

창의설계 프로젝트

횡단보도 보행안전 통합시스템

20160067 구소현
20161049 장주영
20161106 정은희
20161233 최은서

contents

Part. 1 Introduction

- 1) 선정 배경
- 2) 작품 구상

Part. 2 System

- 1) 시스템 구성
- 2) 알고리즘

Part. 3 Code

- 1) 코드 구성
- 2) LED.h
- 3) TM1637.h
- 4) Main Code

Part. 4 Conclusion

- 1) 영상
- 2) 차이점
- 3) 기대효과

Part 1

Introduction

Part 1. Introduction

1) 선정 배경

스몰비Smombie

'스마트폰'과 '좀비'의 합성어로 스마트폰을 눈에서 떼지 못해 걸음이 느리고 주위를 살피지 못하는 사람



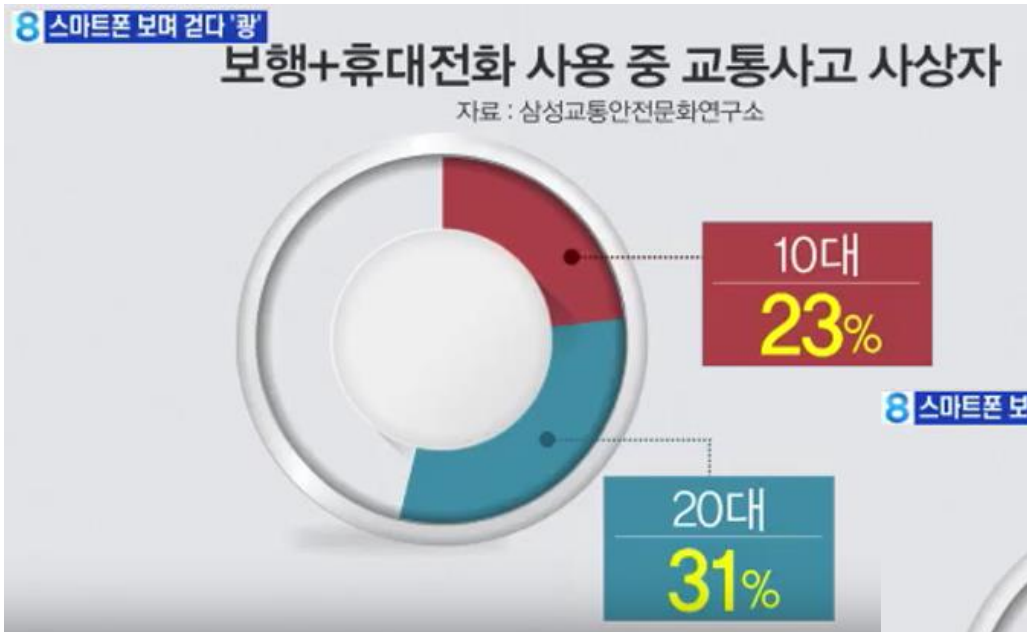
스몰비Smombie

'스마트폰'과 '좀비'의 합성어로 스마트폰을 눈에서 떼지 못해 걸음이 느리고 주위를 살피지 못하는 사람



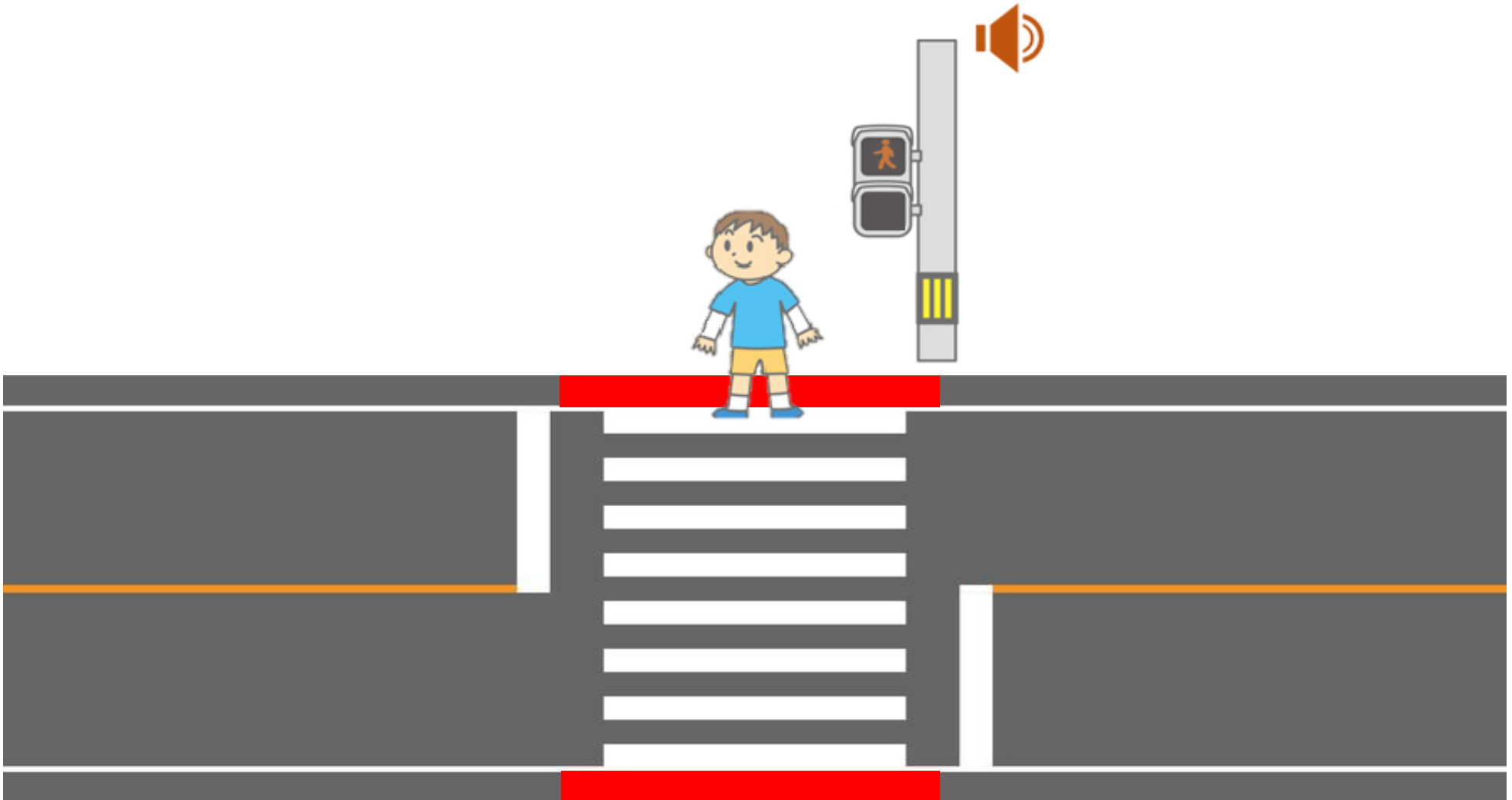
Part 1. Introduction

1) 선정 배경



Part 1. Introduction

2) 작품 구상

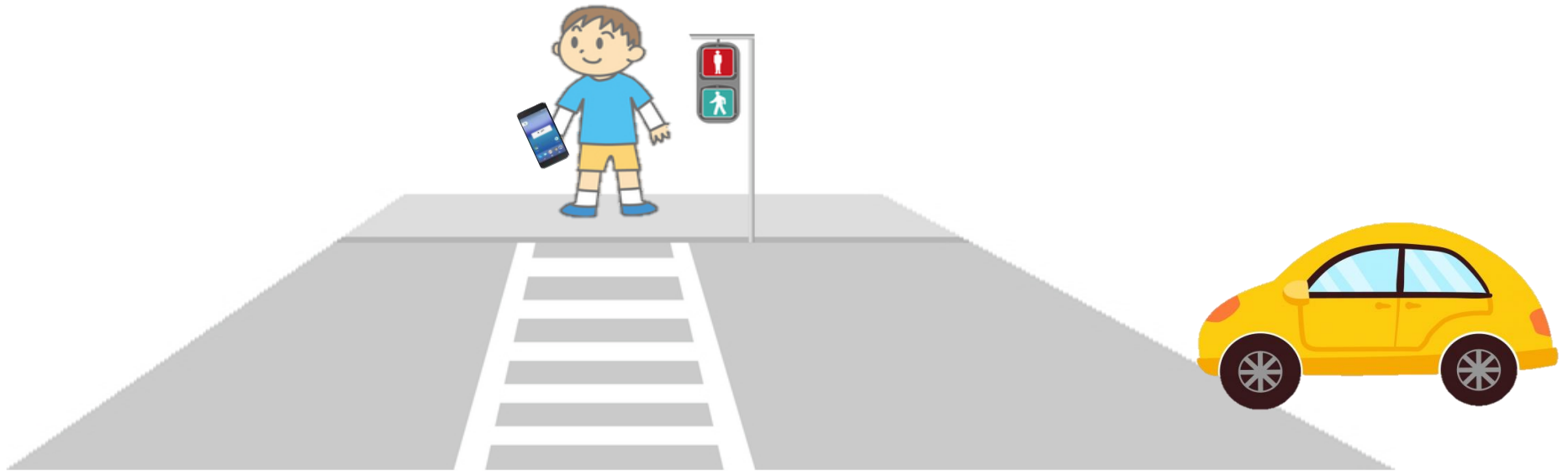


Part 1. Introduction

2) 작품 구상



