**Block Diagram:**

PIR SENSOR

LIGHT DEPEND RESISTOR

ESP8266

AMPLIFIER

VOLTAGE DIVIDER

LOGIC LEVEL CONVERTER

ARDUINO-UNO DEVELOPMENT BOARD

RELAY WITH DRIVER

RELAY WITH DRIVER

SERVO DRIVER

AC LOAD (1)

AC LOAD (2)

SERVO MOTOR

LCD DISPLAY

WEBPAGE

INTERNET CONNECTIVITY

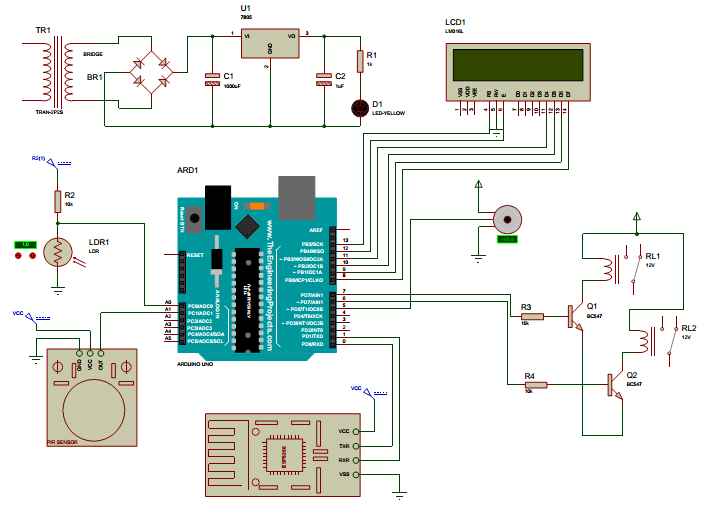
PC (or) LAPTOP

SMART PHONE APPLICATION

INTERNET CONNECTIVITY

SMART PHONE

**Circuit Diagram:**



**Cost Estimation:**

|  |  |  |  |
| --- | --- | --- | --- |
| **SL.NO** | **COMPONENT** | **DETAILS** | **COST** |
| 1 | Arduino – UNO and Related tools | Arduino uno board, USB connector, Liquid crystal display and adaptor | 650/- |
| 2 | Power Supply and Related Components | Step down transformer, Power Supply Board and PCB | 560/- |
| 3 | Relay Driver Related Components | Driver Circuit, Relay, Servo Driver, Servo Motor and PCB | 250/- |
| 4 | ESP Module and Related Tools | ESP Module, Voltage Regulators, Logic Level converters and PCB | 350/- |
| 5 | Sensor and Related Components | PIR Motion Sensor, Light Depend Resistor and Voltage divider | 740/- |
|  |  | Total | 2550/- |

**Program Code:**

#include <Arduino.h>

#include <avr/io.h>

#include <Servo.h>

#include "utils.h"

#include "lcd.h"

#include "adc.h"

#include "uart.h"

#include "esp.h"

#include "mqtt.h"

#ifndef LCDPinConfig

#define LCDPinConfig

#define RS 13

#define EN 12

#define D4 11

#define D5 10

#define D6 9

#define D7 8

#endif

#ifndef SensorConfig

#define SensorConfig

#define LightSensor 0

#define MotionSensor A1

#endif

#ifndef OutputConfig

#define OutputConfig

#define LampRelay 6

#define FanRelay 7

#endif

#ifndef ServoConfig

#define ServoConfig

#define ServoMotor 5

#define DoorClose 90

#define DoorOpen 180

#endif

#ifndef ESPConfig

#define ESPConfig

#define ESPTransmitter 1

#define ESPReceiver 0

#define ESPBaudRate 9600

#endif

Servo MyServo;

unsigned char LightLevel, ISDataReceived, ISAutoMode = True;

unsigned int LastReceivedData;

char ISLampOn, ISFanOn;

void setup()

{

pinMode(LampRelay, OUTPUT);

pinMode(FanRelay, OUTPUT);

digitalWrite(LampRelay, LOW);

digitalWrite(FanRelay, LOW);

MyServo.attach(ServoMotor);

MyServo.write(DoorClose);

pinMode(MotionSensor, INPUT);

ADC\_Initialize();

LCD\_Initialize(RS, EN, D4, D5, D6, D7);

LCD\_Disp(0x80,"SMART HOME AUTO ");

