[2016 Spring Semester System Programming Team Project]

BLACK SIDE

TEAM 11

MEMBER KIM TAE HO
SEO HYUN JUN
KANG JIN SIL
CHO SUNG HAE
LIM CHAE HA

INTRODUCTION

Sometimes we experience problem of existing black box. For example, weakness of side part (hit and run), memory lack (it always on), doesn't detect tiny vibration. So, we design new black box type in order to compensate the defect of existing black box.

We implement application that detects collision direction/intensity/time, takes a photo and send it to cellphone. We use arduino board, ultrasonic sensor, servo motor and it connect with eCube board. And eCube board communicate with android using WiFi.

MAIN IDEA

Function of sensor

Ultra sonic sensor	We use 4 ultrasonic sensors. Put it on servo motor and it traces moving object.
Vibration sensor	We use 2 vibration sensors. It is located in car's side part. It detects collision intensity in analog.
Servo motor	We use 3 servo motors. It carries ultrasonic sensor and vibration sensor. According to calculated value, it moves.

Function of eCube board components

Keypad	Key 4 is password setting key. Key 8 is enter key. If password is correct, we can use all functions.
Buzzer	If keypad is pressed, buzzer make a sound. And if collision is detected, buzzer make a sound.
Dip switch	Control keypad and initializing.
Text LCD	Show accident time.
7-Segment	Show pressed key number
Bus LED	Show collision intensity.
Full color LED	Usually it is green. But it turns red when it detect collision
Dot matrix	Show collision direction(left, fight).
Camera	It takes a suspect photograph.
OLED	Show collision area.
LCD module	It has 4 menu of manual, camera view, Collision direction view, transmit data to cell phone.

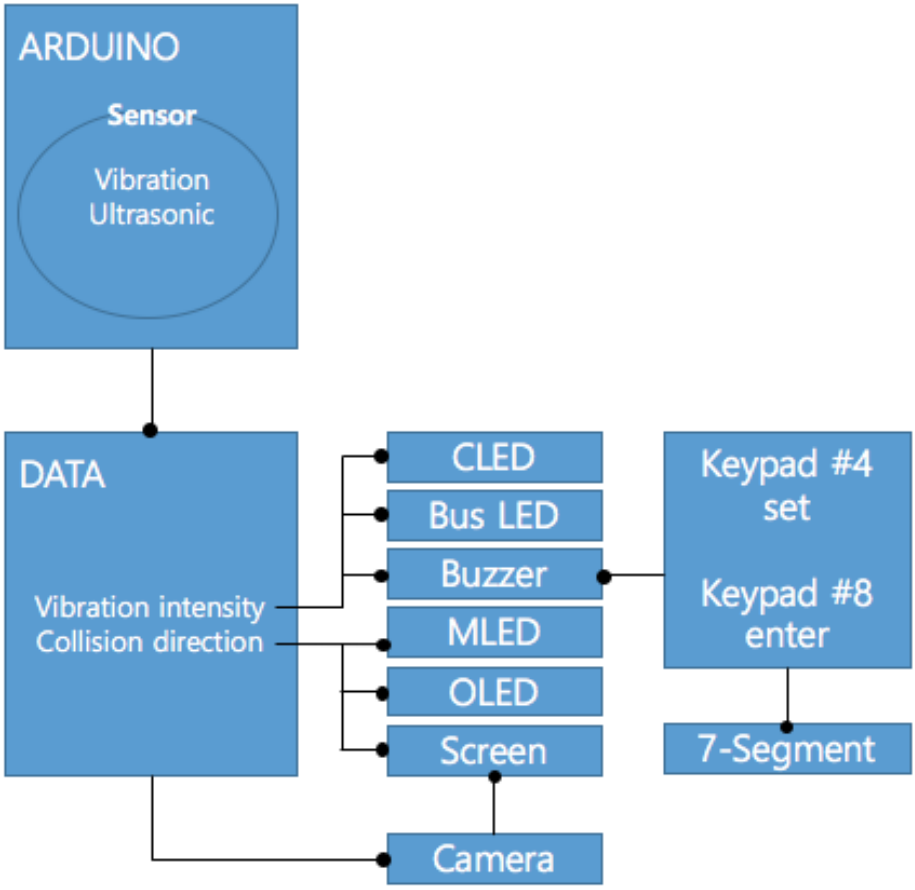
We attach the sensors to Arduino board. Two ultrasonic sensors are attached to car's side part (right side, left side). Servo motor also located in car's side part and they carry each ultrasonic sensor. Vibration sensor are attached to side part too. But it not with motor and ultrasonic senor. One servo motor carries camera and it attached to center of front. If vibration sensors detect collision, Arduino board determines the position. And ultrasonic sensors calculate the angle (object is located) and servo motors follow the angle. Then camera's servo motor moves and camera takes a photograph.