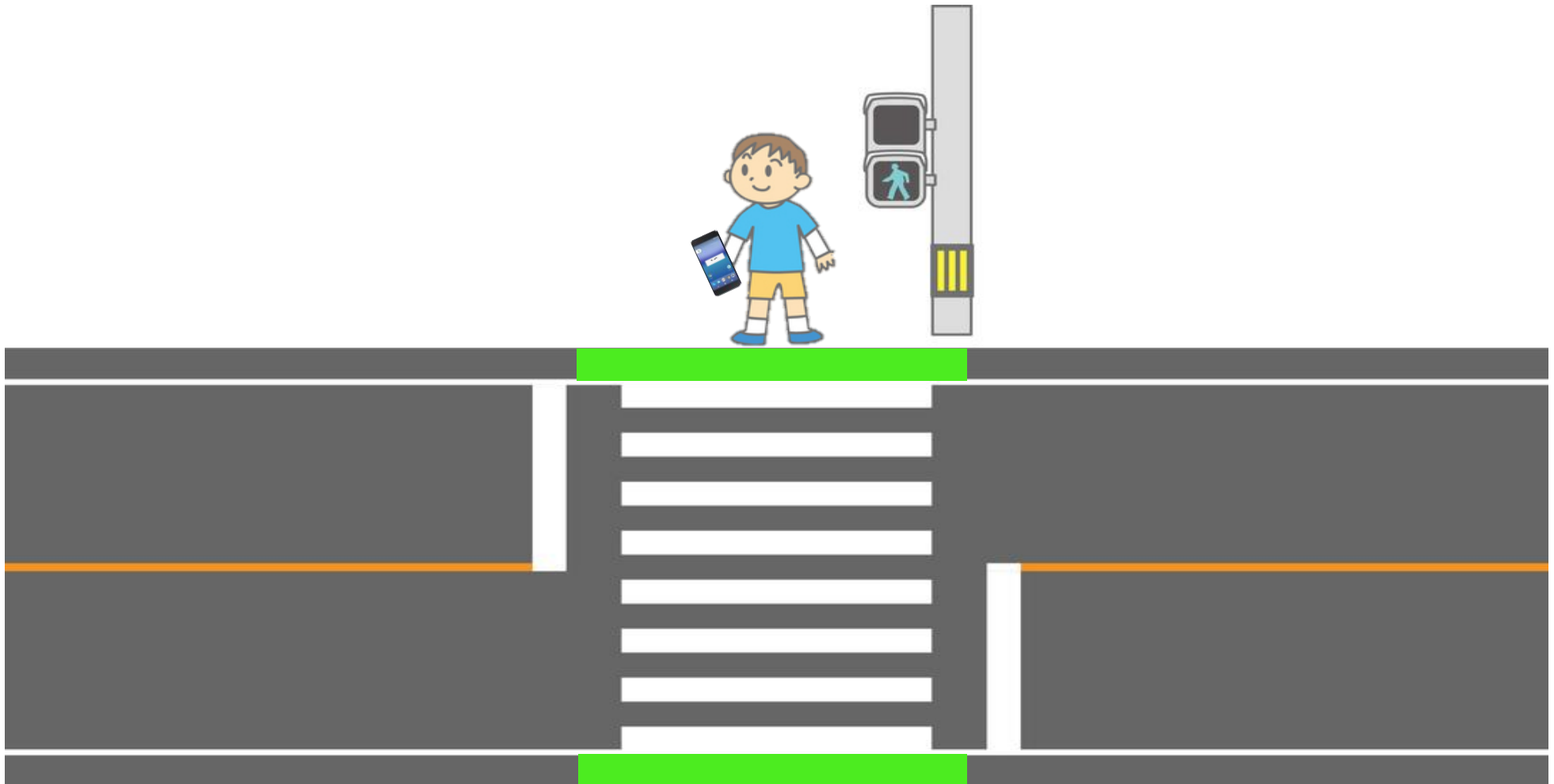
120



Part 1. Introduction

2) 작품 구상

시간표시



Part 1. Introduction

2) 작품 구상



or



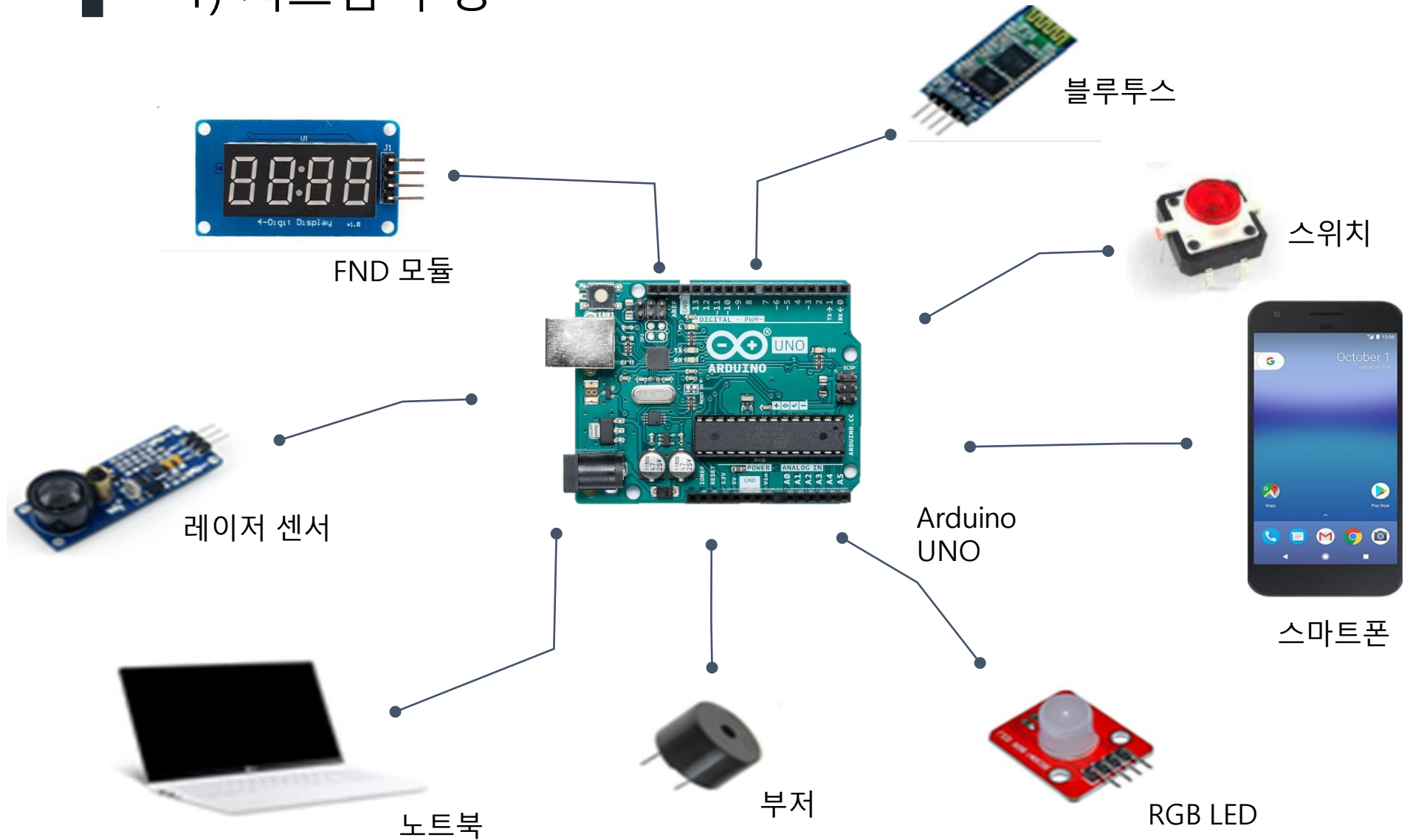
→ 시간연장

Part 2

System

Part2. 작품 구성

1) 시스템 구성

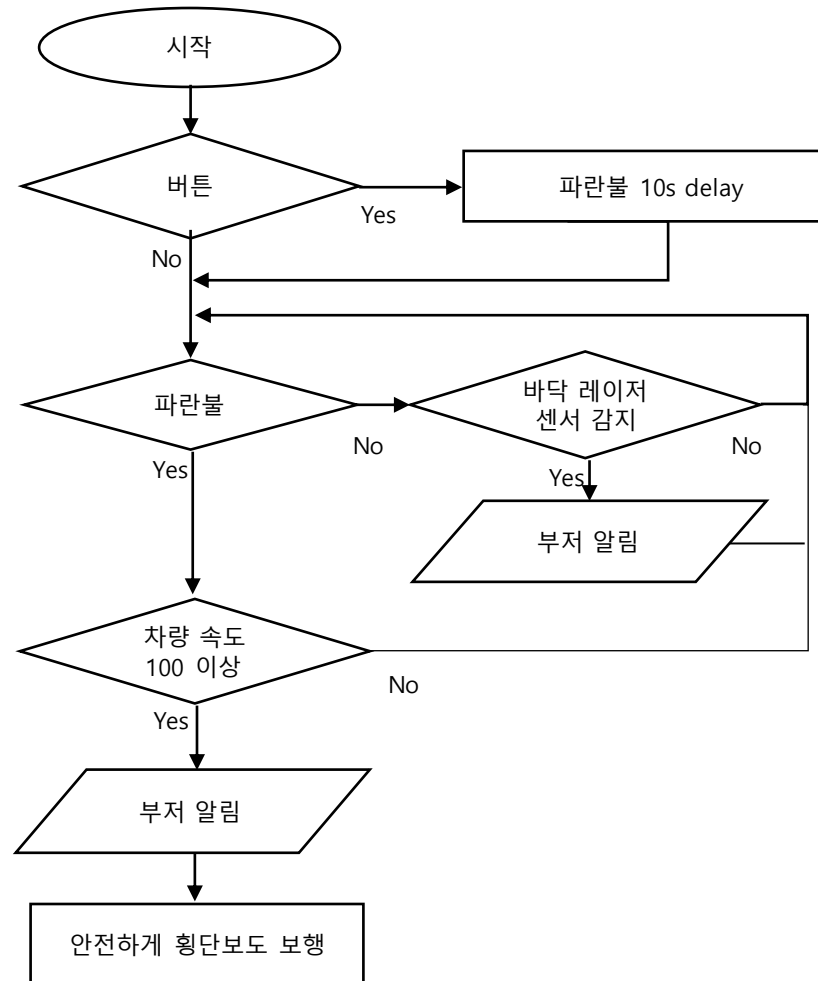


100

10

Part2. 작품 구성

2) 알고리즘



Part 3

Code

Part3. Code

1) Code 구성



Part3. Code

2) LED.h

```
Code Distance.h LED.h TM1637.h
#ifndef _LED_H
#define _LED_H

#define RED 0
#define GREEN 1
#define BLUE 2
#define YELLOW 3
#define OFF 4

class LED {
public:
    LED(uint8_t, uint8_t, uint8_t);
    void setColor(uint8_t);
private:
    uint8_t _pin[3];
};

LED::LED(uint8_t pin0, uint8_t pin1, uint8_t pin2) {
    pinMode(_pin[0] = pin0, OUTPUT);
    pinMode(_pin[1] = pin1, OUTPUT);
    pinMode(_pin[2] = pin2, OUTPUT);
};

void LED::setColor(uint8_t color) {
    uint8_t cList[5] = {0b100, 0b010, 0b001, 0b110, 0b000};
    digitalWrite(_pin[0], !(cList[color] & 0b100));
    digitalWrite(_pin[1], !(cList[color] & 0b010));
    digitalWrite(_pin[2], !(cList[color] & 0b001));
}

#endif
```

(&비트연산 : 1이
겹치는 부분이 있
으면 1)

