

Auto Warehouse System

육군 17사단 정보보호병
상병 방준식

육군 20사단 통신병
병장 김민욱

차례

1. 개요
2. 개발 동기
 - 1) 문제 인식
 - 2) 해결 방안
3. 사업 전망
4. 개발 과정
 - 1) 개발 환경
 - 2) 설계
 - 3) 코드
5. 시연
6. 개선 방향

개요



문제 인식

오늘 안에 00창고에
있는 물건들 수량
파악 완료하도록!

오늘 들어온 보급품
보관할 공간이
어디 있나?

자물쇠도 없는
창고인데....
도둑 안 들겠지..?



문제 인식

움직임을 파악하여
경보를 울리는
방법 기능

보관중인 **물건의 양**을
자동으로 파악하면
편리할 것이다.

남은 공간을 알려주면
유용할 것이다.

온도에 따라 자동으로
냉난방기 가동

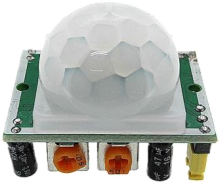
물건의 **이름**으로 보관된
위치를 **검색**하면 좋을
것이다.



해결 방안



거리 측정을 통해 물건의 양 파악
로 물건 위치 및 남은 공간 등 정보 표시



방범기능ON : 움직임 X->0 => 로 경보



온습도 측정 후 정상 범위 벗어나면



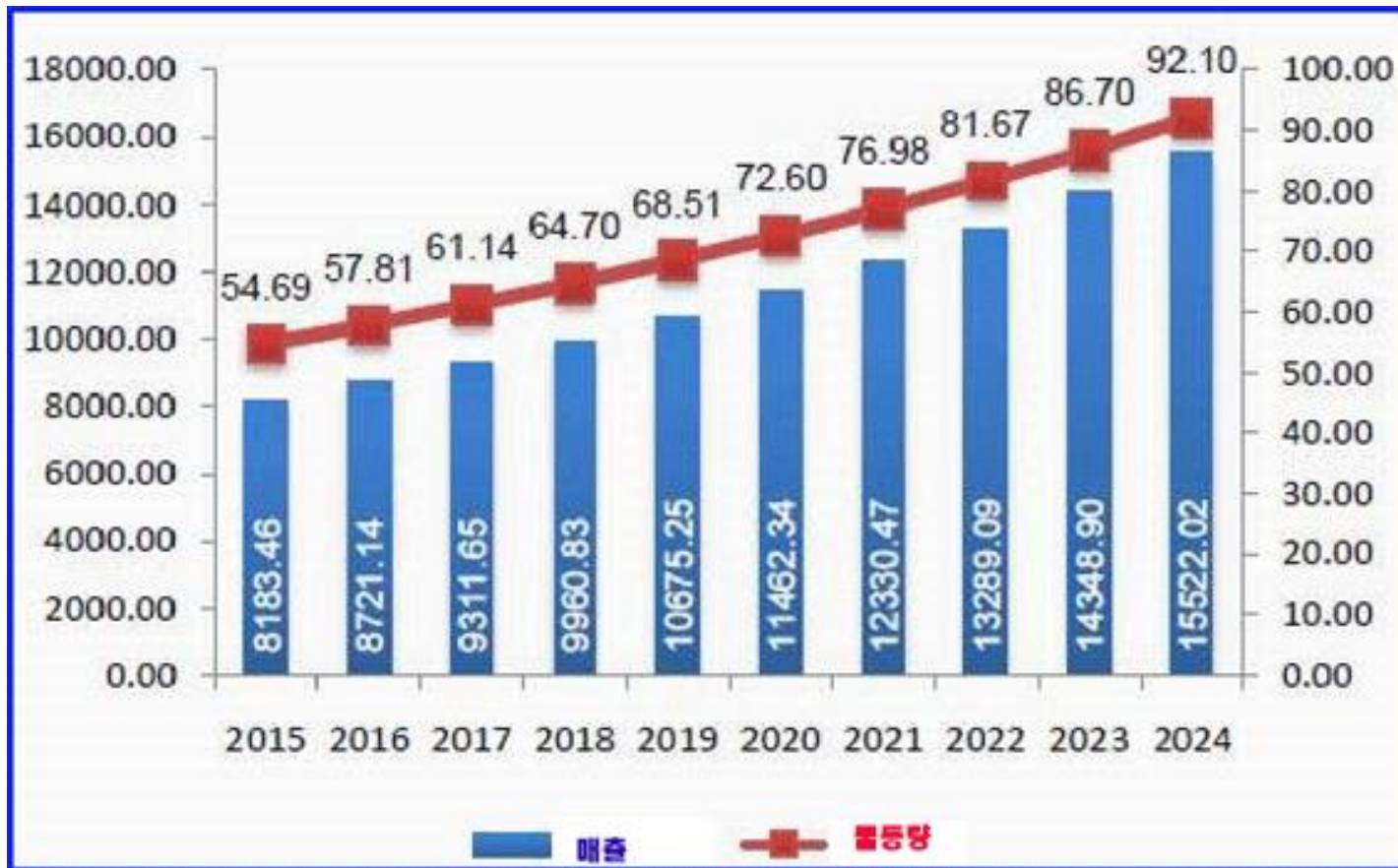
로 냉·난방기 또는 가습·제습기 가동



블루투스를 통해 사용자 스마트폰으로 정보 제공

사업 전망

세계 물류 시장의 성장



출처 : Transparency Market Research, Cargo News

사업 전망

최저 시급의 인상



사업 전망

군 내에서 활용

1 ~ 10종 창고

각 중대 창고



전산소모품
창고

탄약고

육군군수사 제2보급단

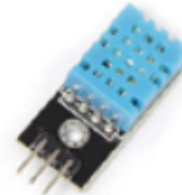
개발 환경



초음파 센서



PIR 모션 인식센서



온습도 센서



HC-06 블루투스



5V 릴레이 모듈



5V 능동부저

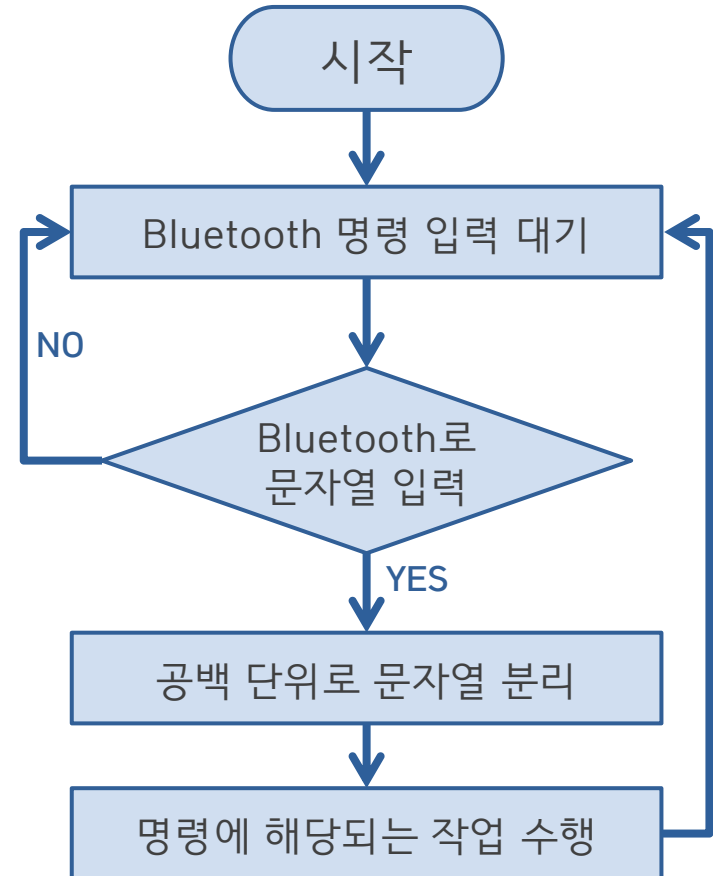
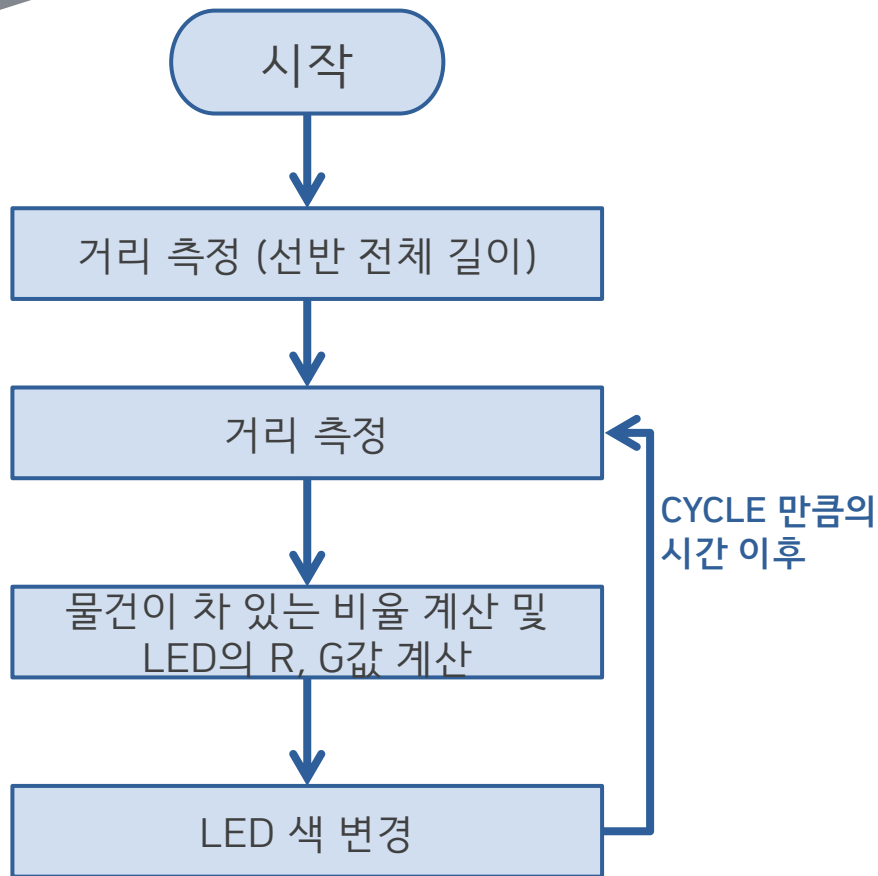


