**Smart Door**

1. **Introduction**

Answering the door is a big issue for some people while they are working or they have assignments to do and do not want to waste their time in opening the door. Some visitors come at the wrong time and some of them not welcome. In this case, answering the door is a big issue when the visitor is unknown and cannot recognize whether this visitor is an important or not and cannot decide to open the door or ignore it. The aim of this experiment to design a smart lock that determines whether will be required to answer the door or not. This lock should know the following:

1. If there is and isn’t a visitor.
2. Should address the visitors.
3. Determine the reason for visiting.
4. Allow the visitor to enter or send him/her away.
5. **Required Component**

Here are the main component that used in this experiment:

1. **Arduino Uno.**



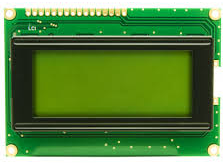
1. **Servo Motor.**



1. **Pressure Sensor**



1. **LCD**



1. **Touch Screen.**



1. **RBG LED.**



1. **Other Parts to build a prototype.**
2. **Design Process**

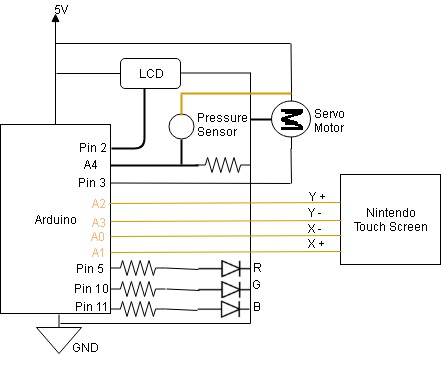
Our design process is to find the best solution to the four points proposed up. To overcome these problems, we started looking for the best electronic parts that could solve them. Here is a description for each part related to each problem:

|  |  |  |  |
| --- | --- | --- | --- |
| Problem | Description | Part used | Description |
| 1 | Determine whether a visitor at the door or not | Pressure Sensor | To determine the weight of human |
| 2 | Send a message to the visitor | LCD | Display words to the visitor |
| 3 | Visitor respond to the system | Touch screen | To choose the reason for visiting. |
| 4 | Inform the visitor if he/she could either enter or should leave | RBG LED | * Red - visitor should leave. * Green- visitor allowed to enter or I am coming to open. * Blue – leave the item at the door. |

**Note**: permissions to enter or leave:

|  |  |  |
| --- | --- | --- |
|  | Type of visitors | Permissions |
| 1 | Family. | Enter |
| 2 | Salesperson. | Leave |
| 3 | Delivery person. | Leave item at the door |
| 4 | Ice cream seller. | Leave |

* **Circuit Schematic**



* **Circuit Diagram**



1. **Methodology**

The following steps illustrate how the experiment works:

* First we started by instigating if there is a visitor at the door or not. For this purpose, a pressure sensor fixed underneath a door mat.
* Once the visitor stand on the mat, the pressure sensor starts reading the weight of a visitor.
* The system will be activated once the weight registered is higher than the threshold.
* The visitor can see a greeting on the LCD and will be asked the reason for visiting.
* Four options will be showed up on the touch screen to allow the visitor choose whether he is a family member or delivery person or salesperson or Ice cream seller.
* The visitor will pick up one of the four options from the touch screen and the rest of the smart lock will be activated.

The second procedure of this experiment is lighting the RGB LED according to the identity of the visitor and this visitor should leave or enter:

* If family members: green LED “ON” > permitted immediate entry > unlock the door using the servo motor > welcoming message will showed on the LCD.
* If a salesperson: Red Led “ON” > not permitted entry > the locked using the servo motor > a message will be showed up on LCD asking them to leave.
* If a delivery person: Blue LED “ON” > not permitted entry > servo motor locks the door > > a message will be showed up on LCD asking them to leave the item at the door.
* If ice cream seller: Green LED “ON” > not permitted to entry > servo unlocks the door > a message will be showed up on LCD asking him to wait until I open the door.

The Arduino used here to control the system by a series of “if” and “else” statements once receives the readings from the touch screen the pressure system.

1. **Arduino Code**

//include libraries

#include "stdint.h"

#include "Servo.h"

#include "TouchScreen.h"

#include "Arduino.h"

//touch screen setup

#define YP A2 // must be an analog pin, use "An" notation!

#define XM A3 // must be an analog pin, use "An" notation!