색에 따라 LED를
켜주는 부분

Part3. Code

3) TM1637.h

```
Code Distance.h LED.h TM1637.h
```

```
#ifndef _TM1637_H
#define _TM1637_H

class TM1637 {
public:
    TM1637(void);
    void init(uint8_t, uint8_t);
    void display(uint8_t, uint8_t, uint8_t = 0);
    void brightness(uint8_t = 5);
    void clear(void);

private:
    void start(void);
    void stop(void);
    uint8_t ack(void);
    void writeBits(uint8_t);
    uint8_t _brightness = 5;
    uint8_t _CLK, _DIO;
    void encoding(uint8_t *);
};

TM1637::TM1637(void) {}
void TM1637::init(uint8_t CLK, uint8_t DIO) {
    _CLK = CLK;
    _DIO = DIO;
    pinMode(_CLK, OUTPUT);
    pinMode(_DIO, OUTPUT);
};

void TM1637::start(void) {
    digitalWrite(_CLK, HIGH);
    digitalWrite(_DIO, HIGH);
    digitalWrite(_DIO, LOW);
    digitalWrite(_CLK, LOW);
};

void TM1637::stop(void) {
    digitalWrite(_CLK, LOW);
    digitalWrite(_DIO, LOW);
    digitalWrite(_CLK, HIGH);
    digitalWrite(_DIO, HIGH);
};
```

```
void TM1637::writeBits(uint8_t data) {
    for(uint8_t i = 0; i < 8; i++) {
        digitalWrite(_CLK, LOW);
        if(data & 0x01) digitalWrite(_DIO, HIGH); //LSB
        else digitalWrite(_DIO, LOW);
        data >>= 1;
        digitalWrite(_CLK, HIGH);
    }
    digitalWrite(_CLK, LOW);
    digitalWrite(_DIO, HIGH);
    digitalWrite(_CLK, HIGH);
    pinMode(_DIO, INPUT);

    delayMicroseconds(50);

    uint8_t ack = digitalRead(_DIO);
    if(ack == 0) {
        pinMode(_DIO, OUTPUT);
        digitalWrite(_DIO, LOW);
    }
    delayMicroseconds(50);
    pinMode(_DIO, OUTPUT);
    delayMicroseconds(50);
};

void TM1637::encoding(uint8_t *data) {
    uint8_t _data[] = {
        0b00111111, //0
        0b00000110, //1
        0b01011011, //2
        0b01001111, //3
        0b01100110, //4
        0b01101101, //5
        0b01111101, //6
        0b00100111, //7
        0b01111111, //8
        0b01101111, //9
        0b01111001, //E
        0b01110111, //R
        0b00111111, //O
        0b00000000 //
    };
};
```

FND에 표시

```
void TM1637::brightness(uint8_t bir) {
    _brightness = bir;
};

void TM1637::display(uint8_t pos, uint8_t data, uint8_t dot) {
    if(data == 'E') data = 10;
    else if(data == 'R') data = 11;
    else if(data == 'O') data = 12;
    else if(data == ' ') data = 13;

    encoding(&data);
    start();
    writeBits(0x44);
    stop();
    start();
    writeBits(0xC0 | (3 - pos));
    writeBits(data | (dot << 7));
    stop();
    start();
    writeBits(0x88 + _brightness);
    stop();
};

void TM1637::clear(void) {
    display(0, ' ');
    display(1, ' ');
    display(2, ' ');
    display(3, ' ');
};

#endif
```

Part3. Code

4) Main Code

```
#include "LED.h"
#include "TM1637.h"
#include <SoftwareSerial.h>
TM1637 FND1;
TM1637 FND2;

#define SOFT_RX 2 // arduino RX, connext Bluetooth TX here
#define SOFT_TX 3 // arduino TX, connext Bluetooth RX here

SoftwareSerial BTSerial(SOFT_RX, SOFT_TX); // (RX, TX) for arduino
const long baud = 9600;

LED Pedestrian_Signal(13, 4, 3); //
LED Vehicle_Signal(A1, A1, A1); // 5 7 6

#define Laser !digitalRead(5)
#define dist_sensor1 !digitalRead(11)
#define dist_sensor2 !digitalRead(12)

#define button !digitalRead(A2)
#define Pedestrian_Signal_Time 20
#define Vehicle_Signal_Time 15

#define distance 13//cm

int32_t uart[9];
```

변수 Define

```
void setup() {

    pinMode(button, OUTPUT);
    Serial.begin(baud);
    Serial.println("Bluetooth control");
    BTSerial.begin(baud); // set the data rate for the BT port

    // put your setup code here, to run once:
    // Pedestrian_Signal.setColor(GREEN);
    FND1.init(7, 6);
    FND1.clear();
    FND2.init(9, 8);
    FND2.clear();
    pinMode(5, INPUT);
    pinMode(11, INPUT);
    pinMode(12, INPUT);
    pinMode(A2, INPUT_PULLUP);
    Serial.begin(9600);

    Pedestrian_Signal.setColor(GREEN);
}

int8_t mode = 0;
uint8_t step = 0;
uint64_t timer = 0;
uint8_t Additional_Delay = 0;
uint32_t speed = 0;
```