RGB LED

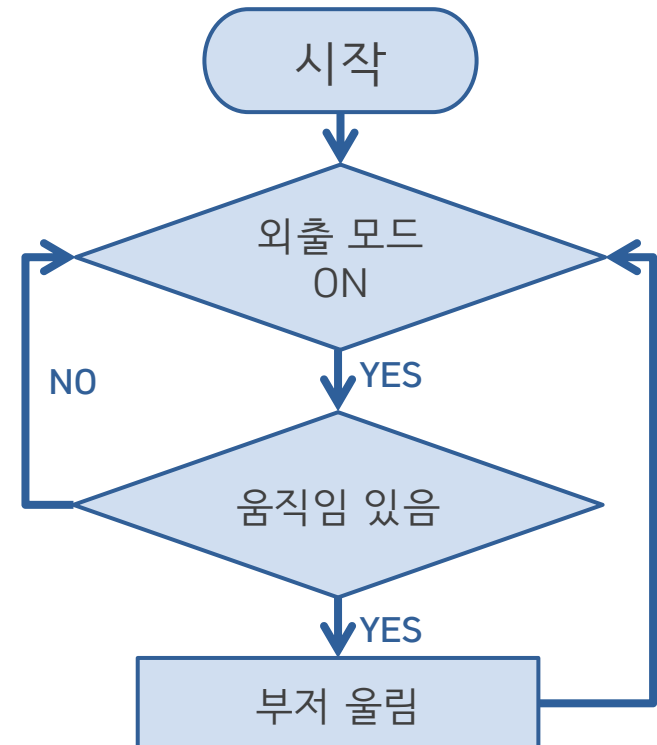
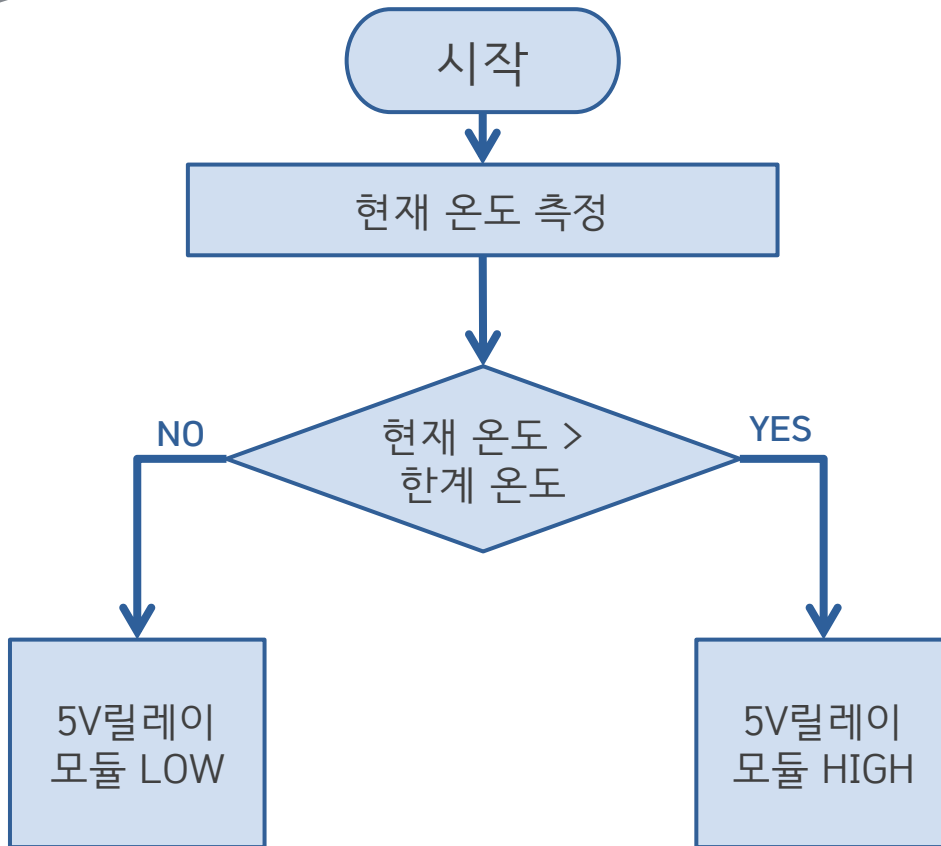


