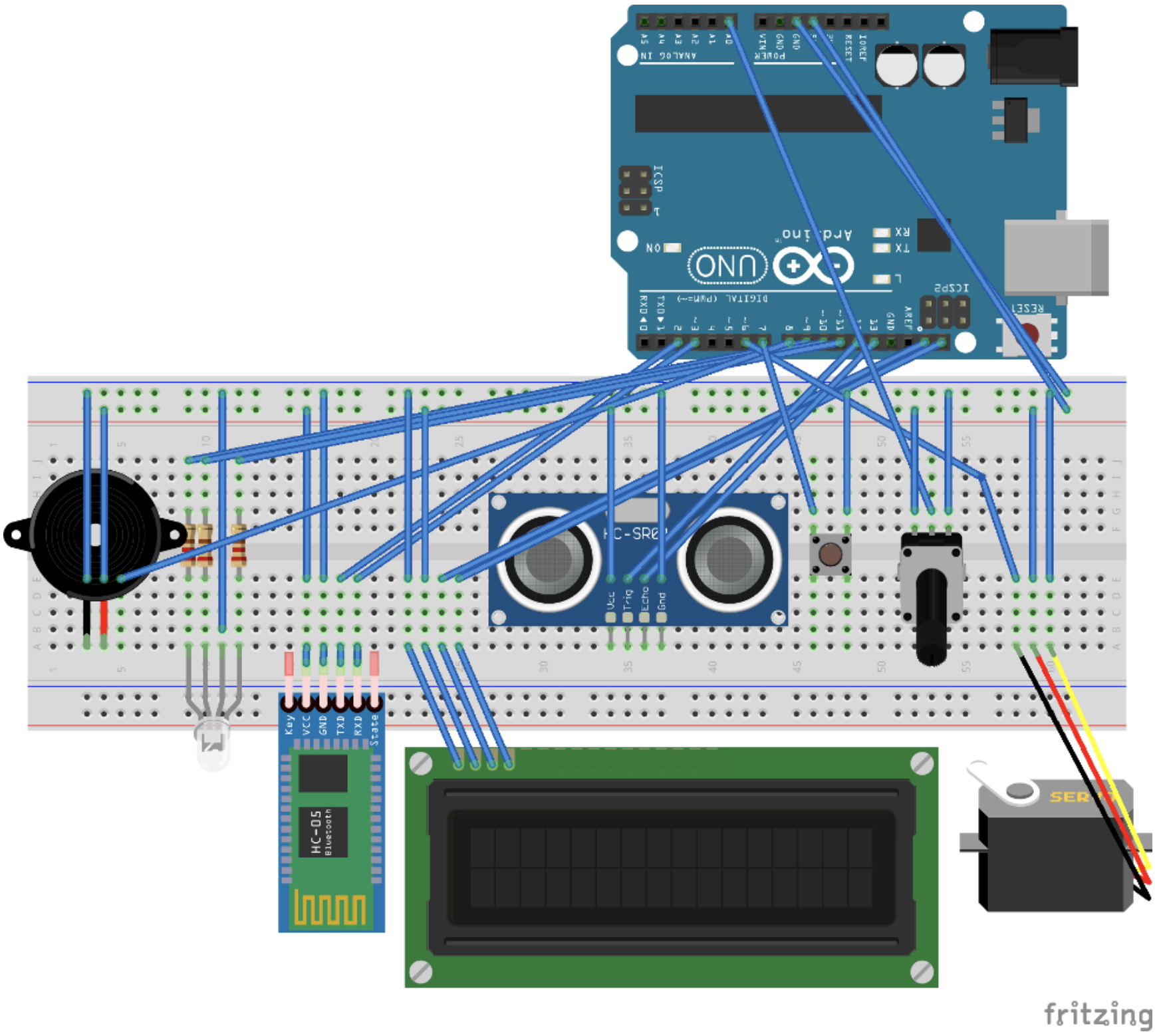
컴퓨터 시스템 입문 프로젝트 데모 보고서

제목 : HAD (Hufs Automatic Door)

기능

1. 자동문 앞에 사람이 있으면 자동문이 열림
2. 버튼을 누르면 자동문이 열림
3. 자동문이 열릴 때 소리가 나며 LED 에 불이 들어옴
4. 가변 저항을 통해 초음파 센서의 인식 거리를 조절 함
5. LCD 를 통해 초음파 센서의 인식 거리, 문이 열린 횟수, 앱의 제어 상황을 보여줌
6. 블루투스 통신을 통해 기기를 제어함 (문 열기, LED 색상 변경, 문이 열린 횟수 초기화, 기기 종료, 실행)

