processes these instructions to operate these

loads accordingly and display the system status on an LCD display.

Hardware Specifications

- . Atmega Microcontroller
- . ESP8266 Wifi Module
- . LCD Display
- . DC Cooling Fan
- . Relay
- . Relay Driver IC
- . Vtg Regulator IC
- . Resistors
- . Capacitors
- . Transistors
- . Cables and Connectors
- . Diodes
- . PCB and Breadboards
- . LED
- . Transformer/Adapter

- . Push Buttons
- . Switch
- . IC
- . IC Sockets

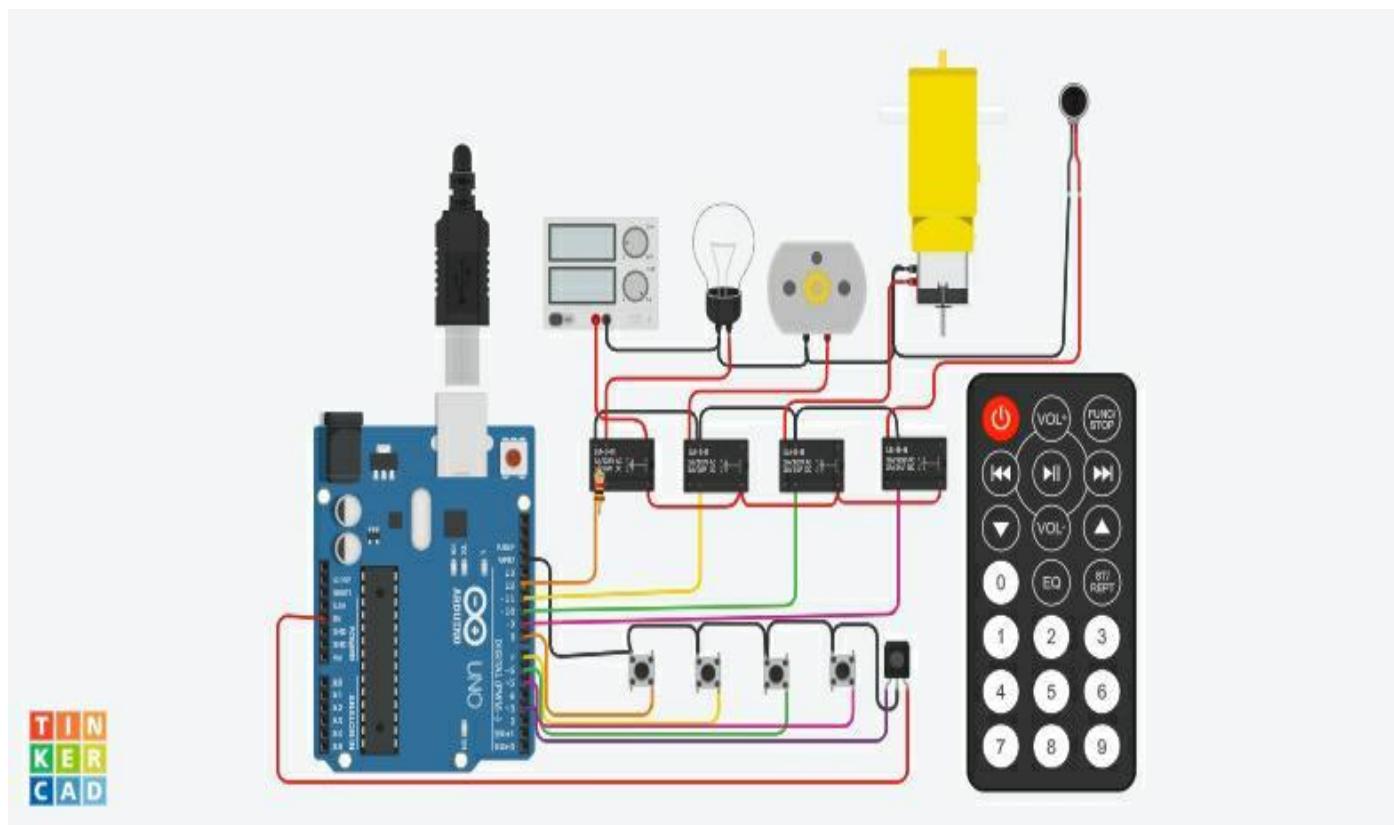
Software Specifications

- . Arduino Compiler
- . MC Programming Language: C
- . IOTGecko

IoT devices for securing your home

- Wall switches
- Voltage Sensors
- Air Conditioner
- Energy Motors
- Smart Door lock

