Control Security Door by Sonic Range Finder

There are 2 modes in this project: Normal mode and Security mode, 2 modes are switched by press the button.

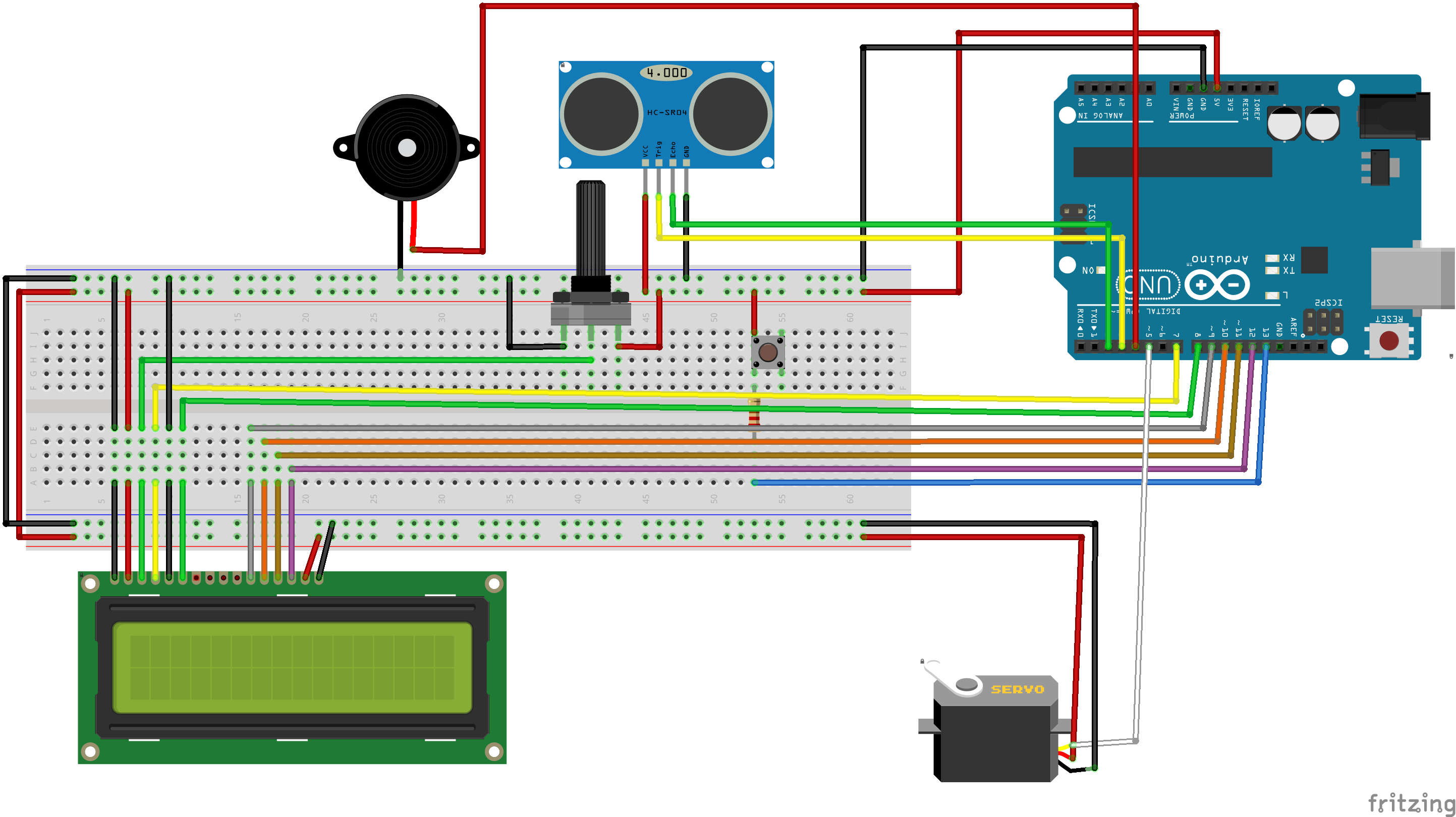
* In Normal mode: sonic range finder will detect the distance of people who come to the door. If the distance is closer than 20cm (approximate 2m in real life), LCD screen will display “Hello” on 1st line, “Wellcome” on 2nd line and the motor will rotate 90 degrees (open the door) and stay in that position until that person go away (distance is more than 20cm from sensor) then motor will rotate 90 degrees in reverse (close the door), and the screen will say “Goodbye” on 1st line, “See you again” on 2nd line.
* In Security Mode: if the door is still open, this will act like emergency situation: the door will close immediately, depend on the distance of target to the sensor and we have different behavior.
* If the distance is more than 20cm, screen will display “CLOSE” and “See you later”.
* If the distance is closer than 20cm, screen will display “Sorry” and “We’re close”
* If the distance is closer than 10cm, screen will display “Warning” and buzzer will raise alert sound until the target get away more than 20cm from sensor.

