

창의 SW 기초 설계
최종 보고서

구해줘 고맙미

탁상에 올려놓을 수 있는
1인가구를 위한
위급상황 감지 경보기

6조

목차

01

팀 소개

02

제작 동기

03

플로우차트

04

제작도면

05

제품설명 및 제작비용

06

아두이노 코딩

07

동작 시연(영상 포함)

08

추후 개선 요구 사항

09

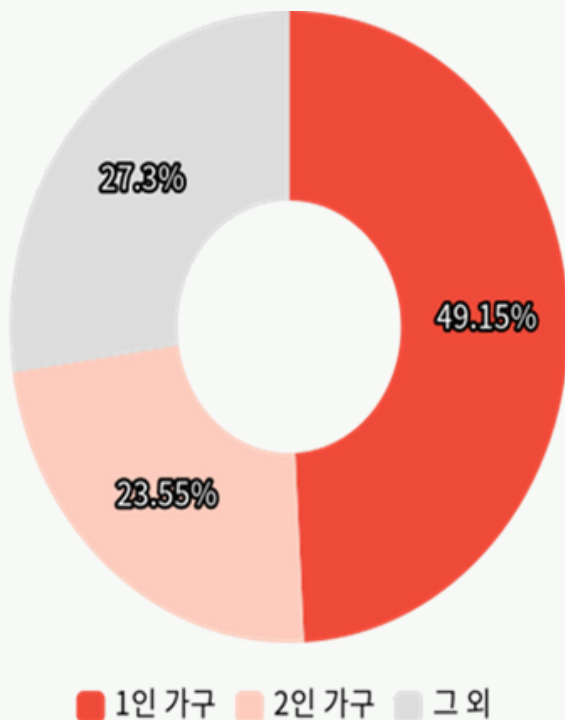
기대효과

팀 소개

<창의SW기초설계 무인 002 분반 6조 ‘나이는 숫자에 불과하다’>

구성원	역할
박병준	조장 및 CAD
황찬미	자료제작 및 하드웨어
변재형	발표 및 SW
최규민	자료조사 및 하드웨어

제작동기



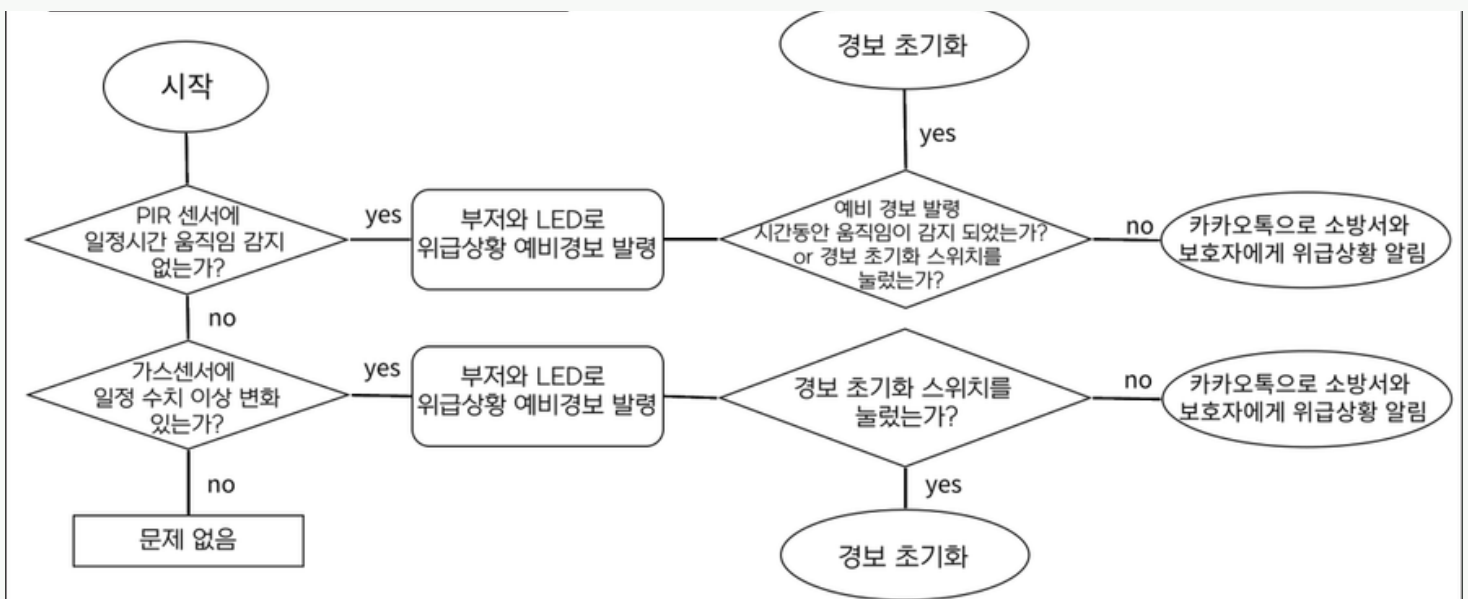
광진구 1인 가구 비율



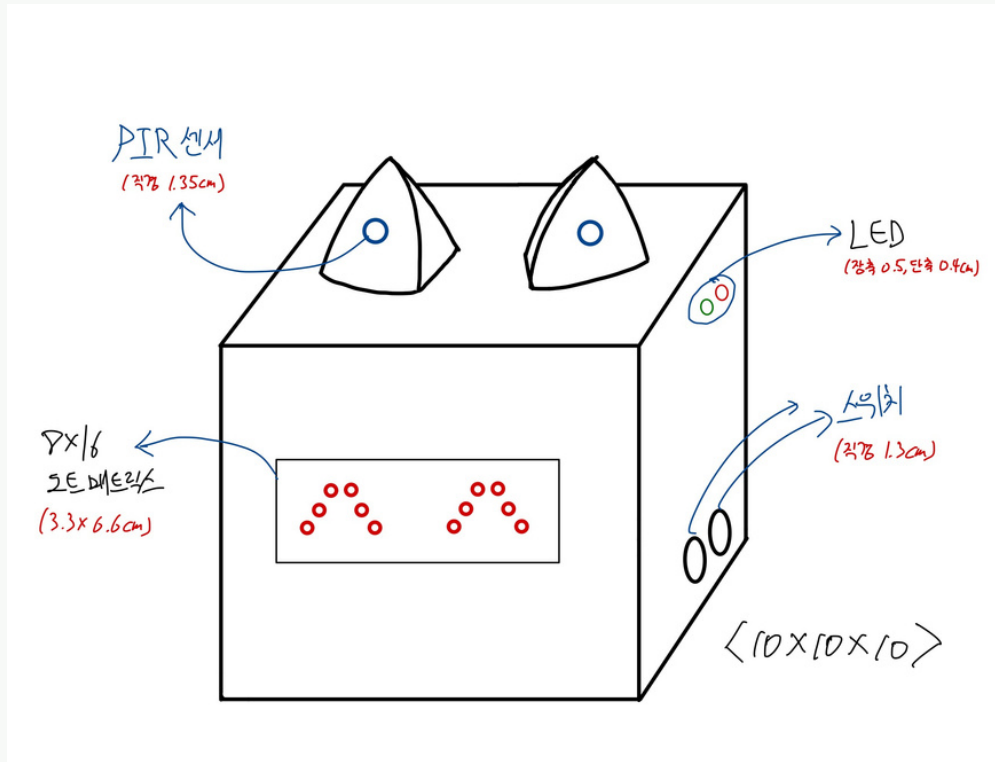
프로젝트 필요성

“함께 사는 광진구”라는 프로젝트 주제에 맞게 광진구에 사는 장애인 고령층과 같은 취약계층의 삶의 질을 높이기 위해 안전, 교통, 건설, 환경 등 다양한 주제로 광진구 거주 취약계층에게 유용한 서비스에 대해서 조원들끼리 같이 고민하고 회의를 해봤다. 먼저 광진구의 특성을 조사해본 결과, 광진구의 1인 가구 비율이 약 49%로 전체 세대 중 가장 높은 비중을 차지하고 있었다. 인구주택 총조사에 따르면 1인 가구가 가장 불편함을 느끼는 부분은 위급상황 대처와 관련된 부분으로 통계자료에서 확인하였고, 위급상황 발생 시 신고자가 부재한다는 1인가구의 문제점 극복을 위해, 고독사 등 1인가구에서 발생 가능한 응급상황을 대처하고 예방하기에 수월한 ‘1인가구를 위한 응급상황 감지 및 경보기’를 제작하기로 회의를 통해 결정을 했다.

