Smart Home System for Elderly Solitudinarians Using Arduino

Justin Jaewoo Park

Saint Paul Preparatory Seoul

Abstract

Passengers' mobility platform Conatus's "half taxi" has been revived for the first time in 40 years. The city expected that the legislation of shared services would solve transportation problems such as difficulties in riding and increase passenger convenience, but limitations were found in the commercialization stage. The key issue is that the percentage of passengers is too low, the taxi fare is too low, or the above conditions are met. To address this, we propose a more efficient system in this paper. CTP solves the problem of the underworld. This ensures a 70% chance of route intersection sharers within 1 km by allowing users to walk reasonable distances. To ensure a higher probability of matching by adopting the Skyline query, the system recommends several taxi choices through the skyline based on several factors. Group Skyline method is adopted to calculate the optimal passenger group. Finally, we adopted a why-not query to motivate drivers and suggest appropriate guidelines for improving performance. In the result section, Adopting CTP helps the matching during the time complexity isn't increasing compared to the original algorithm. The paper concludes with suggestions in the aspect of real-time query and privacy.

Introduction

The arrival of the 21st century coincided with the rise of women's authority and their increased social activity, communication advancement, urbanization, and the increase of overall lifespan. As a result, the number of seniors over 65 and single-person homes has increased significantly. In 2019, approximately 9% of the world's population, or 703 million people, were 65 or older [1]. The senior population will reach 1.5 billion by 2050, according to projections [1]. In 2018, more than 40% of households in Northern Europe and 28.1% of families in the United States were one-person households. Nearly 60% of homes in Stockholm, Sweden, and London, UK were occupied by one person [3]. This development, together with the global growth in the pace of

aging, contributed to the emergence of countless elderly homes all over the world [4]. 16% of people aged 60 and older live alone, including 28% of Europeans and 26% of North Americans [5]. According to an estimate by the Korean Department of Statistics on the elderly population in 2021, 16.5% of Korea's population will be over 65 years of age, making Korea an aging society. This number is expected to reach 20.3% by 2025, making the country super-aged. 23.7% of all households in Korea are elderly households, and 35.1% of these are homes where the elderly live alone [6].

In terms of housing, health, and social life, the everyday lives of elderly persons who live alone (hence referred to as elderly solitudinarians) are less comfortable than their counterparts who live with others. 68.5% of older solitudinarians make less than 500,000 KRW each month, and 32.6% have more than one health-related restriction. The majority of them suffer from arthritis and hypertension. 27.1% of their homes have components that require repair. Over one-third of senior solitudinarians are socially neighbors. Only 18.7% of the population engages in social activities, and only 3% have hobbies like fishing or art [7].

Due to their economic challenges and lack of social engagement, it is usual for senior solitudinarians to die alone, and they tend to have shorter lifespans than other elderly people who live with others [7]. Smart homes are one solution to the aforementioned concerns of aging solitary individuals. Smart homes are automated, integrated electronic systems. Connected home electronics to an IoT (Internet of Things) network are able to communicate with one another for automation and remote access and control [8]. Smart homes have garnered attention for their potential to improve the daily lives of the elderly. Smart homes collect vital information about their users and analyze it to detect emergency situations, ensuring the health and safety of elderly individuals who live alone. Temperature and brightness controls can help to create a comfortable environment. Smart homes are modern, energy-efficient, and allow for customization, a crucial feature for aiding the elderly with disabilities.

Existing smart homes are already being used to aid elderly individuals living alone. The city of Daegu allocated 30 smart home apartments for their use in 2020[9]. Its emphasis on solitary death and safety accident prevention garnered favorable reviews, encouraging the municipality to give additional housing and services [10]. SKT's smart home service NUGU Smart Home currently collaborates with multiple builders, home networks, service providers, automobiles, and home appliances to offer customized services, family care services, safety services, and savings services for household appliances, heating devices, safety devices, lighting devices, and AI voice recognition devices [11]. Through Samsung's SmartThings app, Samsung Electronics users monitor and control its goods such as system air conditioners, built-in home

appliances, and kitchen and household hardware at a glance [12]. LG Electronics products can be remotely controlled via the LG ThinQ app, and LG U+ offers a variety of smart home monthly services for controlling home cameras and smart displays [13].

AI speakers are one of the most used methods for controlling smart homes. This is due to the fact that information can be obtained verbally and intelligent homes may be controlled without needing to know difficult operation procedures. Kakao mini, Naver Clova Friends, SKT NUGU, Google Nest, and KT GIGA Genie are sold in Korea [14-18]. In addition to controlling home appliances, AI-powered speakers connect to the Internet and function as virtual assistants. The utilization of AI speakers is convenient, however, due to poor voice recognition technology, commands are sometimes misconstrued or not recognized [19]. Moreover, the AI has a hard time understanding the user's command if the command structure is not followed; hence, virtual assistants installed on smartphones have replaced the purchase of AI speakers [20-22]. To create an effective smart home, it is necessary to move into a home with smart home technology already installed or to purchase new home products with IoT capabilities. It is nearly impossible for old, financially insecure, and technologically disadvantaged solitudinarians to use complex IoT devices. Using an Arduino board and sensors, this paper has designed a smart home system that is affordable and comfortable for senior individuals who live alone.

Body

To construct a perfect smart home facility for seniors living alone, various smart sensors were installed in a model home.