회로도



소스코드

```ino
#include <LiquidCrystal_I2C.h>
#include <Servo.h>
#include <SoftwareSerial.h>

int echoPin = 13;
int trigPin = 12;
int redPin = 11;
int greenPin = 10;
int bluePin = 9;
int buzzer = 8;
int pushButton = 7;
int rx = 3;
int tx = 2;
SoftwareSerial BTSerial(tx, rx);
bool isOpen = true;
int lastBaseLine = 0;
bool isMoveBaseLine = false;
int doorCount = 0;
bool isOn = true;
char inData[20];
char inChar = -1;
int readIndex = 0;
Servo servo;

LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup()
{
  Serial.begin(9600);
  BTSerial.begin(9600);
  pinMode(echoPin, INPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
  pinMode(buzzer, OUTPUT);
  pinMode(pushButton, INPUT_PULLUP);
  servo.attach(6);
  lcd.init();
  lcd.noBacklight();
  setBaseLCD(10);
}

void loop()
{
  if (Serial.available()) {
    BTSerial.write(Serial.read());
  }

  if (BTSerial.available()) {
    char bt = BTSerial.read();
    if (bt == '\n') {
      for (int i = 0; i < readIndex; i++) {
        Serial.print((int)inData[i]);
        Serial.print(" ");
      }
      String result(inData);
      if (result == "move") {
        moveDoor();
        setBaseLCD(lastBaseLine);
      } else if (result == "reset") {
        doorCount = 0;
        setBaseLCD(lastBaseLine);
      } else if (result == "red") {
        setLCD("red");
        setColor(255, 0, 0);
        delay(2000);
        setBaseLCD(lastBaseLine);
      } else if (result == "green") {
        setLCD("green");
        setColor(0, 255, 0);
        delay(2000);
        setBaseLCD(lastBaseLine);
      } else if (result == "blue") {
        setLCD("blue");
        setColor(0, 0, 255);
        delay(2000);
        setBaseLCD(lastBaseLine);
      } else if (result == "off") {
        if (isOn) {
          setColor(0, 0, 0);
          setLCD("service off ...");
          isOn = false;
        }
      } else if (result == "on") {
        if (!isOn) {
          setLCD("SERVICE ON !!!");
          isOn = true;
          delay(2000);
          setBaseLCD(lastBaseLine);
        }
      }
      for (int i = 0; i < 19; i++) {
        inData[i] = 0;
      }
      readIndex = 0;
    } else {
      inData[readIndex] = bt; // Store it
      readIndex++; // Increment where to write next
    }
  }
  if (isOn) {
    int buttonState = digitalRead(pushButton);
    int regist = analogRead(A0);
    int baseLine = map(regist, 0, 1023, 0, 100);
    int distance = getDistance();

    if (buttonState == 0) {
      Serial.println(buttonState);
      moveDoor();
    }

    if (distance <= baseLine) {
      Serial.println(distance);
      moveDoor();
    }

    if (lastBaseLine != baseLine) {
      isMoveBaseLine = true;
      lastBaseLine = baseLine;
    } else {
      if (isMoveBaseLine == true) {
        setBaseLCD(baseLine);
      }
    }
  }
}

void setBaseLCD(int baseLine) {
  isMoveBaseLine = false;
  setColor(255, 0, 255);
  lcd.clear();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("Distance  : ");
  lcd.print(baseLine);
  lcd.print("cm");
  lcd.setCursor(0, 1);
  lcd.print("Door Move : ");
  lcd.print(doorCount);
}

void reset() {
  doorCount++;
  setColor(255, 0, 255);
}

void moveDoor() {
  Serial.println(isOpen);
  if (isOpen) {
    openDoor();
  }
  else {
    closeDoor();
  }

  isOpen = !isOpen;
  reset();
}

void openDoor() {
  setLCD("Door Open");
  setTone(262, 300);
  setColor(0, 255, 0);
  for (int angle = 0; angle <= 179; angle++) {
    servo.write(angle);
    delay(20);
  }
}

void closeDoor() {
  setLCD("Door Close");
  setTone(262, 300);
  setColor(255, 0, 0); // red
  for (int angle = 179; angle >= 0; angle--) {
    servo.write(angle);
    delay(20);
  }
}

int getDistance() {
  float duration, distance;

  digitalWrite(trigPin, HIGH);
  delay(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);
  distance = ((float)(340 * duration) / 10000) / 2;
  return distance;
}

void setLCD(String message) {
  lcd.clear();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print(message);
}

void setColor(int red, int green, int blue) {
  analogWrite(redPin, red);
  analogWrite(greenPin, green);
  analogWrite(bluePin, blue);
}

void setTone(int freq) {
  tone(buzzer, freq, 300);
}

void setTone(int freq, int duration) {
  tone(buzzer, freq, duration);
}
```