Arduino board sends data that collision or not, collision direction, vibration intensity to eCube board. First, user have to input password in order to use eCube board's components. But user can use manual function in LCD module exceptionally. Keypad are controlled by dip switch. If collision is occurred, buzzer makes sounds and it also controlled by dip switch. Text LCD shows time of accident. Dot matrix and OLED shows collision direction. Bus LED shows collision intensity. Full color LED is usually green light, but collision is occurred it turns red light. Every component is working together by thread. Camera is extended and it working in model. LCD module has 4 menu of manual, collision direction view, camera view and data transmission. If touch data transmission menu, eCube transmit collision data to android using eCube server.

Android receive a transmission of data in real time by using thread. It can check car's state whether to normal or not. And it can show picture if camera takes a photograph. It sends data to web sever (Amazon S3).

DESIGN & IMPLEMENT

Entire algorithm

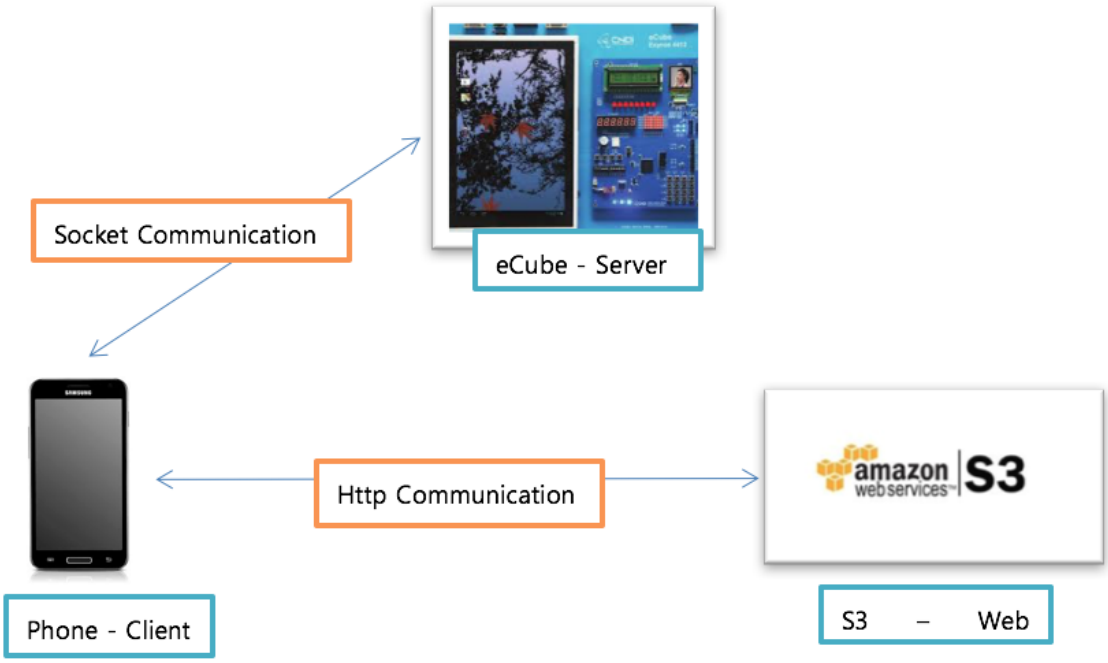


Arduino send vibration intensity, collision direction to eCube components. Each component working with this data. Some components are linked with each other.