MIT
APP INVENTOR

설계

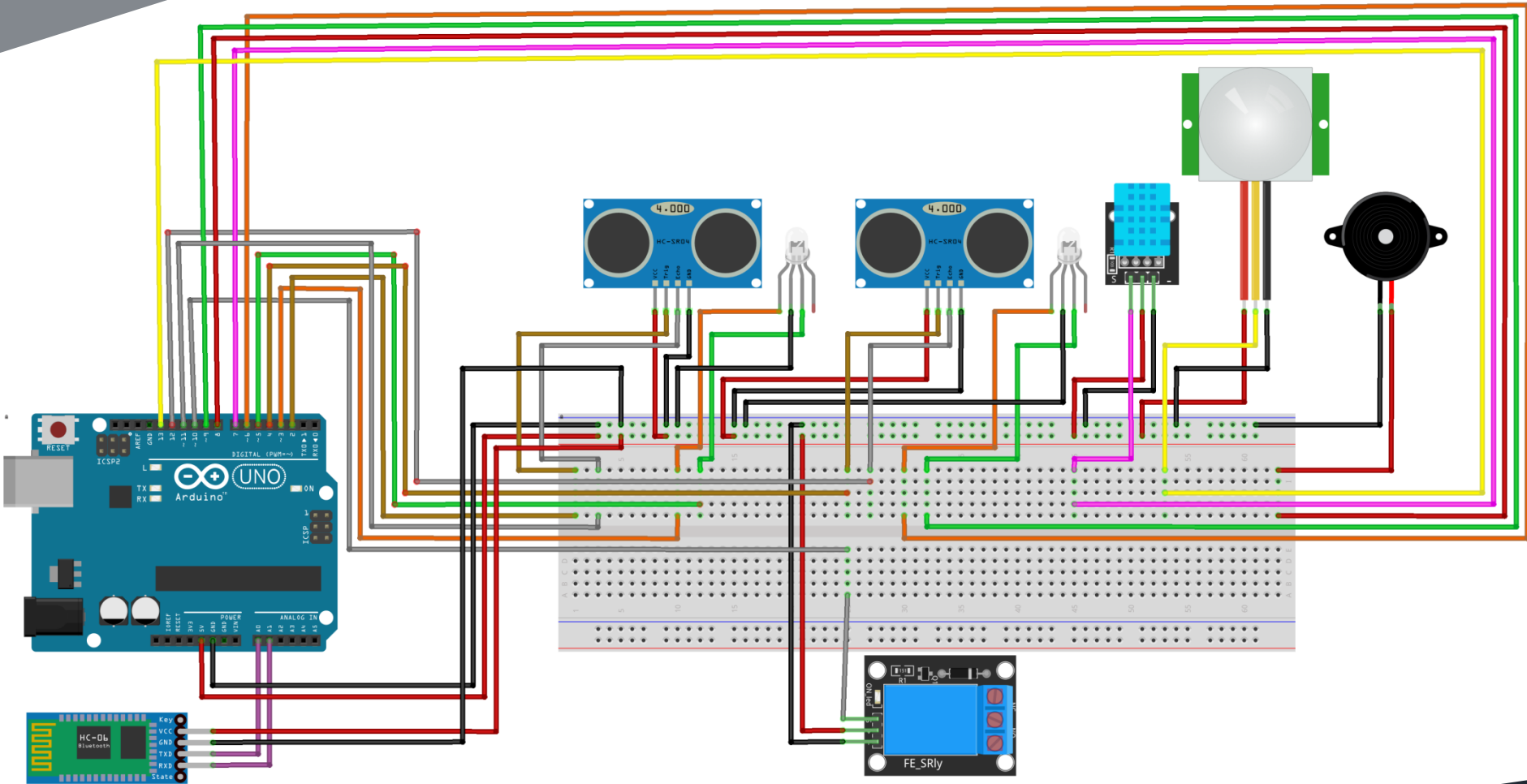


설계



설계

fritzing



코드

pin.h

struct_var.h

auto.h

make.h

command.h

AutoWarehouse
System.ino

```
1  #define DHTTYPE DHT11
2
3  // output pin
4  int dsensor_trig_pin[2] = {2,4};
5  int led_r_pin[2] = {3,6}; // pwm
6  int led_g_pin[2] = {5,9}; // pwm
7  int dht_pin = 7;
8  int beep_pin = 8;
9
10 // input pin
11 int relay_pin = 10;
12 int dsensor_echo_pin[2] = {11,12};
13 int pir_pin= 13;
14
15 // bluetooth pin
16 int rx_pin = 14;
17 int tx_pin = 15;
```

코드

pin.h

struct_var.h

auto.h

make.h

command.h

AutoWarehouse
System.ino

```
5 // struct
6 typedef struct LED_ {
7     int r_pin;
8     int g_pin;
9     int r;
10    int g;
11 }LED;
12
13 typedef struct Shelf {
14     // shelf info
15     int max_d; // 선반 길이
16     int now_d; // 현재 물건 길이
17     int size; // 물건 1개당 길이, 지정 안하면 기본값 0
18     char name[100]; // 물건 이름
19     LED *led;
20     // pin info
21     int trig_pin;
22     int echo_pin;
23 } SHELF;
```

코드

pin.h

struct_var.h

auto.h

make.h

command.h

AutoWarehouse
System.ino

```
26 // variable
27 int i, temperature, limit_temperature, moving, outting, cycle;
28 double ratio;
29 unsigned long pre_time = 0;
30 unsigned long cur_time = 0;
31 char tempc;
32 char temps[100];
33 SHELF *shelves[2];
34
35 SoftwareSerial bluetooth(rx_pin, tx_pin);
36 DHT dht(dht_pin, DHTTYPE);
37
38 void inputString_temps() {
39     for (i = 0; i < 100; i++) {
40         temps[i] = 0;
41     }
42     i = 0;
43     while(bluetooth.available()) {
44         tempc = bluetooth.read();
45         temps[i++] = tempc;
46         delay(10);
47     }
48 }
```

코드

pin.h

struct_var.h

auto.h

make.h

command.h

AutoWarehouse
System.ino

```
3 // auto led control
4 void setLED(LED *led, int r, int g) {
9 }
10
11 void lightControl(SHELF *shelf) {
18 }
19
20
21 // auto check
22 int check_distance(int trig_pin, int echo_pin) {
31 }
32
33 void auto_check_distance(SHELF **shelves, int num, int cycle) {
57 }
58
59
60 //auto working
61 void auto_temperature(int limit) {
68 }
69
70 SHELF* find_name(SHELF **shelves, int num, char *name) {
79 }
80
81 void show_info(SHELF *shelf) {
103 }
104
105 void findspace(Shelf **shelves, int num) {
126 }
127
128 void crime_prevention() {
139 }
```

```

33 void auto_check_distance(SHELF **shelves, int num, int cycle) {
34     int before_d, now_d;
35     cur_time = millis();
36     if(cur_time - pre_time >= cycle) {
37         pre_time = cur_time;
38         for (i = 0; i < num; i++) {
39             before_d = shelves[i]->now_d;
40             now_d = check_distance(shelves[i]->trig_pin,shelves[i]->echo_pin);
41             shelves[i]->now_d = now_d;
42             lightControl(shelves[i]);
43             if(before_d - now_d > 2){
44                 if(shelves[i]->size > 0){
45                     sprintf(temps, "%s %d개가 입고되었습니다. \n", shelves[i]->name, (int)((((double)(before_d - now_d) / shelves[i]->size) + 0.5)));
46                     bluetooth.write(temps);
47                 }
48             }
49             if(now_d - before_d > 2){
50                 if(shelves[i]->size > 0){
51                     sprintf(temps, "%s %d개가 출고되었습니다. \n", shelves[i]->name, (int)((((double)(now_d - before_d) / shelves[i]->size) + 0.5)));
52                     bluetooth.write(temps);
53                 }
54             }
55         }
56     }
57 }

```

millis()

아두이노가 켜진 이후 경과한 시간

일정 주기로 같은 작업을 반복해야 할 때

delay() -> 다른 작업 **X**

millis() -> 다른 작업 **O**

물건 개수 출력에서의 정확도 향상

반올림 구현

코드

pin.h

struct_var.h

auto.h

make.h

command.h

AutoWarehouse
System.ino

```
1  #include "auto.h"
2
3  // object make function
4  LED* makeLED(int r_pin, int g_pin, int r, int g) {
5      LED *led = (LED*)malloc(sizeof(LED));
6      led->r_pin = r_pin;
7      led->g_pin = g_pin;
8      led->r = r;
9      led->g = g;
10     return led;
11 }
12
13 SHELF* makeShelf(int trig_pin, int echo_pin, int r_pin, int g_pin) {
14     SHELF *shelf = (SHELF*)malloc(sizeof(SHELF));
15     LED *led = makeLED(r_pin, g_pin, 0, 0);
16     shelf->max_d = check_distance(trig_pin, echo_pin);
17     shelf->now_d = 0;
18     shelf->size = 0;
19     strcpy(shelf->name, "tempname");
20     shelf->led = led;
21     shelf->trig_pin = trig_pin;
22     shelf->echo_pin = echo_pin;
23     return shelf;
24 }
```