Hardware Required

* Arduino or Genuino Board
* LCD Screen
* 10k ohm potentiometer
* 220 ohm resistor
* Ultrasonic Range Finder SRF05
* Button
* Piezo buzzer
* Servo Motor
* hook-up wires
* breadboard

Circuit

* To wire your LCD screen to your board, connect the following pins:
* LCD RS pin to digital **pin 7**
* LCD Enable pin to digital **pin 8**
* LCD D4 pin to digital **pin 9**
* LCD D5 pin to digital **pin 10**
* LCD D6 pin to digital **pin 11**
* LCD D7 pin to digital **pin 12**
* Additionally, wire a 10k pot to +5V and GND, with it's wiper (output) to LCD screens VO pin. To power the backlight of the display, connect pin A to +5V and K to ground.
* The 5V (VCC) pin of the SRF05 is connected to the 5V pin on the board, the GND pin is connected to the GND pin, the TRIG pin is connected to digital **pin 3** on the board and the ECHO pin is connected to digital **pin 2** on the board.
* Connect the longer leg of buzzer to +5V and the other to ground
* One leg of button is connected to +5V, the other is connected to ground and **pin 13** through a resistor on breadboard.
* Servo motors have three wires: power, ground, and signal. The power wire is typically red, and should be connected to the 5V pin on the Arduino or Genuino board. The ground wire is typically black or brown and should be connected to a ground pin on the board. The signal pin is typically yellow, orange or white and should be connected to pin 4 on the board.



Code

#include <LiquidCrystal.h>

#include <Servo.h>

Servo myservo;

int rs = 7;

int en = 8;

int d5 = 10;

int d6 = 11;

int d7 = 12;

int d4 = 9;

int trigPin = 3;

int echoPin = 2;

int pingTravelTime;

float distance;

int pos, buttonState;

int mode;

int buttonPin = 13;

int buzzState = LOW;

bool lastButtonState = LOW;

bool newButtonState = LOW;

LiquidCrystal lcd(rs,en,d4,d5,d6,d7);

void setup() {

// put your setup code here, to run once:

lcd.begin(16,2);

pinMode(buttonPin, INPUT);

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

pinMode(4, OUTPUT);

myservo.attach(5);

Serial.begin(9600);

}

void normalMode(){

digitalWrite(trigPin, LOW);

delayMicroseconds(10);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

pingTravelTime = pulseIn(echoPin, HIGH);

delay(25);

distance = ((pingTravelTime\*350.\*100.)/1000000)/2;

Serial.println(distance);

if(distance <20.){

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Hello");

lcd.setCursor(0,1);

lcd.print("Wellcome");

while (pos <=90) {

myservo.write(pos);

Serial.println(pos);

delay(15);

if(pos == 90){

myservo.write(90);

delay(1);

break;

}pos += 1;

}

}else{

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Goodbye");

lcd.setCursor(0,1);

lcd.print("See you again");

Serial.println(pos);

while(pos > 0){

myservo.write(pos);

Serial.println(pos);

delay(15);

pos -=1;

}

}

}

void warningSound(){

for(int i = 1000; i >= 0; i--){

digitalWrite(4, HIGH);

delayMicroseconds(i);

digitalWrite(4, LOW);

delayMicroseconds(i);

}for(int i = 0; i <= 1000; i++){

digitalWrite(4, HIGH);

delayMicroseconds(i);

digitalWrite(4, LOW);

delayMicroseconds(i);

}

}

void securityMode(){

digitalWrite(trigPin, LOW);

delayMicroseconds(10);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

pingTravelTime = pulseIn(echoPin, HIGH);

delay(25);

distance = ((pingTravelTime\*350.\*100.)/1000000)/2;

if(distance <20.){

if(distance < 10.){

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Warning");

warningSound();

lcd.setCursor(0,1);

lcd.print("Go Back or DIE");

warningSound();

// delay(500);

}else{

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Sorry");

lcd.setCursor(0,1);

lcd.print("We're closed");

}

}else{

digitalWrite(4, LOW);

lcd.clear();

lcd.setCursor(0,0);

lcd.print("CLOSED");

lcd.setCursor(0,1);

lcd.print("See you later");

}

// delay(150);

}

int buttonListener() {

int ButtonState = digitalRead(buttonPin);

Serial.println(newButtonState);

if (ButtonState == HIGH) {

delay(150);

newButtonState = !newButtonState;

}

if(newButtonState == HIGH){

return 2;

}else{

return 1;

}

}

void loop() {

int mode = buttonListener();

Serial.println(mode);

switch(mode){

case 1:

normalMode();

break;

case 2:

while(pos > 0){

myservo.write(pos);

Serial.println(pos);

delay(15);

pos -=1;

}

securityMode();

break;

}

}