• **Arduino analysis**

If Arduino start firstly, Motor equip camera and Two Motors equip ultrasonic sensors will be initialized to middle direction. If vibration sensor detect collision, Arduino compare left value and right value. And execute the ultrasonic sensor and motors at biggest value direction. For example, left vibration value is bigger than right, then left ultrasonic sensor and motors will be executed and measure the distance. In case, within two sensor's distance are less than limit distance, compare two distances and motor moves to bigger distance's direction in 2 degrees. At the same time, motor equip camera Also moves to Object in 3 degrees.

• **Device - Mobile**



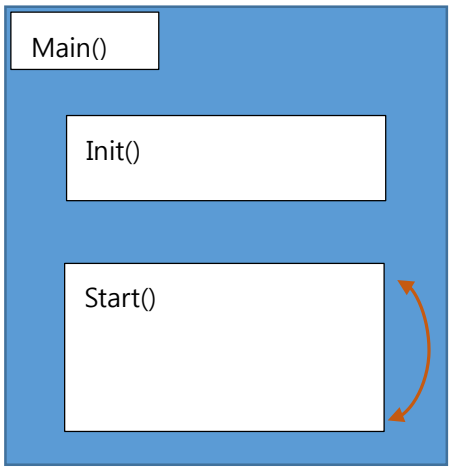
Communication eCube with phone is socket Communication. eCube is server, and phone is client. So they can communicate each other. eCube send vibration value got by vibration sensor to User's phone like form 11L24R. 11L24R mean that Left value is 11 and Right Value is 24. eCube also send picture file name and picture. Then User receive these data so message is printed in alarm message and picture file is immediately transmitted to S3 server, S3 server store pictures and will manage them.

• **Android analysis**

As client, Android communicate with eCube. Android open Thread, and get server's IP and Port Number so that connect socket. If connection is successful, Android will get vibration values like this form '11L24R'. Then 11 , 24 each values are printed at textview. And then Android get Picture and Picture's Name by Server, and Stores them. The stored picture is transmitted to amazon S3 server by using Http communication. "S3로 사진전송" button is confirming camera. This buttons are example buttons, it uses in this case communication is failed. If you click "S3로 사진전송" button, temporary picture is transmitted to amazon S3 server.

Detail algorithm

• **Main**



It parses the value (ex. 11L545R) received from the Arduino. Separate the left and right of collision impact side, also it extracts a vibration value, respectively. Through a separate value controls the thread and each function.

Init() in main() is setting only once threads and pile drivers. Start() in main() is initializes the parameters for control of the board. And then continue to operate the system through the loop.

• Dip switch

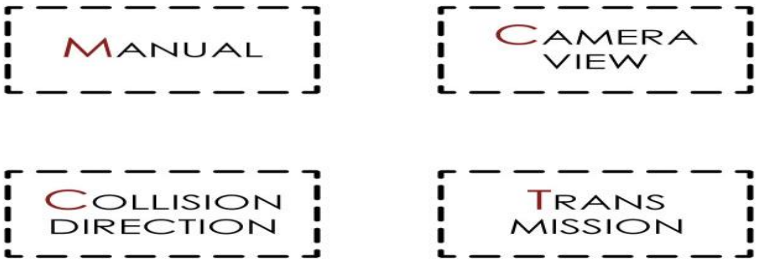
Dip switch	Function
if dip state = 1 ,3	Initialize e-cube board and main variables
if dip state = 2, 3	Lock and unlock the keypad

It is used to switch the control values in the other functions. So dipswtich was used as a thread for the system control.

• Camera

DrawFromRGB565() save the camera's size values as entered in the display frame using RGB. And bmp_generator() make BMP file using a saved IMG file. //team 7 help us how save bmpfile. Camera()is can show in real time and capture through DrawFromRGB565().

• **Touch screen**



Manual is activated regardless of dipswitch of 2,3 values. But another functions is operated only dipswitch's value is 1,3.
Camera features such as a rearview camera. It can determine a user's back in real time.
Collision Direction show the crash happened at any side of your car.
Transmission will be sent to user when user want the user to take pictures.
These are used for the pixel of the bitmap file to the coordinates, and executes each of the functions in the if statement.