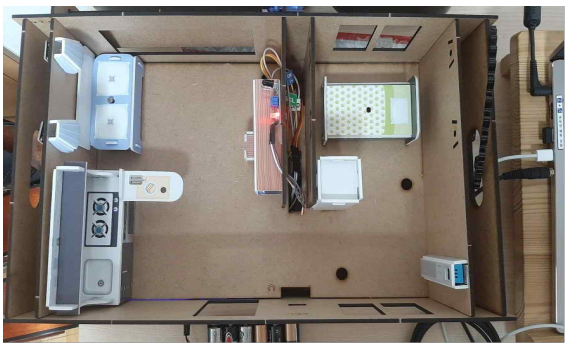
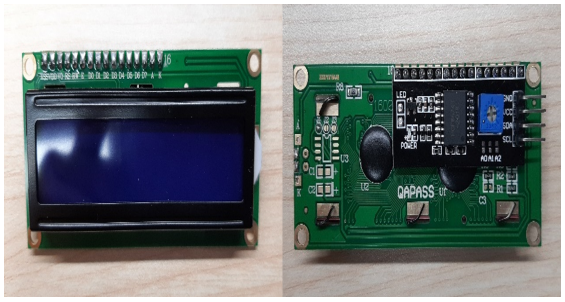


Figure 1. Model house of an elderly solitudinarian

As a microcontroller, Orange Board BLE was utilized in the smart home. Orange Board is a domestically made board compatible with the Arduino Uno Board and can be linked to the smartphone app nrf Toolbox through low-power Bluetooth (BLE) and Bluetooth 5.0. Arduino [23]. Orange Board is built on the Arduino system, a microcontroller board ecosystem that includes an open-source microcontroller and a development tool. Additional modules, such as sensors, can be added and removed indefinitely for a modest cost. Arduino is simple enough to be used by anyone without prior knowledge of hardware or software. As Arduino is based on open source, it offers the benefit of utilizing programs, libraries, and third-party sensors created by Arduino users around the globe. Arduino

boards may connect to Ethernet, WiFi, and SD cards via a variety of board shields. Using the C-based Arduino IDE (Integrated Development Environment), sensors and elements attached to an Arduino microcontroller are operated[24]. The smart home controls LCD modules, servo motors, and RGB LED modules using infrared obstacle detection, temperature and humidity, illuminance, piezo buzzers, and Bluetooth connectivity.

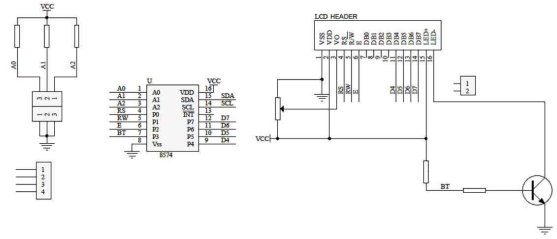
1. LCD Display



(a) (b)

Figure 2. LCD I2C module

(a) LCD I2C Screen (b) LCD I2C Backpack



Reference circuit diagram of the I2C-to-LCD piggy-back board.

Figure 3. LCD I2C Datasheet

The LCD I2C module is used to display the temperature and humidity sensor's readings as well as the status of all output devices. It takes the place of television in real-life homes.

The I2C adapter adds 16-pin LCD module wiring and variable resistance for screen brightness adjustment, making it easier to operate than typical LCD modules. Turning the variable resistance coupled to the module can adjust the luminance of the LCD's displayed characters. Four data-exchange pins, VCC, GND, SDA, and SCL, are linked to Arduino. The SDA receives information to be displayed on the screen as a data line from Arduino, while the SCL functions as the module's clock with a clock pin and receives time-related information from Arduino. The SDA and SCL pins should all be connected to the A0-5 pins of the analog data Arduino. Before displaying text on the LCD, the connection status (I2C address) of the I2C must be established. Before writing the main code, the I2C Scanner Code must be executed to obtain the I2C address [25].

```
/ ------------------------------------------------------------------ /
// Arduino I2C Scanner
// Re-writed by Arbi Abdul Jabbaar
// Using Arduino IDE 1.8.7
// Using GY-87 module for the target
// Tested on 10 September 2019
// This sketch tests the standard 7-bit addresses
// Devices with higher bit address might not be seen properly.
/ ------------------------------------------------------------------ /

#include <Wire.h> //include Wire.h library

void setup()
{
  Wire.begin(); // Wire communication begin
  Serial.begin(9600); // The baudrate of Serial monitor is set in 9600
  while (!Serial); // Waiting for Serial Monitor
  Serial.println("\nI2C Scanner");
}

void loop()
{
  byte error, address; //variable for error and I2C address
  int nDevices;

  Serial.println("Scanning...");

  nDevices = 0;
  for (address = 1; address < 127; address++ )
  {
    // The i2c_scanner uses the return value of
    // the Write.endTransmisstion to see if
    // a device did acknowledge to the address.
    Wire.beginTransmission(address);

    error = Wire.endTransmission();

    if (error == 0)
    {
      Serial.print("I2C device found at address 0x");
      if (address < 16)
        Serial.print("0");
      Serial.print(address, HEX);
      Serial.println("  !");
      nDevices++;
    }
    else if (error == 4)
    {
      Serial.print("Unknown error at address 0x");
      if (address < 16)
        Serial.print("0");
      Serial.println(address, HEX);
    }
  }
  if (nDevices == 0)
    Serial.println("No I2C devices found\n");
  else
    Serial.println("done\n");

  delay(5000); // wait 5 seconds for the next I2C scan
}
```

Check the I2C address with the Serial Monitor and enter it together with the horizontal and vertical lines into the LiquidCrystal I2C lcd() function as shown in the code below to configure the LCD.

