(Change pic)

Fully Functioning House Reference Manual

Asad Bhatti

June 18, 2018

Peter Le

TEJ3MR

**Table of Contents**

Objective…………………………………………………………page 2

Background Research………………………………………………….

Materials…………………………………………………………………

Experimental Procedure……………………………………………….

Conclusion……………………………………………………………….

**Objective:**

Our objective is to create functioning house using buttons to control internal lighting and a security system composed of a sensor and a buzzer.

**Background Research:**

Buttons:

Buttons have been used in many of our modern machines, from arcade machines to keyboards. These buttons are known to be connected through mechanical linkage. This means that the when one button is pressed it causes another to be released such as a start and stop button. When the stop button is pressed it will release the start button breaking the circuit and stopping the circuit from functioning.

<https://en.wikipedia.org/wiki/Push-button>

<https://uk.rs-online.com/web/generalDisplay.html?id=solutions/push-button-switches-overview>

Sensors and Buzzers:

Wiring:

**Materials:**

1. Ultrasonic sensor x1
2. Male to Male wires x(idk)
3. Female to Male wires x(idk)
4. Active Buzzer x1
5. LED x4
6. Buttons x8
7. Cardboard
8. Glue gun x1
9. Electric Tape x1

**Significance of materials:**

The ultrasonic sensor is use to detect intruders by detecting any motion through the main door of the house. Anything within a radius of 10cm will trigger the ultrasonic sensor to signal the alarm. The alarm is sounded of through the active buzzer which makes a constant beeping noise as long as there is anyone within a 10cm radius. The house consists of additional four rooms, each with LED’s that turn on and off when their corresponding button is pressed. The other materials such as duct tape, glue gun and cardboard are used to build the house and cover up any wires.

**Experimental Procedure:**

Arduino Code:

**//Define Variables (Lights)**

const int button1Apin = 2;

const int button2Apin = 3;

**//Buttons for LED A**

const int button1Bpin = 4;

const int button2Bpin = 5;

**//Buttons for LED B**

const int button1Cpin = 6;

const int button2Cpin = 7;

**//Buttons for LED C**

const int button1Dpin = 8;

const int button2Dpin = 9;

**//Buttons for LED D**

const int ledA = A1;

**//LED for Room A**

const int ledB = A2;

**//LED for Room B**

const int ledC = A3;

**//LED for Room C**

const int ledD = A4;

**//LED for Room D**

**//Define Variables (Security System)**

const int trigPin = 10;

const int echoPin = 11;

const int led = 12;

const int BuzzPin = A0;

**// Variables for buzzer duration and the distance**

long duration;

int distance;

void setup()

{

 pinMode(ledA, OUTPUT);

 pinMode(ledB, OUTPUT);

 pinMode(ledC, OUTPUT);

 pinMode(ledD, OUTPUT);

**//LEDs are the Outputs**

 pinMode(button1Apin, INPUT\_PULLUP);

 pinMode(button2Apin, INPUT\_PULLUP);

 pinMode(button1Bpin, INPUT\_PULLUP);

 pinMode(button2Bpin, INPUT\_PULLUP);

 pinMode(button1Cpin, INPUT\_PULLUP);

 pinMode(button2Cpin, INPUT\_PULLUP);

 pinMode(button1Dpin, INPUT\_PULLUP);

 pinMode(button2Dpin, INPUT\_PULLUP);

**//Buttons are Inputs**

 pinMode(trigPin, OUTPUT); **// Sets the trigPin as an Output**

 pinMode(echoPin, INPUT); **// Sets the echoPin as an Input**

 pinMode(led, OUTPUT);

 pinMode(BuzzPin, OUTPUT);

 Serial.begin(9600);

**//Sensor Inputs/Outputs**

}

void loop() {

**//"Remote" with buttons to control lights**

 if (digitalRead(button1Apin) == LOW) {

   digitalWrite(ledA, HIGH);           **//Turn Light On**

 }

 if (digitalRead(button2Apin) == LOW) {

   digitalWrite(ledA, LOW);           **//Turn Light Off**

 }

**//Buttons to turn Room A light on/off**

 if (digitalRead(button1Bpin) == LOW) {

   digitalWrite(ledB, HIGH);          **//Turn Light On**

 }

 if (digitalRead(button2Bpin) == LOW) {

   digitalWrite(ledB, LOW);           **//Turn Light Off**

 }

**//Buttons to turn Room B light on/off**

 if (digitalRead(button1Cpin) == LOW) {

   digitalWrite(ledC, HIGH);         **//Turn Light On**

 }

 if (digitalRead(button2Cpin) == LOW) {

   digitalWrite(ledC, LOW);           **//Turn Light Off**

 }

**//Buttons to turn Room C light on/off**

 if (digitalRead(button1Dpin) == LOW) {

   digitalWrite(ledD, HIGH);          **//Turn Light On**

 }

 if (digitalRead(button2Dpin) == LOW) {

   digitalWrite(ledD, LOW);           **//Turn Light Off**

 }

**//Buttons to turn Room D light on/off**

 for(int i=15;i<=165;i++){

 distance = calculateDistance();

 Serial.print(",");

 Serial.print(distance);

 if(distance <10 && distance > 0) {

digitalWrite(12,HIGH);

digitalWrite(A0,HIGH);

}

else

{

digitalWrite(12,LOW);

digitalWrite(A0,LOW);

}

 Serial.print(".");

 }

}

int calculateDistance(){

 digitalWrite(trigPin, LOW);

 delayMicroseconds(2);

 // Sets the trigPin on HIGH state for 10 micro seconds

 digitalWrite(trigPin, HIGH);

 delayMicroseconds(10);

 digitalWrite(trigPin, LOW);

 duration = pulseIn(echoPin, HIGH); // Reads the echoPin, returns the sound wave travel time in microseconds

 distance= duration\*0.034/2;

 return distance;

}

Wiring Diagram:

Physical Layout:

**Conclusion:**

answer engineering design process, discuss project, application & future uses, sources of error   underlined = done

While constructing the house, there could have been numerous errors that limited our project. First of all, our group had planned to add a lot more features than we actually added but due the many errors that occurred, we had to resort to a quicker and easier to build model house. The error that had the biggest effect on our project was, non-functioning components. For example, our group had decided to work on one component of the house at a time, and we had planned to begin with security alarm system. Our plan was, we would only move on to the next component once we had finished the security alarm system but numerous days were wasted on trying to fix our code and wiring setup, which was not at fault, but our buzzer was the source of the error. When we switched our buzzer with a buzzer from that of a different group, we had noticed our code and wiring setup was not at fault but by this time many days were already wasted. This resorted us too limiting our model house and reducing many components. Another source of error that limited our project derived from the breadboard. For example, our group had only one breadboard to work with and had to use it efficiently and half way into completing our project we noticed that our breadboard will not be big enough to support all the components we envisioned. This also resorted us into limiting our house into less components then that were mentioned in our proposal. Finally another source of error would be the wireless remote. The main purpose of the project was to build a house using an ir sensor and a remote. But the problem we continuously faced was a working remote. Regardless of the numerous amounts of code, including the built-in