코드

pin.h

struct_var.h

auto.h

make.h

command.h

AutoWarehouse
System.ino

```
3 void commandProcessing(char *command, SHELF **shelf, int num) {
4     char opcode[10] = {0};
5     char value1[10] = {0};
6     char value2[10] = {0};
7     char value3[100] = {0};
8     char temp = 0;
9     int index = 0;
10    int index_op = 0;
11    int index_v1 = 0;
12    int index_v2 = 0;
13    int index_v3 = 0;
14    while(command[index] != ' ') {
15        opcode[index_op] = command[index];
16        index_op++;
17        index++;
18    }
19    while(command[index] != ' ') {
20        value1[index_v1] = command[index];
21        index_v1++;
22        index++;
23    }
24    while(command[index] != ' ') {
25        value2[index_v2] = command[index];
26        index_v2++;
27        index++;
28    }
29    while(command[index] != ' ') {
30        value3[index_v3] = command[index];
31        index_v3++;
32        index++;
33    }
34    if(strcmp(opcode, "SET") == 0) {
35        if(strcmp(value1, "NAME") == 0) {
36            if(strcmp(value2, "SIZE") == 0) {
37                if(strcmp(value3, "COLOR") == 0) {
38                    // ...
39                }
40            }
41        }
42        if(strcmp(value1, "NAME") == 0) {
43            if(strcmp(value2, "SIZE") == 0) {
44                if(strcmp(value3, "COLOR") == 0) {
45                    // ...
46                }
47            }
48        }
49        if(strcmp(opcode, "SET") == 0) {
50            // ...
51        }
52        else if(strcmp(opcode, "FIND") == 0) {
53            // ...
54        }
55        else if(strcmp(opcode, "OUTTING") == 0) {
56            // ...
57        }
58        else if(strcmp(opcode, "SHOW") == 0) {
59            // ...
60        }
61        else {
62            bluetooth.write("op code error");
63        }
64    }
65 }
```

코드

pin.h

struct_var.h

auto.h

make.h

command.h

AutoWarehouse
System.ino

```
1 #include"make.h"
2
3 void processingOP(const char* , const char* , SHELF** , char* , char* , char* , int* , int* , int* , const int );
4 void commandInput(const char* , SHELF** , char* , char* , char* , int* , int* , int* );
5 void commandSet(const char* , char* , char* , int* , int* );
6 void commandFind(const char* , char* , char* , int* , int* , const int );
7 void commandOutting(const char* );
8 void commandShow(const char* );
9
10 // 명령어 처리 메인
11 void commandProcessing(char *command, SHELF **shelves, const int num) {
12 }
13
14 // OPCODE 처리
15 void processingOP(const char *opcode, const char *command, SHELF **shelves, char *value1, char *value2, char *value3, int *index, int *index_v2, const int num) {
16 }
17
18 // 입력 명령 처리
19 void commandInput(const char *command, SHELF **shelves, char *value1, char *value2, char *value3, int *index, int *index_v2, const int num) {
20 }
21
22 // 설정 명령 처리
23 void commandSet(const char *command, char *value1, char *value2, int *index, int *index_v2){
24 }
25
26 // 찾기 명령 처리
27 void commandFind(const char *command, char *value1, char *value2, int *index, int *index_v2, const int num){
28 }
29
30 // 외출(방범) 명령 처리
31 void commandOutting(const char *value1){
32 }
33
34 // 현재 온도 표시 명령 처리
35 void commandShow(const char* value1){
36 }
```

코드

pin.h

struct_var.h

auto.h

make.h

command.h

AutoWarehouse
System.ino

```
1  #include "command.h"
2
3  void setup() {
4      for (i = 2; i < 10; i++) {
5          pinMode(i, OUTPUT);
6      }
7      for (i = 10; i < 14; i++) {
8          pinMode(i, INPUT);
9      }
10     Serial.begin(9600);
11     bluetooth.begin(9600);
12     dht.begin();
13     pre_time = millis();
14     limit_temperature = 30; // default
15     cycle = 1000; // default
16     outting = 0; // default
17     SHELF* shelf1 = makeShelf(dsensor_trig_pin[0], dsensor_echo_pin[0], led_r_pin[0], led_g_pin[0]);
18     SHELF* shelf2 = makeShelf(dsensor_trig_pin[1], dsensor_echo_pin[1], led_r_pin[1], led_g_pin[1]);
19     shelves[0] = shelf1;
20     shelves[1] = shelf2;
21 }
22
23 void loop() {
24     moving = digitalRead(pir_pin);
25     crime_prevention();
26     auto_temperature(limit_temperature);
27     if(outting == 0) auto_check_distance(shelves, 2, cycle);
28     inputString_temps();
29     if(temps[0] != 0) commandProcessing(temps, shelves, 2);
30 }
```

시연

물건 양 파악

물건 이름 및 길이
입력

찾기 기능

외출(방법) 기능

현재 온도 보기

자동 온도 조절

측정 주기 설정



시연

물건 양 파악

물건 이름 및 길이
입력



찾기 기능

외출(방법) 기능

현재 온도 보기

자동 온도 조절

측정 주기 설정

KT     LTE+ 75% 오전 1:21

Autowarehousesystem

나가기

shelves

이름 입력 : 전송

크기 입력 : 전송

연결되었습니다
shelf0

CLEAR

시연

물건 양 파악

물건 이름 및 길이
입력

찾기 기능

외출(방법) 기능

현재 온도 보기

자동 온도 조절

측정 주기 설정



시연

물건 양 파악

물건 이름 및 길이
입력

찾기 기능

외출(방법) 기능

현재 온도 보기

자동 온도 조절

측정 주기 설정

Autowarehousesystem

나가기

찾을 이름 :