LiquidCrystal_I2C lcd(0x27, 16, 2);

In order to use the code to control the LCD, the LCD Library was first added as #include <LiquidCrystal_I2C.h>.

```
lcd.init(); // initialize the lcd
lcd.backlight(); //turns on backlight
lcd.setCursor(0, 0);         // move cursor to   (0, 0)
```

As described above, the LCD prepares to output a temperature and humidity sensor value.

The temperature and humidity sensor value was outputted to the LCD screen using the lcd.print() function. The LCD panel is cleared every 500 milliseconds using the millis() function, which monitors the amount of time since the execution of the code. The correct temperature and humidity information can be refreshed by wiping the screen and outputting a new value.

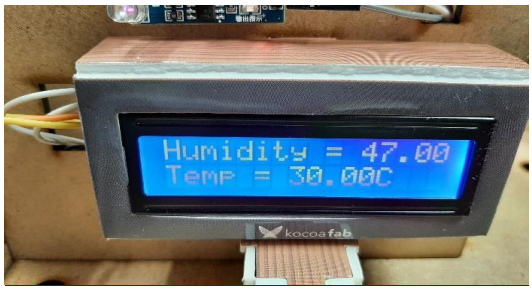


Figure 4. Humidity and Temperature Displayed on LCD

Instead of LCDs, more efficient and versatile touch screen displays can be utilized for monitoring in real-life homes. By displaying the status of all home appliances in a single location, it streamlines smart home control and makes it more intuitive through touch control.

2. Servo Motor

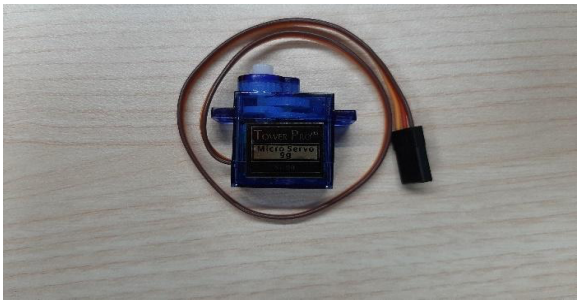
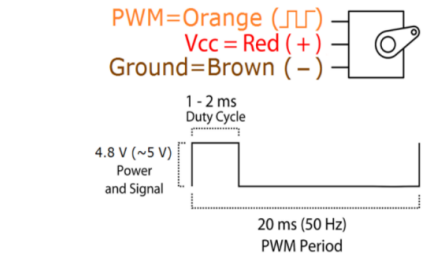


Figure 5. Mini Servo SG-90 SG90 DM147



Position "0" (1.5 ms pulse) is middle, "90" (~2 ms pulse) is all the way to the right, "-90" (~1 ms pulse) is all the way to the left.

Figure 6. Servo motor datasheet

The servo motor was used to control the model's blinds. Depending on the amount of light measured by the illuminance sensor, the blinds open or close automatically and without human input. It automatically delivers sunlight to elderly individuals who reside alone with a weak body and memory.

This model used the Arduino Mini Servo SG-90 SG90 DM147 because of its compatibility with Arduino products and its user-friendliness.

In the case of using the Arduino Library Servo.h, the servo motor can be rotated 90

degrees in both the left and right directions. It interfaces with the Arduino board by connecting its PWM, VCC, and GND pins. Connecting the PWM pin with the PWM digital pin on the Arduino board allows the user to issue a command through that pin. Specifying an angle from the reference line by a value between 0 and 180 with the myservo.write() function moves the motor.

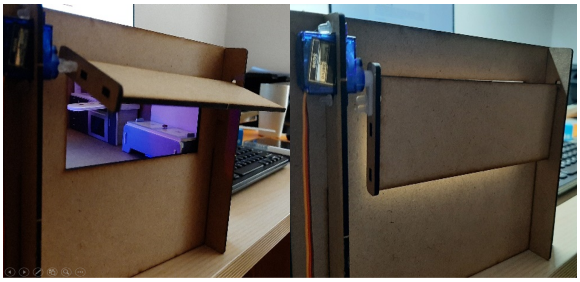
The motor is set up by the code

```
Servo myservo;  //Servo motor setup
```

and then in the code

```
int notes[] = {523, 1046, 2093};
int cdsValue = analogRead(cdsPin);

int blind = 0;

if (myservo.read()==0) {
  blind = 0;                //Sets current blind position
}

else if (myservo.read()==100)  {
  blind = 1;
}

if (cdsValue > 500 && blind == 1)  {
  myservo.write(0);
  for (int i = 0; i < 3; i++) {
  tone(piezoPin, notes[i], 200);
  delay(200);
  }                         //Closes blinds
}

else if (cdsValue < 500 && blind == 0)  {
  myservo.write(100);
  for (int i = 3; i >= 0; i--) {
  tone(piezoPin, notes[i], 200);
  delay(200);
}                           //Opens blinds
```

the angle of the servo motor is read using the myservo.read() function. When the angle is 0, closed (0) is stored and when the angle is 100, open (1) is stored in the variable blind. When the illuminance sensor detects darkness, it checks whether the blind is already open (blind = 1) and runs myservo.write(100); . If the sensor detects brightness, it checks whether the blind is currently closed (blind = 0) and runs myservo.write(0); Adjusting the servo motor's angle based on whether it is light or dark helps save power consumption, and checking the brightness before starting the code avoids the servo motor from continuously adjusting its angle.