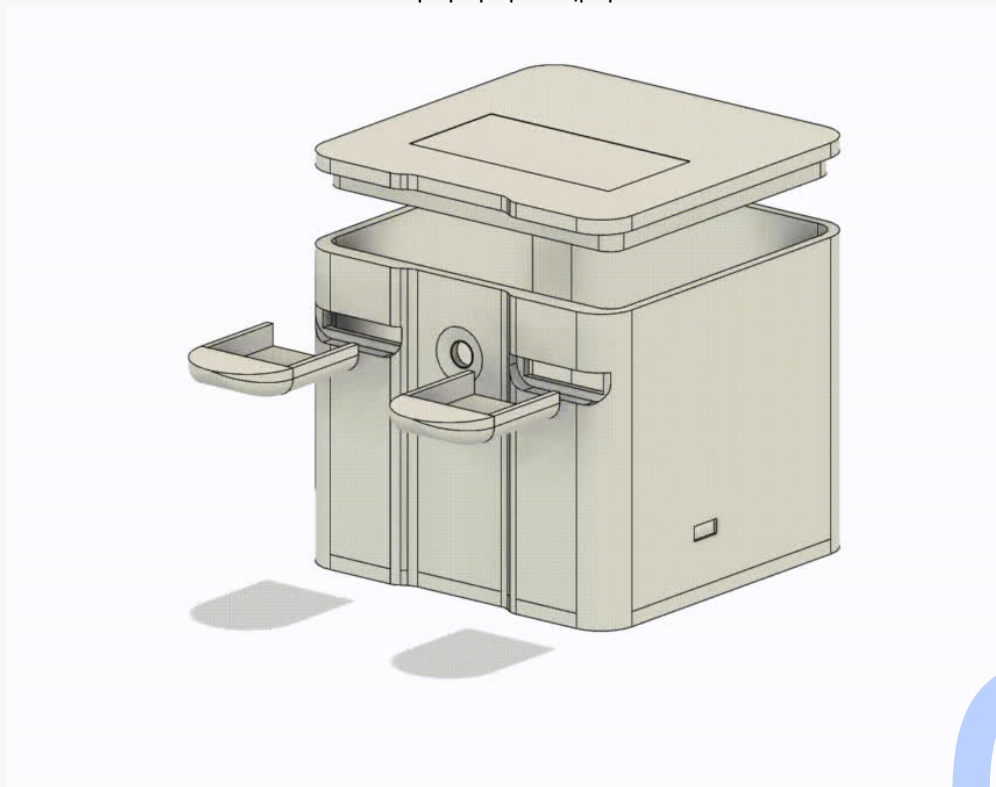
플로우 차트



제작도면



아이디어 스케치



3d 모델링

작품설명

제품은 10 X 10 X 10 의 정육면체로, 탁상에 올려놓고 사용이 가능한 고양이 형상의 소형가전 제품이다. 고양이 디자인을 활용해 양쪽귀에 PIR 센서를 장착 시켜 거주자의 움직임을 파악해 장시간 거주자의 움직임이 없을 경우 led와 부저를 통해 거주자에게 응급상황인지 아닌지에 대해 묻는다. 응급상황일 경우 카카오톡으로 근처 소방서 등에 알릴 수 있게 했고, 응급상황이 아닐 경우 거주자가 고양이를 쓰다듬거나 움직임을 보여 응급상황이 아님을 알릴 수 있다. 귀와 귀 사이에는 터치 센서를 넣어 쓰다듬으면 도트매트릭스(8 X 16)에 이모티콘을 이용해 표정을 나타내고, 카카오톡으로 알림을 보낼 수도 있다. 제품 뒷면에는 가스 센서를 부착해 응급상황을 감지하고 마찬가지로 카카오톡을 통하여 빠르게 대처할 수 있도록 했다.