• **Bitmap**

This is a function of displaying bmp file on LCD module. It can display bmp file (maximum 1280 X 800 pixels). It can locate the file like this. This means, you can discover x and y's location by letting the location where two arrows meet zero.

This function represents 'bitmaptest (file, pfbmap, x, y)', role of which is displaying file saved in putty on x, y exactly. The function, 'read_bmp()', in 'bitmaptest', identifies file if this is 24 bits and bmp file, because LCD module cannot display file which is not bmp file and 24 bits. We let this to display background bmp file and other bmp files by touching buttons on x < 640 pixel location.

• **Text LCD**

This is function of displaying string maximum size 16 on text LCD. We use the function 'tLCDtest (mode, line, column)' mainly, which has two modes. One is 'displayMode()', which displays string converted from numbers listed time. The other is 'clearscreen()', which resets and energize text LCD.

• **OLED**

This is function of displaying img file on oled. We use the function 'oledtest(mode)', which has several modes. It can display file when the functions are energized in order, reset(), init(), imageloading(file). Each role is resetting file, initiating oled, and displaying file. If these functions are energized in order, oled can display file.

Originally, we planned to displaying several img files for car's condition. But unfortunately, this occur damaging image. So, we changed plan to displaying only one car image file.

• **Bus LED**

This is function of turning on 8 leds. The form of function is 'ledDisplay(wdata)'. According to this 'wdata', leds can be changed to be turned on or off. We can see more easily if 8-binary numbers are reversed. For example, if you want turn on left-1st led, the form looks like '10000000'. When it is reversed, it can be 000000001. But, this is not end. Wdata should be hexadecimal. So, it should be 0X01.

On this, we planed to represent damage by using this. When wdata becomes larger and larger, turned on leds increase. For this, we planed to represent damage by inserting (2^n-1) into wdata

• **Buzzer**

	Warning_Buzzer	lockOpen_Buzzer	lockClose_Buzzer
Sound tone			

We make buzzer to active when press key or password is right or wrong and collision is detected. Buzzer driver file is located in "dev/cnbuzzer", and give integer buzzer_fd value if file is not there, error will be detected. Wen we press keypad, keyBuzzer() function is executed so that makes sound. When car detect collision, warningBuzzer() is executed so that high tone sound is occurred. When we enter password correctly, then lockOpen_Buzzer() is executed. But lockClose_Buzzer will be executed if we enter wrong password.

• **Dot matrix (MLED)**

	Left Dot				Right Dot				
0x01				■	■				
0x02			■			■			
0x04		■					■		
0x08	■							■	
0x10		■					■		
0x20			■			■			
0x40				■	■				

Dot matrix's driver is located in "dev/cnmled" so return integer mled_fd. Function mledDisplay() will select location using argument integer left, right. If left value is 1, num value will 1. And right value is 1, then num value is 0. Else situation will return num value 2. Num also select position. Because durationTime gives 8, so dot matrix stays within 8 seconds.

There are two arrays in Const unsigned short Numdata this arrays are left, right dot values. First row value is 0x01, second row value is 0x02, third row value is 0x04… last row value is 0x40 in order that you print firt row and last row, you should add 0x01 and 0x40. And 0x41 is one column value.

• **KeyPad**

0)	1)1	2)2	3)3
4)98	5)4	6)5	7)6
8)97	9)7	10)8	11)9
12)99	13)99	14)0	15)99

This key will be executed by switching on dip switch 2. Example, if dip_state that indicate dip switch value is 2 or 3, keypad will be executed. Keypad driver is in "dev/cnkey". If is opened at function openKeypad() and return key_fd. Each keys return rdata value. Because keypads role is entering password, so it returns 0~9 rdata values. Reference picture, rdata 98 will start entering password. Rdata 97 will end entering password. Rdata 9 is nothing. Rdata returns to tmp at main.c and tmp will calculate with enter_pass, finally enter to 7-segment

CHANGED MATTER

Sensor/Actuator/Module

In proposal, we were trying to use one stepping motor. But we came to know that using servo motor is more efficient than using stepping motor through TA's advice. Also we felt the need to use two more motors. Motor was originally scheduled to carry only camera. But in order to track moving object, we had to use two servo motor that carries four ultrasonic sensors.