(a) (b)

Figure 7. Blinds' movement based on brightness

(a) dark environment (b) bright environment

DC motors, which are rotatable and more powerful than servo motors, can be utilized in blinds and windows in real-life situations to allow them to open and close in response to the brightness of the light or as defined by the user. DC motors, unlike servo motors, can rotate a full 360 degrees, enabling curtains to be rolled up and down and windows to be opened and closed in conjunction with gears or hinges, allowing not only sunshine but also ventilation without the resident's personal control.

3. Piezo Buzzer

Figure 8. Piezo Buzzer PKM13EPY-4002-B0

```
int notes[] = {523, 1046, 2093};
int cdsValue = analogRead(cdsPin);

int blind = 0;

if (myservo.read()==0) {
  blind = 0;                //Sets current blind position
}

else if (myservo.read()==100)  {
  blind = 1;
}

if (cdsValue > 500 && blind == 1)  {
  myservo.write(0);
  for (int i = 0; i < 3; i++) {
  tone(piezoPin, notes[i], 200);
  delay(200);
  }                         //Closes blinds
}

else if (cdsValue < 500 && blind == 0)  {
  myservo.write(100);
  for (int i = 3; i >= 0; i--) {
  tone(piezoPin, notes[i], 200);
  delay(200);
}                           //Opens blinds
```

Notes 523 Hz (C5), 1046 Hz (C6), and 2093 Hz (C7) are stored in the integer array Array notes[], then tone (piezoPin, notes[i], 200); output i in the order of 0, 1, 2, 523 Hz, 2093 Hz, and close i in the order of 1, 03, 1046 Hz with a for-loop.

In the actual home, the piezo buzzer would be replaced with a conventional speaker system that can provide numerous voices and sound effects in addition to electrical beeps. It can be used as an alarm, a smart home sound effect, or a notification for the elderly with poor hearing and memory. A smart assistant provided by the user's smartphone manufacturer can be utilized to operate the home using only the voice when connected to the smartphone.

4. Infrared Sensor

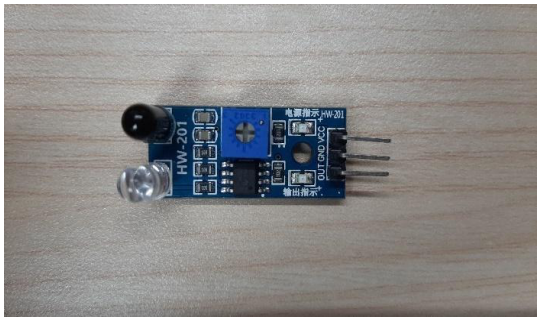


Figure 10. Infrared Sensor Module

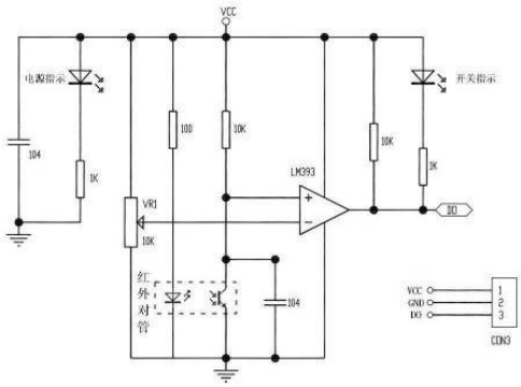


Figure 11. Infrared Sensor Module datasheet

To ascertain the location of the user, an infrared obstacle detection sensor is installed at a spot where the user is frequently present, such as in front of the television. With the user's location information, home appliances can be turned on or off based on it, alerting elderly people with impaired memories who live alone if they leave an electronic application or gas stove on when they leave home.

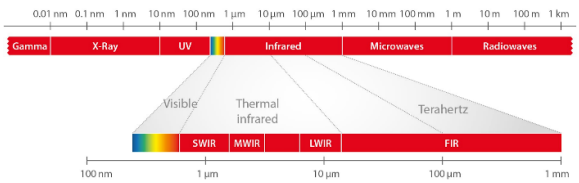


Figure 12. Wavelength Band - Infrared exists between 0.75㎛ and 1,000㎛

An infrared sensor generates infrared rays outside the red spectrum using LEDs and scans infrared light reflected on objects with phototransistors to discern the presence or absence of obstacles. Infrared is not visible to the human eye, thus it does not interfere with

daily activity. However, because black objects may not reflect infrared rays, they may not be recognized as obstacles.