제작비용

도트 매트릭스X2	1900원
ESP32 Dev Kit 듀얼 보드	5970원
MQ-2 가스센서모듈	1060원
PIR센서X2	9600원
점퍼케이블	2400원
캡 포함 푸시버튼	100원
3d 프린팅	15000원
총액	36110원

아두이노 코딩



아두이노 코딩

```
1  /*----- macro -----*/
2  // kakao api 토큰
3  #define APP_KEY "1fafcdf5ce97c4ccd2ec0faffbe8b20d"
4  #define REFRESH_TOKEN "G-Vk3R74XUeF2qmB6WqtItulAcgbjppwKQvGYa6ZCiolkAAAAYikMAMZ"
5
6  // dot matrix 핀 번호
7  #define dataIn 18
8  #define cs 19
9  #define clk 21
10 #define gas 35
11
12 /*----- Libraries -----*/
13 #include <WiFi.h>
14 #include <HttpClient.h>
15 #include <base64.h>
16 #include "LedControl.h"
17
18 /*----- WIFI ID & PW -----*/
19 const char *ssid = "testpilot";
20 const char *password = "smarcle2017";
21
22 // kakao api web connect
23 const char *host = "https://kapi.kakao.com/v2/api/talk/memo/default/send";
24 String access_token;ss
```

아두이노 코딩

```
26  /*----- pin allocation -----*/
27  int green = 5;
28  int red = 4;
29  int buz = 27;
30  int pir1 = 25;
31  int pir2 = 26;
32  int switchon = 32;
33
34  /*----- global -----*/
35  unsigned long past = 0;    // 시간 측정에 사용할 변수
36  float gasValue;    // 가스센서 입력값 저장 변수
37
38  /*----- Dot Matrix control -----*/
39  LedControl lc = LedControl(dataIn, clk, cs, 2);
40
41  // emotions
42  byte smile[8] =
43  {
44      B00000000,
45      B00000000,
46      B00011000,
47      B00100100,
48      B01000010,
49      B00000000,
50      B00000000,
51      B00000000
52  };
```

아두이노 코딩

```
164  /*----- setup func -----*/
165  void setup()
166  {
167    // Serial settings
168    Serial.begin(115200);
169    Serial.println(F("Hello, ESP32!\n"));
170
171    // Pin Mode settings
172    pinMode(green, OUTPUT);
173    pinMode(red, OUTPUT);
174    pinMode(switchon, INPUT);
175    pinMode(buz, OUTPUT);
176    pinMode(pir1, INPUT);
177    pinMode(pir2, INPUT);
178
179    // LED dot matrix settings
180    lc.shutdown(0, false);
181    lc.shutdown(1, false);
182    lc.setIntensity(0, 15);
183    lc.setIntensity(1, 15);
184    lc.clearDisplay(0);
185    lc.clearDisplay(1);
```

아두이노 코딩

```
187 // WIFI settings
188 WiFi.begin(ssid, password);
189 Serial.println("Connecting");
190 while (WiFi.status() != WL_CONNECTED)
191 {
192     Serial.print(".");
193     delay(250);
194 }
195 Serial.println("");
196 Serial.print("Connected to WiFi network with IP Address: ");
197 Serial.println(WiFi.localIP());
198 }
```

```
200 /*----- main func -----*/
201 void loop()
202 {
203     int i = 0;
204     int j;
205     int cnt = 0;
206     int p1 = digitalRead(pir1);
207     int p2 = digitalRead(pir2);
208     int pb = digitalRead(switchon);
209     int touchValue = touchRead(33);
210     unsigned long emer;    // 시간 측정할 변수
211     gasValue = analogRead(gas);
```