LCD\_Disp(0xC0,"SECURITY SYSTEM ");

DelayMS(2500); LCDClear();

if(ESP\_Initialize(ESPBaudRate)) LCD\_Disp(0x80,"ESP FOUND...");

else LCD\_Disp(0x80,"ESP NOT FOUND"); DelayMS(500); LCDClear();

if(ESP\_Mode(3)) LCD\_Disp(0x80,"MODE CONFIG");

else LCD\_Disp(0x80,"MODE ERROR"); DelayMS(500); LCDClear();

if(ESP\_WiFi()) LCD\_Disp(0x80,"WIFI CONNECTED");

else LCD\_Disp(0x80,"WIFI ERROR"); DelayMS(500); LCDClear();

if(ESP\_Start(ServerName, ServerPort)) LCD\_Disp(0x80,"SERVER CONNECTED");

else LCD\_Disp(0x80,"SERVER ERROR"); DelayMS(500); LCDClear();

if(MQTT\_Connect()) LCD\_Disp(0x80,"MQTT ESTABLISHED");

else LCD\_Disp(0x80,"MQTT FAILED"); DelayMS(500); LCDClear();

MQTT\_Publish(VariableLabel1, False);

MQTT\_Publish(VariableLabel2, False);

MQTT\_Publish(VariableLabel3, False);

MQTT\_Subscribe(VariableLabel4, True);

DelayMS(500); Serial\_Flush();

}

void loop()

{

ISDataReceived = MQTT\_Receive(VariableLabel4, &LastReceivedData);

LightLevel = MapValue(ReadRawADC(LightSensor, 10), 0, 1023, 0, 100);

LCD\_Disp(0x80, (ISAutoMode ?"\*\*\*AUTO-MODE\*\*\* " :"\*\*MANUAL-MODE\*\* "));

LCD\_Disp(0xC0,"L:"); LCD\_Decimal(0xC2, LightLevel, 3, DEC);

LCD\_Disp(0xC6,"M:"); LCD\_Disp(0xC8, (digitalRead(MotionSensor) ?"DETECT" :"NORMAL"));

if(ISAutoMode && LightLevel < 50 && !ISLampOn)

{

ISLampOn = True;

digitalWrite(LampRelay, HIGH);

MQTT\_Publish(VariableLabel1, ISLampOn);

Serial\_Flush();

}

else if(ISAutoMode && LightLevel > 90 && ISLampOn)

{

ISLampOn = False;

digitalWrite(LampRelay, LOW);

MQTT\_Publish(VariableLabel1, ISLampOn);

Serial\_Flush();

}

if(ISAutoMode && digitalRead(MotionSensor) && !ISFanOn)

{

ISFanOn = True;

digitalWrite(FanRelay, HIGH);

MQTT\_Publish(VariableLabel2, ISFanOn);

Serial\_Flush();

}

else if(ISAutoMode && !digitalRead(MotionSensor) && ISFanOn)

{

ISFanOn = False;

digitalWrite(FanRelay, LOW);

MQTT\_Publish(VariableLabel2, ISFanOn);

Serial\_Flush();

}

if(ISDataReceived)

{

switch(LastReceivedData)

{

case 1:

if(ISAutoMode) ISAutoMode = False;

break;

case 2:

if(!ISAutoMode) ISAutoMode = True;

break;

case 3:

if(!ISLampOn) ISLampOn = True;

digitalWrite(LampRelay, HIGH);

MQTT\_Publish(VariableLabel1, ISLampOn);

break;

case 4:

if(ISLampOn) ISLampOn = False;

digitalWrite(LampRelay, LOW);

MQTT\_Publish(VariableLabel1, ISLampOn);

break;

case 5:

if(!ISFanOn) ISFanOn = True;

digitalWrite(FanRelay, HIGH);

MQTT\_Publish(VariableLabel2, ISFanOn);

break;

case 6:

if(ISFanOn) ISFanOn = False;

digitalWrite(FanRelay, LOW);

MQTT\_Publish(VariableLabel2, ISFanOn);

break;

case 7:

MyServo.write(DoorOpen);

MQTT\_Publish(VariableLabel3, True);

break;

case 8:

MyServo.write(DoorClose);

MQTT\_Publish(VariableLabel3, False);

break;

default: break;

}

Serial\_Flush();

}

}