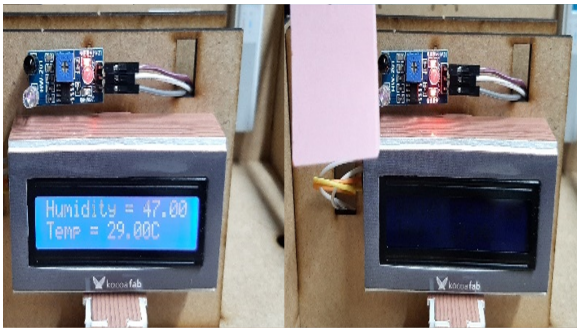
Among GND, VCC, OUT, and EN, the 3 pins excluding EN are connected to Arduino. Using the analogRead() function, the OUT pin of the infrared sensor can be read to output a value between 0 and 1023. The potentiometer on the sensor module can be used to adjust the sensitivity of the sensor. For obstacle detection, the smart home with the infrared sensor utilized an Arduino Infrared (IR) transmission and reception sensor module.

```
pinMode(IRPin, INPUT);  //IR Sensor Input setup
```

Setup the IR sensor to input with the above code, then with the code

```
int IRValue = analogRead(IRPin);

if (IRValue < 1000) {
  lcd.noBacklight();
}

else if (IRValue > 1000) {
  lcd.backlight();
}
```

the variable IRValue is designated as the infrared sensor value. If there was an obstacle in front of the sensor, the value would be less than 1000, so the LED on the LCD is turned off, and if there was no obstacle, the value would be more than 1000, so the LED is switched back on. If the numbers are inverted, the screen will automatically turn off if there is nothing in front of it.



(a) **(b)**

Figure 13. LCD backlight turns on or off based on obstacles
(a) No obstacles (b) Obstacles present

In reality, ultrasonic sensors with comparable functionalities but greater measuring distances (up to 4-5 m) are employed to compensate for the narrow detection ranges of infrared sensors (2-30 cm). In addition, if the user remains immobile for an extended period of time or if the sensors detect a fall, the smart home will request help and call the user's caretaker. Additionally, ultrasonic sensors can be set at easily accessible locations, such as windows, to alert users and police of home intrusions.

5. Temperature and Humidity Sensor

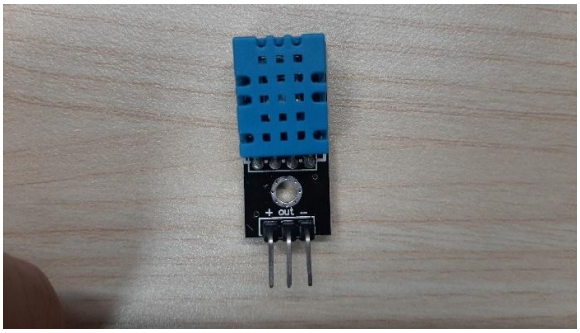


Figure 14. KY-015 DHT11 Temperature and Humidity Sensor Module

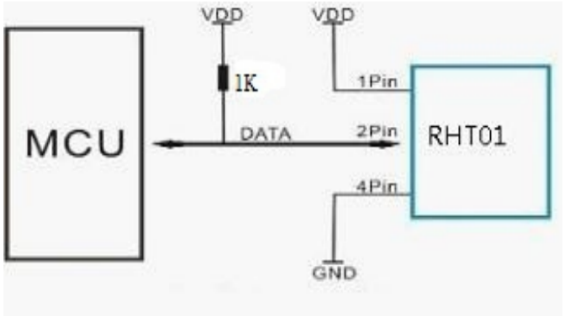


Figure 15. Temperature and Humidity Sensor datasheet

Temperature and humidity sensors monitor the surrounding environment's temperature and humidity, and can therefore be used to determine how comfortable a home is. Temperature and humidity are exhibited on an LCD display. Connecting with air conditioners or humidifiers to automatically adjust the home environment might reduce pain and prevent diseases and accidents for sensitivity-prone elderly solitudinarians.

The temperature and humidity sensor used in the smart home is a KY-015 DHT11 digital temperature and relative humidity sensor module, which combines a capacitive humidity sensor whose resistance changes according to humidity and a thermistor whose resistance changes according to temperature. The Arduino library dht.h is needed to output each value in units of RH (Relative Humidity) relative humidity and °C.

The sensor is used by connecting 3 pins of VCC, GND, and DATA with the Arduino Board. Reading the DATA pin connected to a digital pin on the board using the DHT.read11() function, the DHT.humidity variable stores the relative humidity value, and the DHT.temperature variable saves the temperature value.

```
DHT.read11(dhtPin);
lcd.print("Humidity = ");
lcd.print(DHT.humidity);
lcd.print("%");
lcd.setCursor(0,1);
lcd.print("Temp = ");
lcd.print(DHT.temperature);
lcd.print("C   ");

unsigned long currentMillis = millis();
if (currentMillis==500)  {
  lcd.clear();    //resets screen & result
  currentMillis = 0;
}
```

The lcd.print() function causes each variable to be output to the LCD screen with each respective name.

In actual homes, air conditioners, boilers, humidifiers, and dehumidifiers can be adjusted based on the temperature and humidity sensor readings to maintain optimal temperature and humidity, and anomalous variations in data can be identified to prevent tragedies. Additionally, a gas detection sensor is installed to assure the user's safety.