아두이노 코딩

```
213  /*----- touch sensing -----*/
214  if (touchValue < 70){    // 터치 감지 시
215      digitalWrite(red, LOW);
216      digitalWrite(buz, LOW);
217
218      while(i < 3){        // cute eyes
219          for (j = 0; j <8; j++){
220              lc.setRow(0, j, right_cute1[j]);
221              lc.setRow(1, j, left_cute1[j]);
222          }
223          digitalWrite(green, HIGH);
224          delay(300);
225
226          for (j = 0; j <8; j++) {
227              lc.setRow(0, j, right_cute2[j]);
228              lc.setRow(1, j, left_cute2[j]);
229          }
230          digitalWrite(green, LOW);
231          delay(300);
232          i++;
233      }
```

아두이노 코딩

```
235 // heart emotion
236 for (j = 0; j <8; j++){
237     lc.setRow(0, j, heart[j]);
238     lc.setRow(1, j, heart[j]);
239 }
240 digitalWrite(green, HIGH);
241
242 // send kakao message
243 if ( update_access_token() == true ) {
244     send_message3();
245 }
246 }
```

아두이노 코딩

```
248  /*----- gas sensing -----*/
249  if (gasValue > 300){    // 가스 감지 시
250      while(1){
251          digitalWrite(green, LOW);
252          digitalWrite(red, HIGH);
253          digitalWrite(buz, HIGH);
254
255          // angry face
256          for (j = 0; j <8; j++) {
257              lc.setRow(0, j, angry[j]);
258              lc.setRow(1, j, angry[j]);
259          }
260
261          // send kakao message
262          if ( update_access_token() == true ) {
263              if (cnt == 0) {
264                  send_message1();
265              }
266              cnt++;    // only once
267          }
268      }
269  }
```


아두이노 코딩

```
271  /*----- human motion not detected -----*/
272  if (p1 == LOW && p2 == LOW) {
273      digitalWrite(green, LOW);
274      digitalWrite(red, HIGH);
275      digitalWrite(buz, HIGH);
276
277      emer = millis();    // millis() 함수 이용해 사후 시간 측정 시작
278
279      // surprised face
280      for (j = 0; j <8; j++) {
281          lc.setRow(0, j, open_eyes[j]);
282          lc.setRow(1, j, open_eyes[j]);
283      }
284
285      if (emer - past >= 5000){
286          // sad face
287          for (j = 0; j <8; j++) {
288              lc.setRow(0, j, sad[j]);
289              lc.setRow(1, j, sad[j]);
290          }
291
292          // send kakao message
293          if ( update_access_token() == true ) {
294              if (cnt == 0) {
295                  send_message2();
296              }
297              cnt++;
298          }
299      }
300  }
```

아두이노 코딩

```
302  /*----- human motion detected -----*/
303  else {
304      digitalWrite(green, HIGH);
305      digitalWrite(red, LOW);
306      digitalWrite(buz, LOW);
307      past = millis();
308      cnt = 0;
309
310      // default face
311      for (j = 0; j < 8; j++) {
312          lc.setRow(0, j, smile[j]);
313          lc.setRow(1, j, smile[j]);
314      }
315  }
```

아두이노 코딩

```
317 // kakao API
318 if ( Serial.available() )
319 {
320     int id = Serial.read();
321
322     if ( id == '1' )
323     {
324         send_message1();
325         send_message2();
326     }
327     // token 갱신하기 6시간 마다 갱신해야 함
328     if ( id == '2' )
329     {
330         if ( update_access_token() == true )
331         {
332             send_message1();
333             send_message2();
334             send_message3();
335         }
336     }
337 }
338 }
```

아두이노 코딩

```
340  /*----- Kakao API funcs -----*/
341  String urlencode(String str)  //
342  {
343      String encodedString = "";
344      char c;
345      char code0;
346      char code1;
347      char code2;
348      for (int i = 0; i < str.length(); i++) {
349          c = str.charAt(i);
350          if (c == ' ') {
351              encodedString += '+';
352          } else if (isalnum(c)) {
353              encodedString += c;
354          } else {
355              code1 = (c & 0xf) + '0';
356              if ((c & 0xf) > 9) {
357                  code1 = (c & 0xf) - 10 + 'A';
358              }
359              c = (c >> 4) & 0xf;
360              code0 = c + '0';
361              if (c > 9) {
362                  code0 = c - 10 + 'A';
363              }
364              code2 = '\\0';
365              encodedString += '%';
366              encodedString += code0;
367              encodedString += code1;
368              //encodedString+=code2;
369          }
370          yield();
371      }
372      return encodedString;
373  }
```