라면

전송

가장 비어있는 선반

전송 완료

name = 라면

max = 24 cm

current amount = 10 cm, 41 % There is 1 라면 on this

Autowarehousesystem

나가기

찾을 이름 :

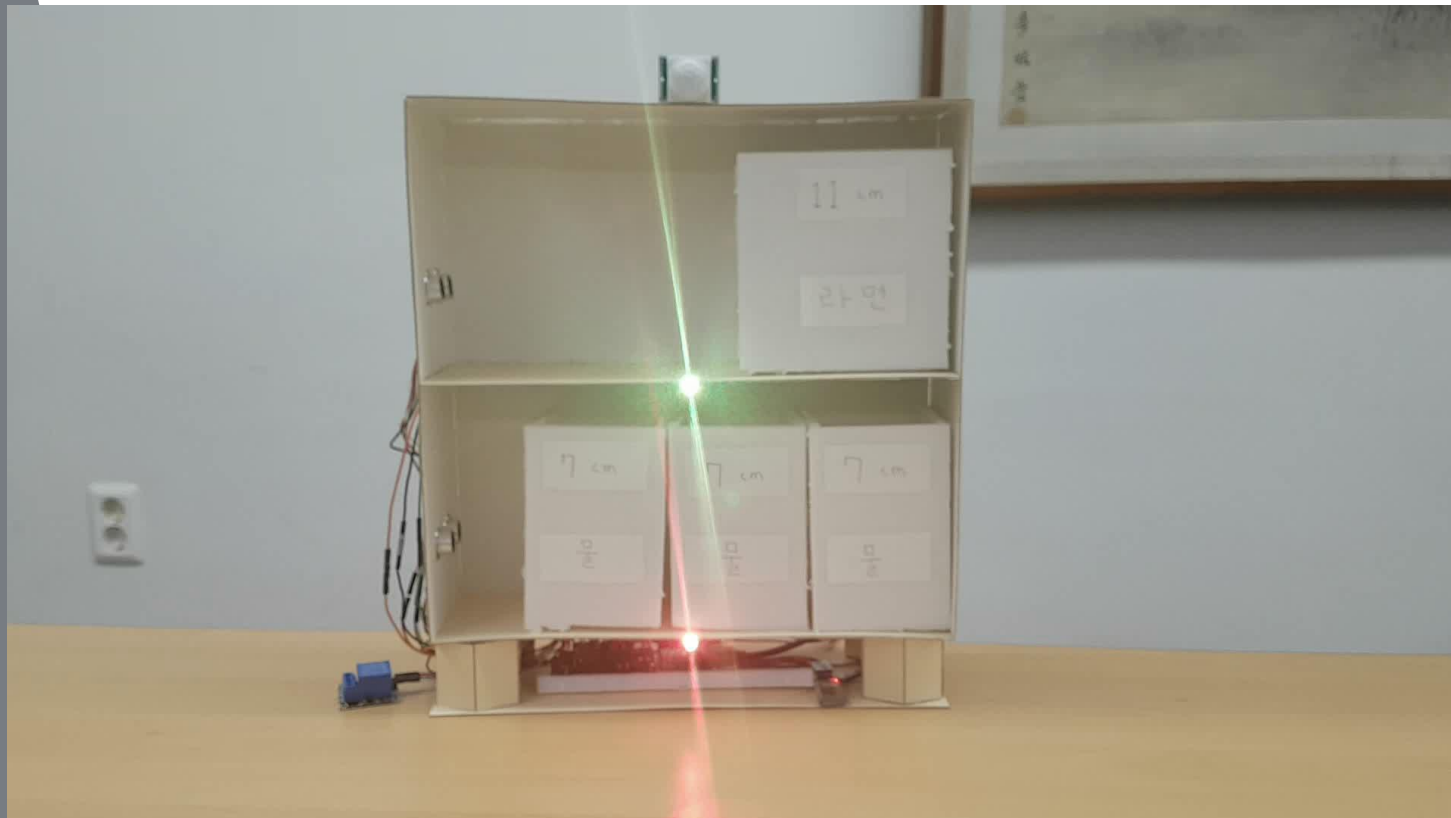
라면

전송

가장 비어있는 선반

전송 완료

가장 많은 빈 공간이 있는 곳은 라면 이(가) 보관된 칸으로 14 cm만큼 더 보관 가능합니다.