6. Illuminance Sensor

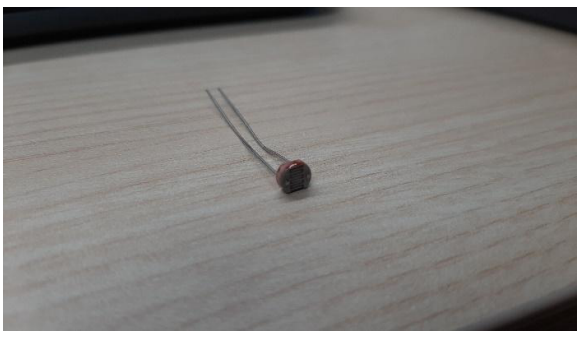


Figure 16. Illuminance Sensor

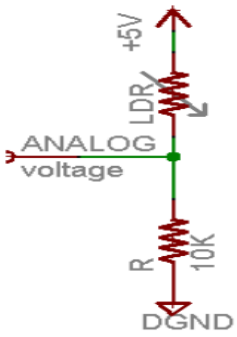


Figure 17. Illuminance Sensor datasheet

An illuminance sensor, commonly known as a light meter or lux meter, detects the brightness of the home and enables the automatic adjustment of light-handling elements, such as blinds and light bulbs, to meet the appropriate brightness for the environment. The illuminance sensor is a component whose brightness resistance value varies based on the brightness of its surroundings. In a range between about 1 kΩ and 50 kΩ, the brighter the light, the smaller the resistance value, and the darker the resistance, the larger it becomes. Adjusting the brightness of the home conserves energy and improves the optical wellness of the elderly who live alone.

The illuminance sensor quantifies brightness in lux. The lower the lux, the darker the environment, whereas the greater the lux, the brighter the surroundings. Lux can go from 0 to 130,000. 0 to 1000 lux can be detected using Arduino illuminance sensors, which are especially sensitive to 500 (green) to 700 (red) nm light.

```
int notes[] = {523, 1046, 2093};
int cdsValue = analogRead(cdsPin);

int blind = 0;

if (myservo.read()==0) {
  blind = 0;                //Sets current blind position
}

else if (myservo.read()==100)  {
  blind = 1;
}

if (cdsValue > 500 && blind == 1)  {
  myservo.write(0);
  for (int i = 0; i < 3; i++) {
  tone(piezoPin, notes[i], 200);
  delay(200);
  }                         //Closes blinds
}

else if (cdsValue < 500 && blind == 0)  {
  myservo.write(100);
  for (int i = 3; i >= 0; i--) {
  tone(piezoPin, notes[i], 200);
  delay(200);
}                           //Opens blinds
```

The cdsPin is read as with the analogRead() function and its value is stored in the variable cdsValue. The smart home model is programmed to open the blinds by moving the servo motor when it was less than 500 lux, and to close the blinds when it was more than 500 lux.

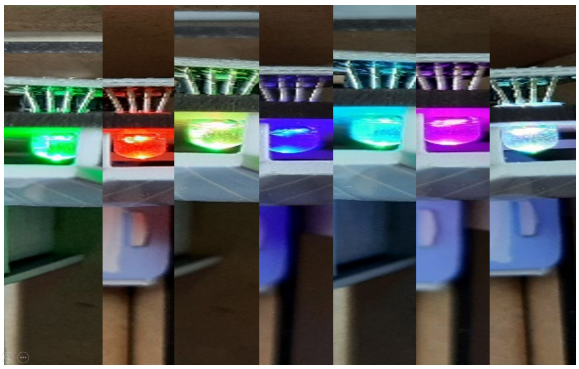
The sensor is utilized by plugging it into a breadboard and connecting it to the analog pin and GND of the Arduino board. In this model, the Arduino illuminance sensor photocell cds photo register is used to produce analog values based on the intensity of light at the moment. Even in actual homes, illumination sensors can be applied to operate the blinds and lighting optimally.

7. RGB LED

The RGB (Red, Green, and Blue) LED illuminates the smart home. In contrast to conventional lighting, RGB LEDs may combine red, green, and blue LEDs to produce magenta,

yellow, turquoise, and white based on human input. LEDs use less power than standard incandescent lights and allow the user to alter the intensity of RGB by adjusting the temperature of light.

LED Red, LED Green, LED Blue, and GND are the four pins, and each color LED is connected to a digital pin of Arduino to turn on or off the red, green, and blue parts of the LED.



(a) (b) (c) (d) (e) (f) (g)

Figure 18. RGB LED color lights

(a) Green (b) Red (c) Yellow – red and green (d) Blue (e) Turquoise – green and blue (f) Magenta – blue and red (g) White – red, blue, and green

```
// Array that stores the LED's current state. 0 if OFF, 1 if ON
byte sensor[] = {0, 0, 0, 0, 0, 0, 0};
```

```
//LED Output setup
pinMode(led,OUTPUT);
pinMode(ledR1,OUTPUT);
pinMode(ledG1,OUTPUT);
pinMode(ledB1,OUTPUT);
pinMode(ledR2,OUTPUT);
pinMode(ledG2, OUTPUT);
pinMode(ledB2, OUTPUT);
```

