

- **Project code :**

```
#define relay_pin 7
#define echo 8
#define trig 9
#define servo_pin 10
#define receiver_pin 11

#define remote_open_button 0xFF30CF
#define remote_close_button 0xFF18E7
#define servo_open_angle 100
#define servo_close_angle 0

#include <IRremote.h>
#include <Servo.h>
#include "HCSR04.h"

IRrecv irrecv(receiver_pin);
decode_results results;
Servo servo;
HCR04ProxSensor distanceSensor(echo , trig);

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  Serial.println("Enabling IRin");
  irrecv.enableIRIn(); // Start the receiver
  Serial.println("Enabled IRin");
```

```

servo.attach(servo_pin);
pinMode(relay_pin,OUTPUT);

}

void loop() {
// put your main code here, to run repeatedly:
long remote_reading; //declare long variable to save signal of remote in it
double distance = distanceSensor.readSensor(); //this variable used to save distance measured by
ultrasonic sensor

if( distance < 5.0){
digitalWrite(relay_pin,HIGH); //out high signal on relay
}else{
digitalWrite(relay_pin,LOW); //out low signal on relay
}
if (irrecv.decode(&results)) {
remote_reading = results.value; //read the value of button that pressed
if(remote_reading == remote_open_button){
servo.write(servo_open_angle);
delay(15);
}else if( remote_reading == remote_close_button){
servo.write(servo_close_angle);
delay(15);
}
irrecv.resume(); // Receive the next value
delay(100);
}
}

```

- **.cpp**

```
#include "HCSR04.h"

HCR04ProxSensor::HCR04ProxSensor(int echo_pin, int trigger_pin){
    triggerPin = trigger_pin;
    echoPin = echo_pin;
    pinMode(triggerPin, OUTPUT); // Sets the trigPin as an Output
    pinMode(echoPin, INPUT); // Sets the echoPin as an Input
}

double HCR04ProxSensor::readSensor(){
    double waittingTime;
    static double distance=0;
    setLastValue(distance);
    digitalWrite(triggerPin,LOW);
    delayMicroseconds(60);
    digitalWrite(triggerPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(triggerPin, LOW);
    waittingTime = pulseIn(echoPin, HIGH);
    distance = waittingTime*speedOfSound*0.0001/2;
    return distance;
}

void HCR04ProxSensor::setLastValue(double last_value){
    LastValue = last_value;
}

double HCR04ProxSensor::getLastValue(void){
    return LastValue;
}
```

- **.h**

```
/*
 * this header file was written for proximity sensor HCSR04
 * this
 */
#ifndef HCSR04_H
#define HCSR04_H

#if (ARDUINO >= 100)
    #include "Arduino.h"
```

```
#else
  #include "Wprogram.h"
#endif

#ifndef speedOfSound
#define speedOfSound 340
#endif

class HCR04ProxSensor{
public:

  HCR04ProxSensor(int echo_pin, int trigger_pin);
  double readSensor(void);
  double getLastValue(void);

private:
  int echoPin,triggerPin;
  double LastValue;
  void setLastValue(double last_vale);

};

#endif
```