시연

물건 양 파악

물건 이름 및 길이
입력

찾기 기능

외출(방범) 기능

현재 온도 보기

자동 온도 조절

측정 주기 설정

Autowarehousesystem

나가기

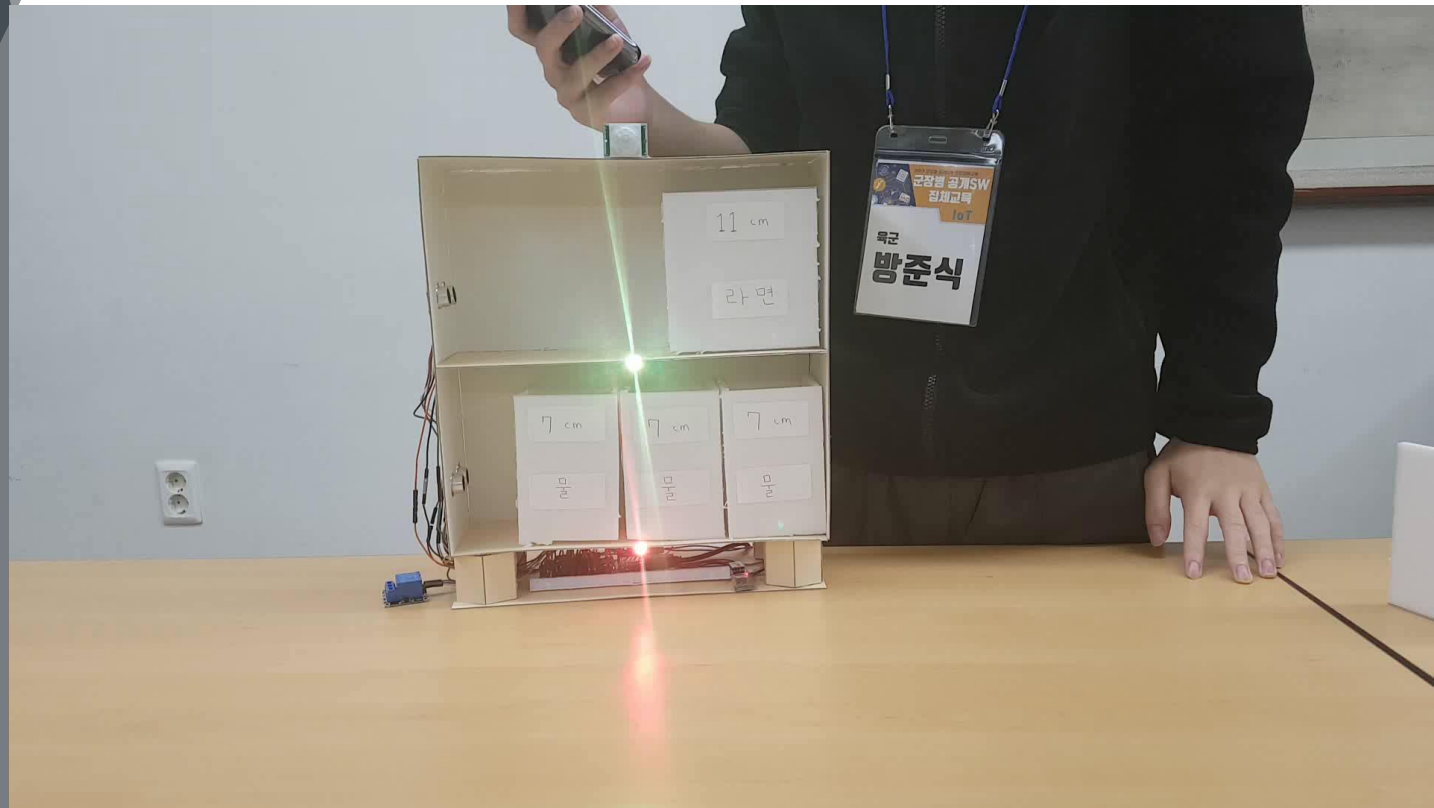
도난 방지

ON

OFF

전송 완료
outting on

crime
전송 완료
outting off



시연

물건 양 파악

물건 이름 및 길이
입력

찾기 기능

외출(방법) 기능

현재 온도 보기

자동 온도 조절

측정 주기 설정

KT

Bluetooth, HD, LTE+, 84%, 오후 10:54

Autowarehousesystem

연결하기

연결끊기

크레딧

전송 완료

Temperature : 21°C

CLEAR

입력

설정

찾기

경보

현재
온도

시연

물건 양 파악

물건 이름 및 길이
입력