아두이노 코딩

```
375 /*----- kakao messages -----*/
376 // fire alert
377 void send_message1()
378 {
379     HTTPClient http;
380     if (!http.begin(host))
381     {
382         Serial.println("\nfailed to begin http\n");
383     }
384     http.addHeader("Authorization", "Bearer " + access_token); // <== 카카오 OPEN API의 Access Token을 입력하세요
385     http.addHeader("Content-Type", "application/x-www-form-urlencoded");
386
387     int http_code;
388
389
390     String data = "template_object={\"object_type\": \"text\", \"text\": \"거주지 내 화재 발생!!!\", \"link\": {\"web_url\": \"https://www.naver.com\", \"mobile_web_url\": \"https://www.naver.com\"}, \"button_title\": \"바로 확인\"}";
391     Serial.println(data);
392     http_code = http.sendRequest("POST", data);
393     Serial.print("HTTP Response code: ");
394     Serial.println(http_code);
395
396
397     String response;
398     if (http_code > 0)
399     {
400
401         response = http.getString();
402         Serial.println(response);
403     }
404     else
405     {
406     }
407     http.end();
408 }
```

아두이노 코딩

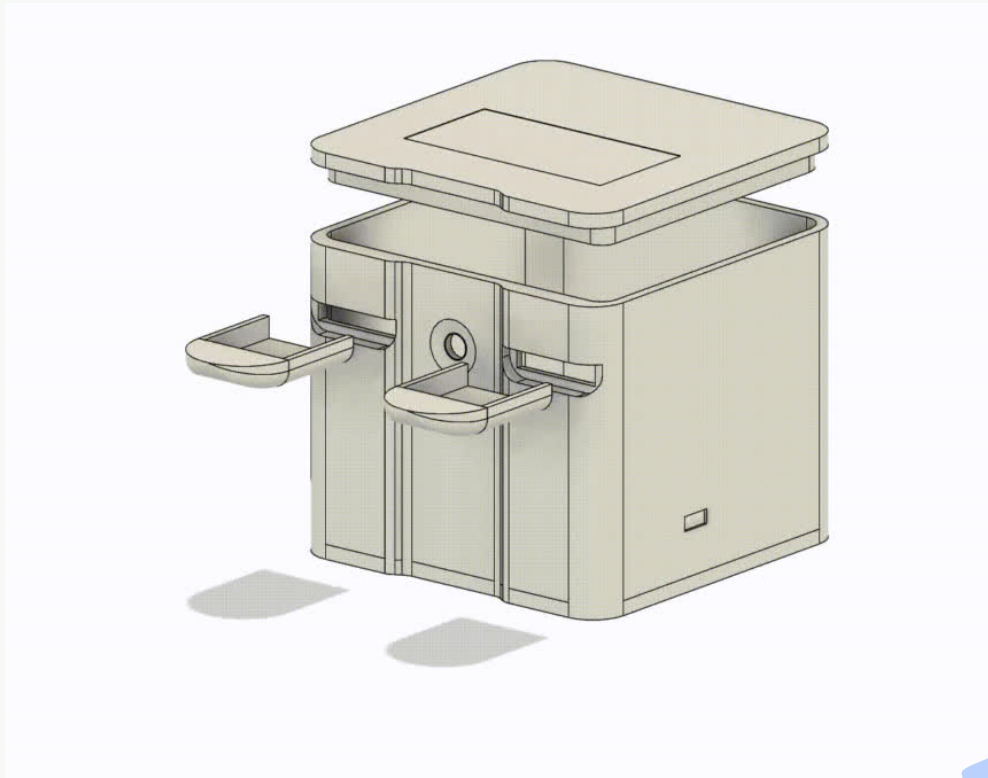
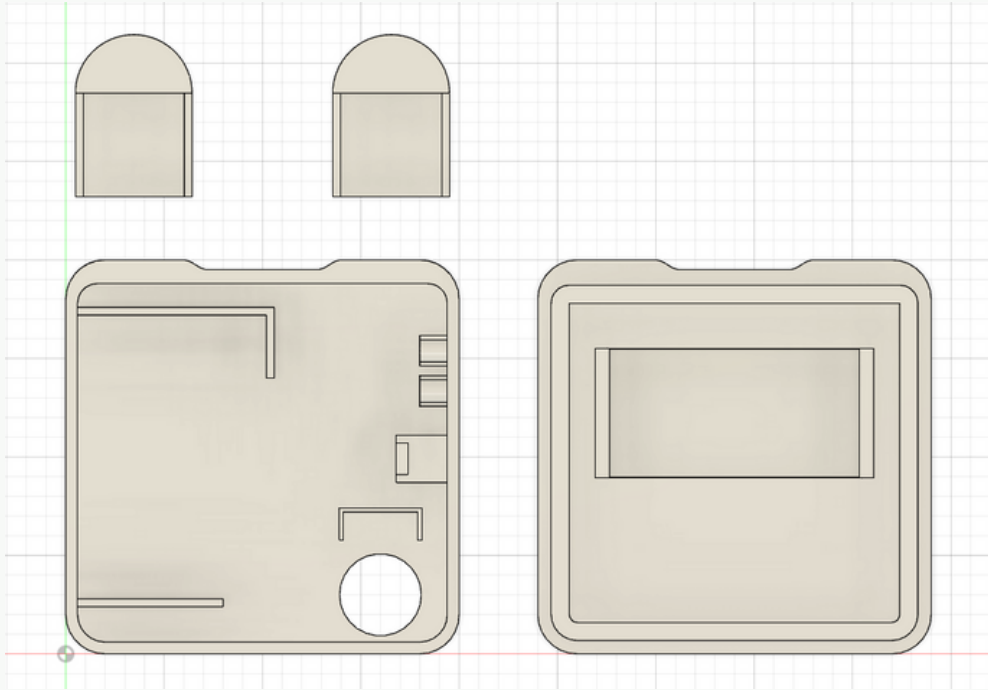
```
476 String extract_string(String str, String start_string, String end_string)
477 {
478     int index1 = str.indexOf(start_string) + start_string.length();