The sensor [] array in the code above stores the status of each of the three color pins of the RGB LEDs (one LED representing the air conditioner uses only blue).

```
void setToggle(int PIN, byte& stat) {
  if (stat == 0) {
    digitalWrite(PIN, HIGH);
    stat = 1;
  }
  else {
    digitalWrite(PIN, LOW);
    stat = 0;
  }
}
```

```
if (BTSerial.available()) {
  char input = BTSerial.read();

  switch (input) {
    case 'a':                          //RGB LED Red
      setToggle(ledR1, sensor[0]);
      break;

    case 'b':                          //RGB LED Green
      setToggle(ledG1, sensor[1]);
      break;

    case 'c':                          //RGB LED Blue
      setToggle(ledB1, sensor[2]);
      break;

    case 'd':                          //RGB LED1 Red
      setToggle(ledR2, sensor[3]);
      break;

    case 'e':                          //RGB LED1 Green
      setToggle(ledG2, sensor[4]);
      break;

    case 'f':                          //RGB LED1 Blue
      setToggle(ledB2, sensor[5]);
      break;

    case 'g':                          //aircon LED
      setToggle(led, sensor[6]);
      break;
  }
}
```

A new function called setToggle (int PIN, byte& stat) was created to write over the value of the sensor [] array as 0 and 1 as 0, when certain alphabetic characters were inputted via a Bluetooth signal.

RGB LEDs are suitable for uses in the real life because of their efficiency in comparison to incandescent bulbs and their ability to change the color of the light. In addition, a web server enables the user to monitor and control the

lights from outside the home, where Bluetooth signals cannot reach, through the Internet.

8. Bluetooth Connectivity

This smart home system is operated automatically based on sensor data or manually using a Bluetooth app on a smartphone. The automatic system does not require user input, yet the RGB LEDs are directly controlled via Bluetooth signals using a dedicated mobile app. Orange Board BLE, the Arduino Board utilized for this system, is equipped with Bluetooth 5.0 and BLE modules and can connect to a smartphone via the smartphone application nRF Toolbox[26]. Adding the Arduino Library SoftwareSerial.h adds the functions required for Bluetooth communication between the Orange Board and nRF Toolbox.

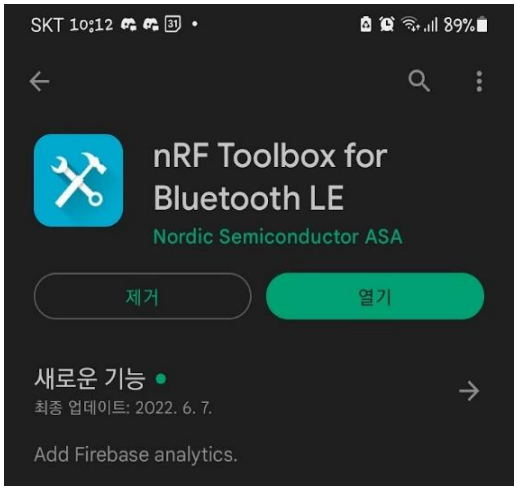


Figure 19. The nRF Toolbox for Bluetooth LE app on Google Play Store

The Orange Board BLE board sets up the Bluetooth module by assigning the connection to serial ports 4 and 5, which are the Bluetooth connection ports for the board, and putting it into the BTS serial() function.

```
void setToggle(int PIN, byte& stat) {
  if (stat == 0) {
    digitalWrite(PIN, HIGH);
    stat = 1;
  }
  else {
    digitalWrite(PIN, LOW);
    stat = 0;
  }
}
```

```
if (BTSerial.available()) {
  char input = BTSerial.read();

  switch (input) {
    case 'a':                          //RGB LED Red
      setToggle(ledR1, sensor[0]);
      break;

    case 'b':                          //RGB LED Green
      setToggle(ledG1, sensor[1]);
      break;

    case 'c':                          //RGB LED Blue
      setToggle(ledB1, sensor[2]);
      break;

    case 'd':                          //RGB LED1 Red
      setToggle(ledR2, sensor[3]);
      break;

    case 'e':                          //RGB LED1 Green
      setToggle(ledG2, sensor[4]);
      break;

    case 'f':                          //RGB LED1 Blue
      setToggle(ledB2, sensor[5]);
      break;

    case 'g':                          //aircon LED
      setToggle(led, sensor[6]);
      break;
  }
}
```

The code above receives user input with the function BTSserial.available() and reads the received input with the function BTSserial.read().

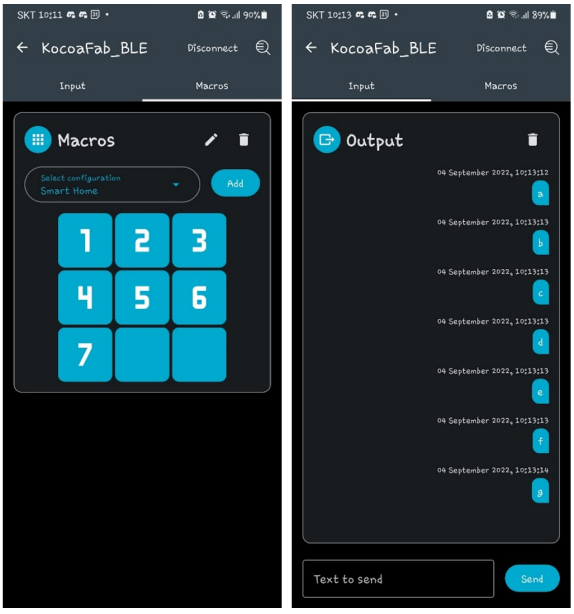


Figure 20. Inputs on the nFT Toolbox for the app

9. Remote Access and Security

However, there aren't enough home appliances that can be controlled outside the house where the Bluetooth connection isn't available. To tackle this issue, it is possible to create a separate web server for monitoring and controlling home environments via the Internet. Due of the size of an average home, Arduino sensors cannot be directly linked to the Arduino board if smart homes are integrated into real households. Consequently, the Arduino Wi-Fi connection module enables the sensor and main board to share data. It adds safeguards to prevent external cyberattacks. Security is maintained by specifying a security message on each sensor and eliminating any devices that do not respond to the security message prompt.