Circuit Diagram For Smart Home



Coding

```
#include <SPI.h>
```

```
#include <Wire.h>
```

```
#include <IRremote.h>
```

```
const int relay_1 = 12; const  
int relay_2 = 11; const int  
relay_3 = 10; const int  
relay_4 = 9;
```

```
const int mswitch_1 = 8; const  
int mswitch_2 = 7; const int  
mswitch_3 = 6; const int  
mswitch_4 = 5; int  
RECV_PIN = 3;
```

```
IRrecv irrecv(RECV_PIN); decode_results  
results;
```

```
int toggleState_1 = 0; int  
toggleState_2 = 0; int  
toggleState_3 = 0; int  
toggleState_4 = 0;
```

```
void setup() {
```

```
    Serial.begin(9600);  
    irrecv.enableIRIn();
```

```
    pinMode(relay_1,          OUTPUT);  
    pinMode(relay_2, OUTPUT); pinMode(relay_3,  
    OUTPUT); pinMode(relay_4, OUTPUT);
```

```
pinMode(mswitch_1, INPUT_PULLUP);  
pinMode(mswitch_2, INPUT_PULLUP);  
pinMode(mswitch_3, INPUT_PULLUP);  
pinMode(mswitch_4, INPUT_PULLUP);  
}
```

```
void relayOnOff(int relay){
```

```
    switch(relay){
```

```
    case 1:
```

```
        if(toggleState_1 == 0){  
digitalWrite(relay_1, HIGH); // turn on relay  
1          toggleState_1 = 1;  
        }  
        else{          digitalWrite(relay_1,  
LOW); // turn off relay 1  
toggleState_1 = 0;
```



```

        }
delay(100);
break;    case 2:
            if(toggleState_2 == 0){
digitalWrite(relay_2, HIGH); // turn on relay
2          toggleState_2 = 1;
            }          else{
digitalWrite(relay_2, LOW); // turn off
relay 2      toggleState_2 = 0;
            }
delay(100);    break;
case 3:
            if(toggleState_3 == 0){
digitalWrite(relay_3, HIGH); // turn on relay
3          toggleState_3 = 1;
            }else{
digitalWrite(relay_3, LOW); // turn off
relay 3      toggleState_3 = 0;

```

```

        }
delay(100);
break;    case 4:
        if(toggleState_4 == 0){
digitalWrite(relay_4, HIGH); // turn on relay
4        toggleState_4 = 1;
        }
        else{                digitalWrite(relay_4,
LOW); // turn off relay 4
        toggleState_4 = 0;
        }
delay(100);    break;

        default : break;
    }

}

```

```
void loop() {  
  
    if (digitalRead(mswitch_1) == LOW){  
        delay(200);  
        relayOnOff(1);  
    }  
    else if (digitalRead(mswitch_2) == LOW){  
delay(200);    relayOnOff(2);  
    }  
    else if (digitalRead(mswitch_3) == LOW){  
delay(200);    relayOnOff(3);  
    }  
    else if (digitalRead(mswitch_4) == LOW){  
delay(200);    relayOnOff(4);  
    }  
  
    if (irrecv.decode(&results)) {  
switch(results.value){
```

```
case 0xFD08F7:
    relayOnOff(1);        break;
case 0xFD8877:
    relayOnOff(2);        break;
case 0xFD48B7:
    relayOnOff(3);        break;
case 0xFD28D7:
    relayOnOff(4);        break;
default : break;

    }

    irrecv.resume();
}
}
```