#define YM A0 // can be a digital pin

#define XP A1 // can be a digital pin

#define MINPRESSURE 10

#define MAXPRESSURE 1000

TouchScreen ts = TouchScreen(XP, YP, XM, YM, 300);

//LCD setup

#if defined(ARDUINO\_ARCH\_SAMD) || defined(\_\_SAM3X8E\_\_)

#define lcd Serial1

#else

#include <SoftwareSerial.h>

SoftwareSerial lcd = SoftwareSerial(0,2);

#endif

//Pressure sensor setup

int fsrAnalogPin = A4; // FSR is connected to analog 0

int fsrReading; // the analog reading from the FSR resistor divider

//RGB LED setup

const int RLED = 5;

const int GLED = 10;

const int BLED = 11;

//servo setup

int servoPin = 3;

Servo servo;

void setup(void) {

Serial.begin(9600); //use to determine values for sensors

//set RGB LED as outputs

pinMode(RLED, OUTPUT);

pinMode(GLED, OUTPUT);

pinMode(BLED, OUTPUT);

pinMode(servoPin,OUTPUT);

//setup servo

servo.attach(servoPin);

//setup LCD

lcd.begin(9600);

lcd.write(0xFE);

lcd.write(0xD1);

lcd.write(16);

lcd.write(2);

delay(10);

//set contrast to 200

lcd.write(0xFE);

lcd.write(0x50);

lcd.write(200);

delay(10);

//max brightness

lcd.write(0xFE);

lcd.write(0x99);

lcd.write(255);

delay(10);

//turn off cursors

lcd.write(0xFE);

lcd.write(0x4B);

lcd.write(0xFE);

lcd.write(0x54);

//create custom character

lcd.write(0xFE);

lcd.write(0x4E);

lcd.write((uint8\_t)0);

lcd.write((uint8\_t)0x00);

lcd.write(0x0A);

lcd.write(0x15);

lcd.write(0x11);

lcd.write(0x11);

lcd.write(0x0A);

lcd.write(0x04);

lcd.write((uint8\_t)0x00);

delay(10);

// clear screen

lcd.write(0xFE);

lcd.write(0x58);

delay(10);

// go 'home'

lcd.write(0xFE);

lcd.write(0x48);

delay(1000);

}

void loop(void) {

//read values Pressure Sensor

fsrReading = analogRead(fsrAnalogPin);

delay(500);

//set pikachu weight

int pikachuWeight=100;

//if there is a visitor

if (fsrReading > pikachuWeight){

lcd.print("Hello!");

delay(1000);

lcd.write(0xFE);

lcd.write(0x58);

delay(10);

TSPoint p = ts.getPoint();

if (p.z > ts.pressureThreshhold) {

Serial.print("X = "); Serial.print(p.x);

Serial.print("\tY = "); Serial.print(p.y);

Serial.print("\tPressure = "); Serial.println(p.z);

}

int xval = p.x;

int yval = p.y;

if (xval > 550 && xval <940 && yval > 180 && yval < 430)

{

salesPerson();

delay(1000);

}

else if (xval > 550 && xval <940 && yval > 550 && yval < 830)

{

delivery();

delay(1000);

}

else if (xval > 110 && xval <500 && yval > 550 && yval < 830)

{

relatives();

delay(1000);

}

else if (xval > 110 && xval <500 && yval > 180 && yval < 430)

{

icecream();

delay(1000);

}

}

else if (fsrReading<pikachuWeight){

noVisitors();

delay(1000);

}

}

void salesPerson(){

digitalWrite(RLED,HIGH);

digitalWrite(BLED,LOW);

digitalWrite(GLED,LOW);

lcd.println("Not available.");

servo.write(0);

delay(3000);

lcd.write(0xFE);

lcd.write(0x58);

}

void delivery(){

analogWrite(RLED,200);

analogWrite(BLED,0);

analogWrite(GLED,90);

lcd.println("Leave it.");

servo.write(0);

delay(3000);

lcd.write(0xFE);

lcd.write(0x58);

}

void relatives(){

digitalWrite(RLED,LOW);

digitalWrite(BLED,LOW);

digitalWrite(GLED,HIGH);

lcd.println("Come inside!");

servo.write(180);

delay(3000);

lcd.write(0xFE);

lcd.write(0x58);

}

void icecream(){

digitalWrite(RLED,LOW);

digitalWrite(BLED,LOW);

digitalWrite(GLED,HIGH);

lcd.println("Be right there!");

servo.write(180);

delay(3000);

lcd.write(0xFE);

lcd.write(0x58);

}

void noVisitors(){

digitalWrite(RLED,LOW);

digitalWrite(BLED,LOW);

digitalWrite(GLED,LOW);

servo.write(0);

}