Pin 지정

Part3. Code

4) Main Code

신호등 시간에 따른 step 설정

```
void loop() {  
  if (millis() - timer > 1000) {  
    if (mode == -Additional_Delay) { //보행자 신호 GREEN  
      Pedestrian_Signal.setColor(GREEN);  
      step = 0;  
    } else if (mode == Pedestrian_Signal_Time + 0) { //보행자 신호 RED  
      Pedestrian_Signal.setColor(RED);  
      Additional_Delay = 0;  
      step = 1;  
    } else if (mode == Pedestrian_Signal_Time + 1) { //운전자 신호 YELLOW  
      Vehicle_Signal.setColor(YELLOW);  
      step = 2;  
    } else if (mode == Pedestrian_Signal_Time + 2) { //운전자 신호 GREEN  
      Vehicle_Signal.setColor(GREEN);  
      step = 3;  
    } else if (mode == Pedestrian_Signal_Time + 2 + Vehicle_Signal_Time) { //운전자 신호 YELLOW  
      Vehicle_Signal.setColor(YELLOW);  
      step = 4;  
    } else if (mode == Pedestrian_Signal_Time + 2 + Vehicle_Signal_Time + 1) { //운전자 신호 RED  
      Vehicle_Signal.setColor(RED);  
      mode = -1 - Additional_Delay;  
      step = 5;  
    }  
  }  
}
```

STEP	보행자신호	MODE	운전자신호
0	G	0	R
1	R	20	R
2	R	21	Y
3	R	22	G
4	R	37	Y
5	G	38	R

Part3. Code

4) Main Code

```
if (step == 0) {  
  
    uint8_t timer = Pedestrian_Signal_Time - mode;  
    uint8_t digit4 = timer / 1000;  
    uint8_t digit3 = (timer % 1000) / 100;  
    uint8_t digit2 = (timer % 100) / 10;  
    uint8_t digit1 = timer % 10;  
    FND1.display(3, digit4);  
    FND1.display(2, digit3);  
    FND1.display(1, digit2);  
    FND1.display(0, digit1);  
} else FND1.clear();  
timer = millis();  
mode++;  
}
```

파란불일 때
FND에 남은
시간 표시

```
static uint32_t t1 = 0;  
static uint32_t t2 = 0;  
static boolean flag1 = true;  
if (dist_sensor1) {  
    t1 = millis();  
    flag1 = false;  
} if (dist_sensor2) {  
    if(flag1 == false) t2 = millis() + 1;  
    flag1 = true;  
    speed = float(distance) / float(t2 - t1) * 1000.0;  
    uint8_t digit4 = speed / 1000;  
    uint8_t digit3 = (speed % 1000) / 100;  
    uint8_t digit2 = (speed % 100) / 10;  
    uint8_t digit1 = speed % 10;
```

파란불일 때
레이저센서값
차이에 의한 속
도가 100 이상
일 때 부저울림

```
FND2.display(3, digit4);  
FND2.display(2, digit3);  
FND2.display(1, digit2);  
FND2.display(0, digit1);  
if (step == 0 && speed > 100) {  
    tone(A3, 1760, 2000);  
}  
else noTone(A3);  
if (step != 0 && Laser) tone(A0, 880);  
else noTone(A0);  
  
if (button) Additional_Delay = 10;  
  
if(BTSerial.available()){  
    char bt_val = BTSerial.read();  
  
    if(bt_val == 'a') {  
        digitalWrite(button, HIGH);  
        Additional_Delay = 10;  
    }  
}  
  
}
```