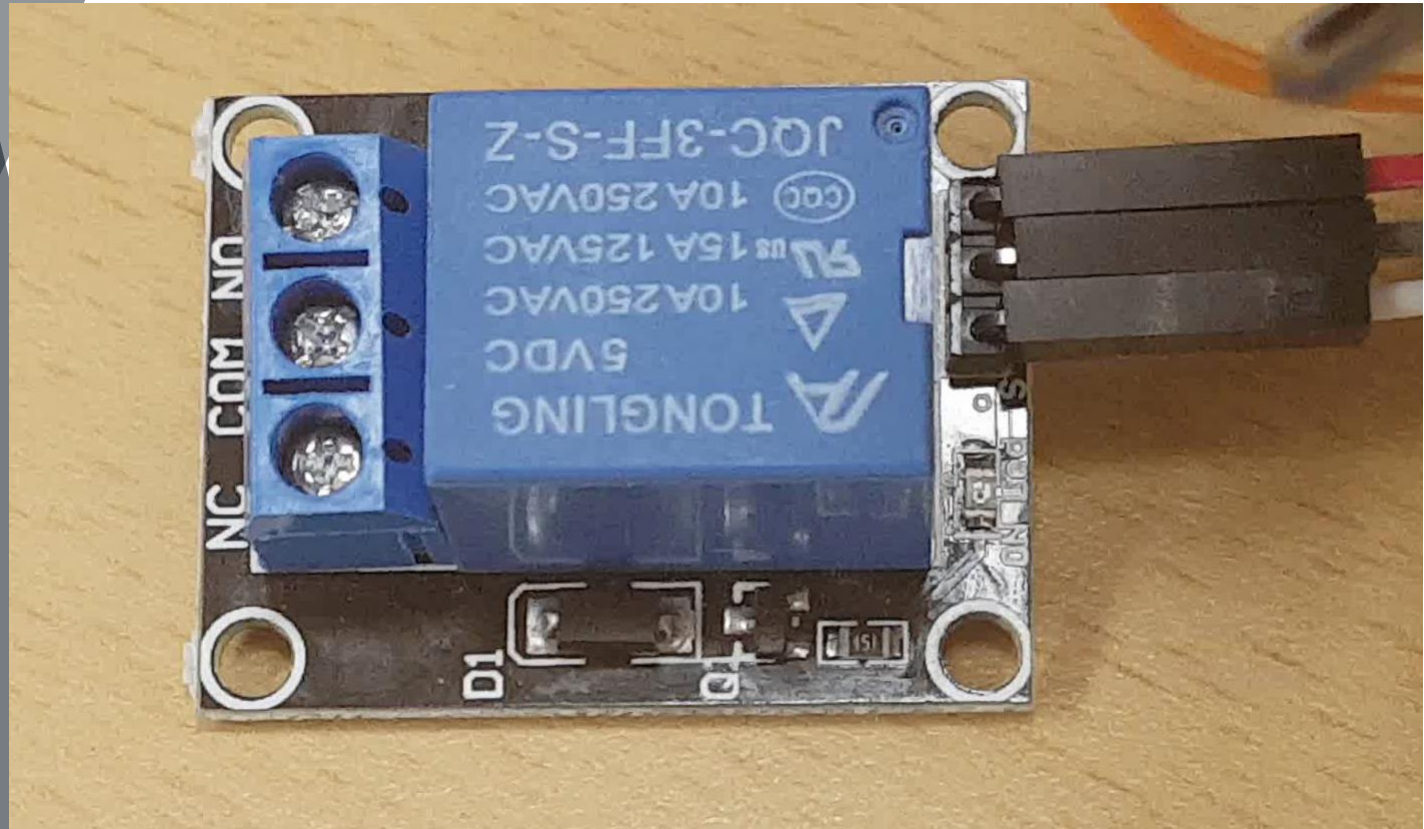
찾기 기능

외출(방법) 기능

현재 온도 보기

자동 온도 조절

측정 주기 설정



시연

물건 양 파악

물건 이름 및 길이
입력

찾기 기능

외출(방법) 기능

현재 온도 보기

자동 온도 조절

측정 주기 설정

Autowarehousesystem

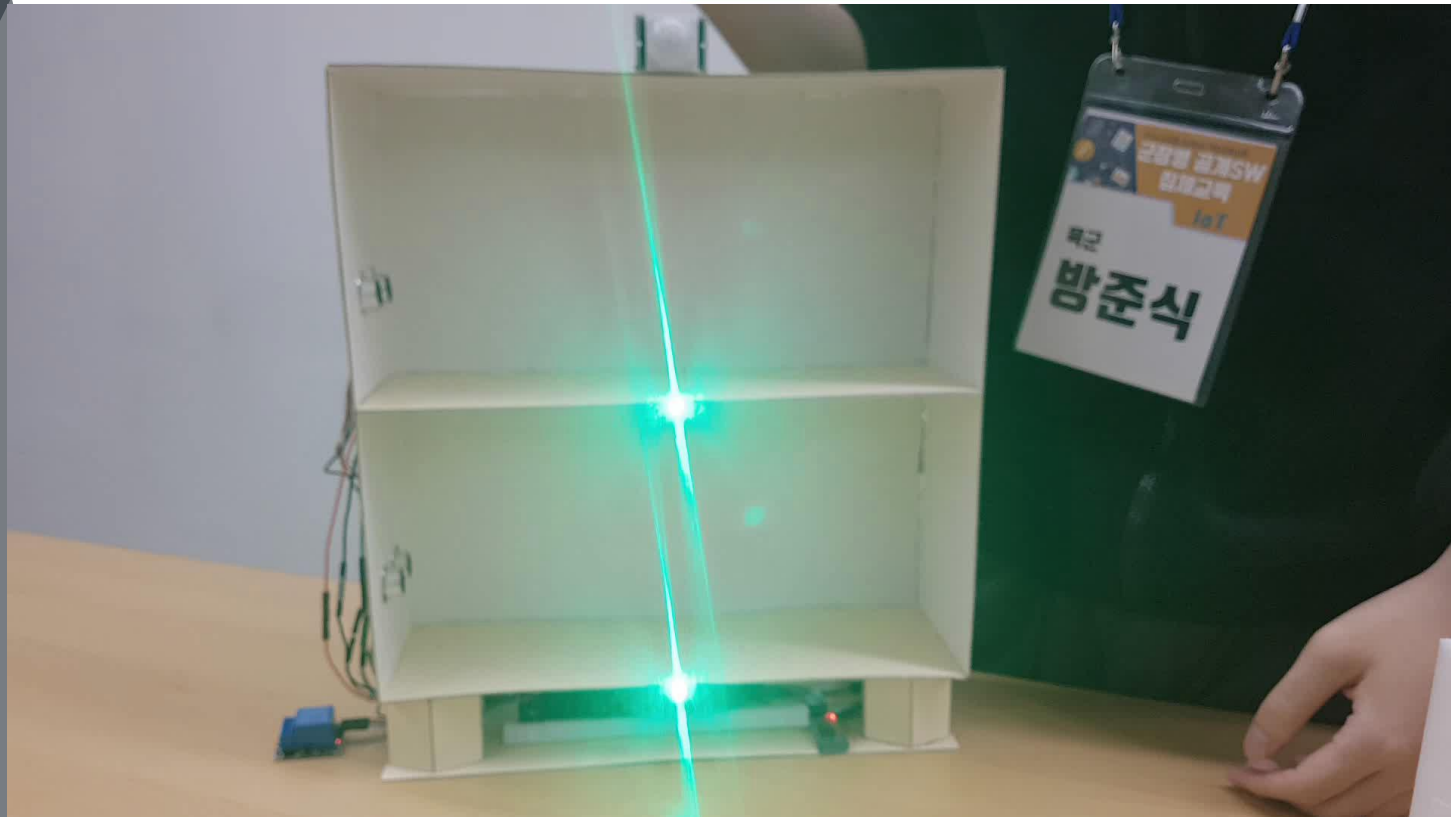
나가기

위험 온도 : temperature

전송

측정 주기 : cycle

전송



기대 효과



빠른 재고 조사



쉬운 위치 파악



자동으로 온도 관리



도둑 방지

개선 방향

자연어 처리 + 음성 인식

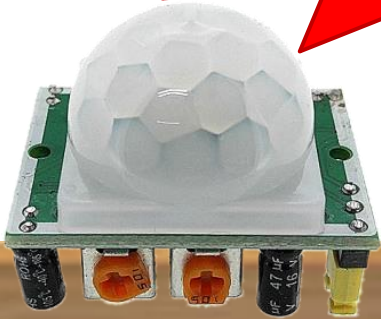
지니야 ~
창고에 삼성 프린터 토너
몇 개 남아있는지 알려줘



개선 방향

카메라 모듈

침입 발생!



도둑 이미지 촬영 후 전송



Q & A