Conclusion

The number of elderly solitudinarians in Korea has been and continues to rise, and their living conditions and surroundings are inferior to those of other seniors. The deteriorating health of the elderly who live alone makes it difficult for them to operate electronic gadgets, such as home appliances and smartphones, that enhance their quality of life. This paper proposes a smart home to alleviate the daily inconveniences of living alone for the elderly.

Based on the Arduino ecosystem, a smart home model was created at a lower cost than existing smart home systems. In addition, Arduino-compatible sensors and output devices may be added in the future as necessary. By connecting an infrared obstacle detection sensor, a temperature and humidity sensor, and an illumination sensor to the Orange Board BLE, the LCD I2C module, the servo motor, and the piezo buzzer are controlled and managed automatically based on the sensor value. The RGB LED can be directly controlled by the user via a Bluetooth-connected smartphone app. Depending on the user's location, the sensors above can be used to adjust the blinds, alarms, and lighting in the residences of elderly solitudinarians, as well as switch off or on LCD screens that display temperature and humidity.

For practical scenarios, this smart home system should be equipped with emergency detection, anti-intruder security, remote control via a web server, voice recognition, and gas detection sensor functions managed by a smart hub to ameliorate the daily inconveniences of lone seniors. With smarter homes, it is projected that

elderly solitudinarians will engage in more social activities and experience fewer accidents at home.

References

1. United Nations, Department of Economic and Social affairs, Population Division, "World Population Ageing 2019: Highlights," https://www.un.org/en/development/desa/population/publications/pdf/ageing/WorldPopulationAgeing2019-Highlights.pdf, 2019
2. Our World in Data, "The rise of living alone: how one-person households are becoming increasingly common around the world," https://ourworldindata.org/living-alone, 2019
3. K. D. M. Snell, "The rise of living alone and loneliness in history," Social History, Vol. 42, pp. 2-28, 2017
4. Klinenberg, Eric, "Going Solo: The Extraordinary Rise and Surprising Appeal of Living Alone," Duckworth, 2014
5. Pew Research Center, "Older people are more likely to live alone in the U.S. than elsewhere in the world," https://www.pewresearch.org/fact-tank/2020/03/10/older-people-are-more-likely-to-live-alone-in-the-u-s-than-elsewhere-in-the-world/, 2020
6. 통계청, "2021 고령자 통계," https://www.kostat.go.kr/portal/korea/kor_nw/1/1/index.board?bmode=read&aSeq=403253, 2021 (In Korean)
7. Jeong-Kwan Lee, Jun-Hyeon Kim, "서울특별시 독거노인 실태에 관한 연구," Seoul Studies, 14(3), pp. 191-211, 2013 (In Korean)
8. R.L. Robles, T.H. Kim, "Applications, Systems and Methods in Smart Home Technology: A Review," International Journal of Advanced Science and Technology, Vol. 15, pp. 37-48, 2010.
9. 연합뉴스, "대구시 독거노인 대상 스마트홈 서비스 실증," https://www.yna.co.kr/view/AKR20200923153300053, 2020 (In Korean)
10. 매일뉴스, "스마트도시 대구, 스마트홈 기술로 고독사 사전 예방 눈길," http://news.imaeil.com/page/view/2021121216121012125, 2021 (In Korean)
11. SK Telecom, NUGU Smart Home, https://www.sktsmarthome.com/
12. Samsung, SmartThings Home Solution, https://www.samsung.com/sec/business/b-iot-solutions/smart-home-solution-iot
13. LG, LG ThingQ, https://www.lge.co.kr/lg-thinq
14. Kakao i, Kakao Mini, https://kakao.ai/product/kakaomini
15. Naver, CLOVA Friends, https://clova.ai/ko/ko-product-friends.html
16. SK Telecom, NUGU nemo, https://www.nugu.co.kr/device/nugu_nemo
17. KT, GiGA Genie, https://gigagenie.kt.com/main.do

18. Google, Nest https://store.google.com/kr/?hl=ko
19. Dason Lee, Young June Sah, Sangwon Lee, "Improving Usability Perception of Error-Prone AI Speakers: Elaborated Feedback Mitigates Negative Consequences of Errors," International Journal of Human-Computer Interaction, Vol. 35, pp. 1645-1652, 2019
20. Google, Google Assistant, https://assistant.google.com/
21. Apple, Siri, https://www.apple.com/siri/
22. Samsung, Bixby, https://www.samsung.com/global/galaxy/what-is/bixby/
23. Kocoafab, Orange Board BLE, https://kocoafab.cc/product/orangeble
24. Arduino, https://www.arduino.cc/
25. Arduino, I2C Scanner Arduino Code, https://create.arduino.cc/projecthub/abdularbi17/how-to-scan-i2c-address-in-arduino-eaadda
26. Nordic Semiconductor, nRF Toolbox https://www.nordicsemi.com/Products/Development-tools/nRF-Toolbox