빨간불일 때
레이저 감지 →
부저울림

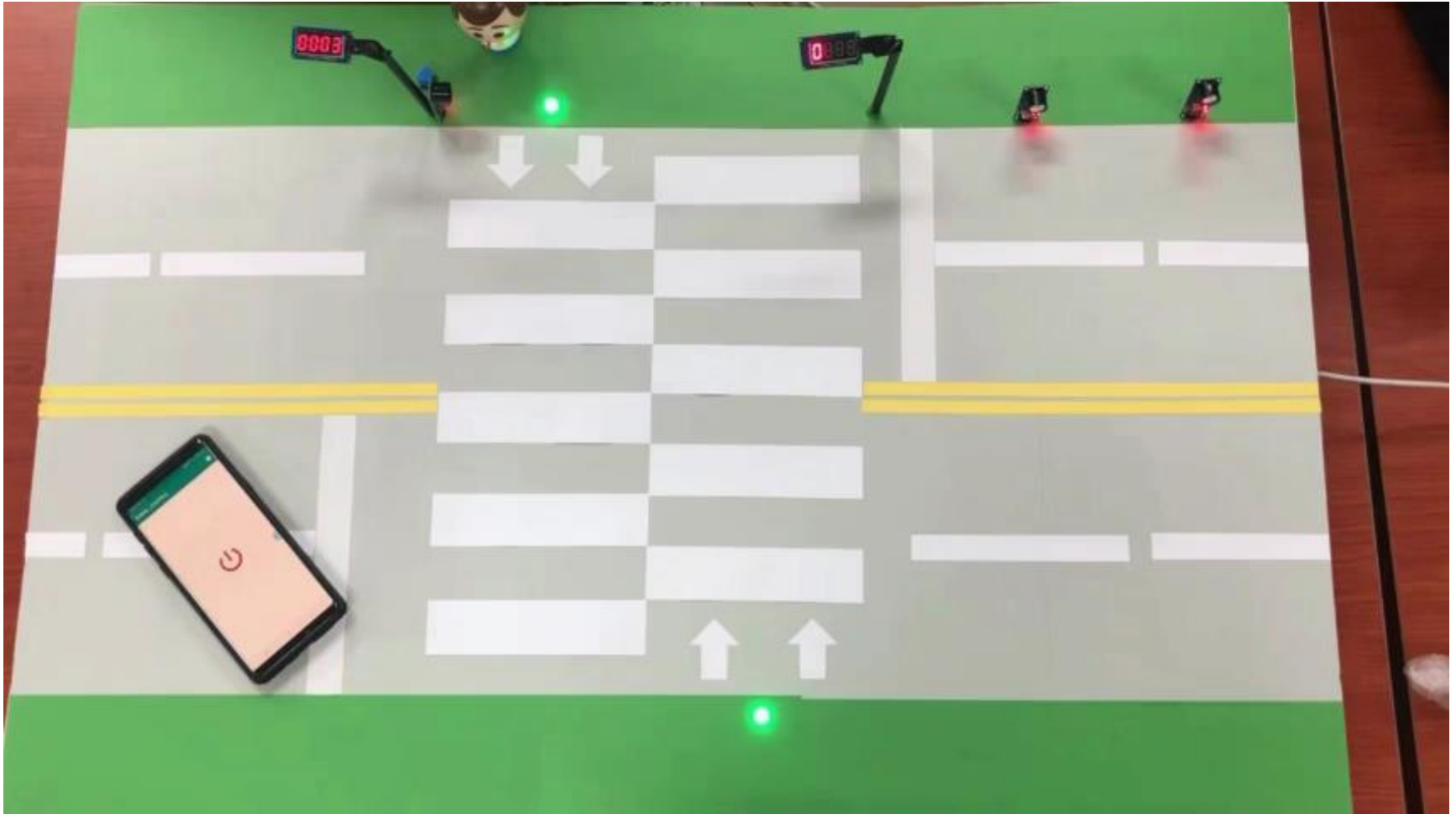
버튼이 눌렸을
때 (스위치, 핸
드폰에서) → 파
란불 시간 delay

Part 4

Conclusion

Part4. Conclusion

1) 동작 영상



Part4. Conclusion

2) 기존 구상 작품과의 차이점

- ▷ 안드로이드 프로그래밍으로 어플리케이션 설계 → 블루투스 컨트롤러 사용
- ▷ 라이더 센서(속도 측정용) → 레이저 센서 2개 이용하여 속도측정 & 속도 표시
- ▷ 신호등 남은 시간 표시 (추가)

Part4. Conclusion

3) 기대효과

- 주의분산 보행사고 중 휴대전화 가 원인인 사고의 비율을 줄일 수 있다.
- 교통약자들의 보행시간이 늘어남으로써 심리적 안정감을 느낀다.
- 스마트 폰을 사용하는 사람이라면 앱을 통하여 누구나 쉽게 접근할 수 있다.
- 빠른 속도로 접근하는 차량 때문에 생기는 인명피해를 줄일 수 있다.

Thank you