479     int index2 = str.indexOf(end_string, index1);
480     String value = str.substring(index1, index2);
481     return value;
482 }
```

```
484 bool update_access_token()
485 {
486     HTTPClient http;
487     String url = "https://kauth.kakao.com/oauth/token";
488     if (!http.begin(url))
489     {
490         Serial.println("\nfailed to begin http\n");
491     }
492     int http_code;
493
494     String client_id = String(APP_KEY);
495     String refresh_token = String(REFRESH_TOKEN);
496     String data = "grant_type=refresh_token&client_id=" + client_id + "&refresh_token=" + refresh_token;
497     Serial.println(data);
498     http_code = http.POST(data);
499     Serial.print("HTTP Response code: ");
500     Serial.println(http_code);
}
```

아두이노 코딩

```
502 String response;  
503 if (http_code > 0)  
504 {  
505  
506     response = http.getString();  
507     Serial.println(response);  
508     access_token = extract_string(response, "{\"access_token\":\"\", \"\"");  
509     http.end();  
510     return true;  
511 }  
512 else  
513 {  
514 }  
515 http.end();  
516 return false;  
517 }
```

CAD 설계



동작시연(영상 포함)



추후 개선 요구사항

첫번 째

GPS 모듈을 시도했으나 실내에서 작동하지 않은 점

두번 째

와이파이 정보, 거주지 등의 정보를 하드코딩을 통해 수정해야한다는 점. 웹 서버를 사용하려 하였으나, 구현 실패했다는 점.

기대효과

첫 번째

위급상황 발생 시 발견 시간을 감소시켜 1인가구의 안전을 도모할 수 있다.

두 번째

사고를 미리 방지하여 불필요한 사고가 안 생기도록 한다