Also we planned to use two ultrasonic sensors (each one side). But it cannot measure angle with single ultrasonic sensor. Because tracking target moves. So we use four ultrasonic sensors.

And we were trying to use Arduino Wi-Fi shield. It causes lack of budget because it is very expensive. And we came to know it is unnecessary. We didn't know eCube board's serial communication function. Because we got to know eCube board's serial communication, we exclude Wi-Fi shield from budget.

eCube board components

We change some of eCube components function. We planned 7-segment display hour, minute and second in real time. But it has big problem for thread. During time of 7-segment working, the rest of components are not working. Only 7-segment is working. Therefore, we changed it display pressed keypad's number.

Text LCD just displayed year, month, day. But 7-segment function is changed, text LCD display hour and minute too.

Keypad function was text LCD and 7-segment setting. But time can be set by putty. So keypad function is changed it set password and. We implement enter key in keypad too.

We planned dip switch control text LCD and 7-segment. We got to know it is unnecessary. It changed 1th switch is initializing all eCube components and 2nd switch control keypad function. If dip switch is down, corresponding function is locked.

LCD module displayed camera photo printing and save accident record. But it is so limited. We add a four menu that is manual, camera view, collision direction view, data transmission. Manual help user who first use our product. Data transmission function planning to communicate with android. In proposal, implementing android is abstract. So we materialize it use LCD module.

Others

Our proposal did not consider the web server. Because make the most of function, we use amazon web server S3. We can upload image file, save and share using web server. It communicates with android.

TEST & EVALUATION

Compile environment

We work in the Ubuntu environment. And through the gcc cross compiler made the object files for ecube 4412 board.

we use bash script. Compile command is here.

```
arm-none-linux-gnueabi-gcc -o main main.c ledtest.c ledtest.h cledtest.c cledtest.h oledtest.c oledtest.h mledtest.c mledtest.h fndtest.c fndtest.h tlcdtest.c tlcdtest.h dipswtest.c dipswtest.h keytest.c keytest.h buzzertest.c buzzertest.h bitmap.c bitmap.h touchapp.c touchapp.h camera.c camera.h SecBuffer.h videodev2.h -pthread –lrt
```

Test result

Vibration value of the left and right sides operate each of the servo motor and an ultrasonic

sensor. And Users will only be able to activate the keypad using the DIP switch and control the system via password entered using the keypad. 7-segment show the inputting passwords in real time. The intensity of vibration values entered in the board and it is expressed by busled, cled and the location of the vibration was appeared collision detection menu in touch screen and mled. Take pictures when the arduino's servo motor starts turning. And then the ecube board is transfer photos to your Android.

Bug report

1. Camera bug. We use two camera function. One is for real time rear view, another is for take a picture. First bug occurs using camera function (take a picture or real time view) when using different camera function. We don't fix this bug.

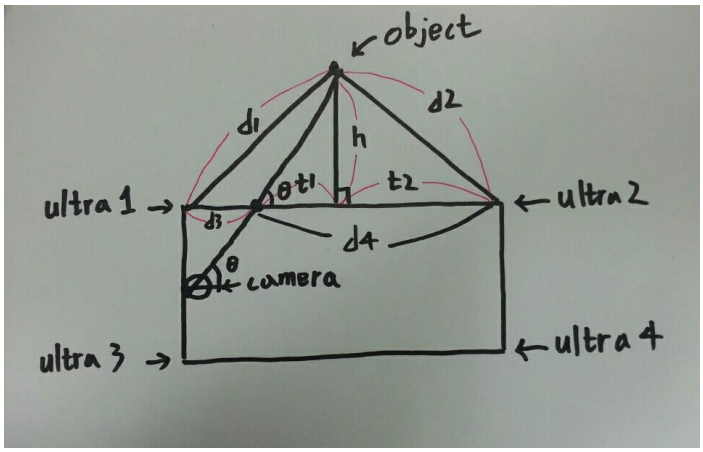
2. It did not synchronize between Dot matrix's operate time and arduino's rotate time. So if the incoming continuous vibration value, the output of the matrix becomes odd.

Improvement

1. If we try to control the Arduino via the target board, the synchronization between the Arduino and board would have done better.

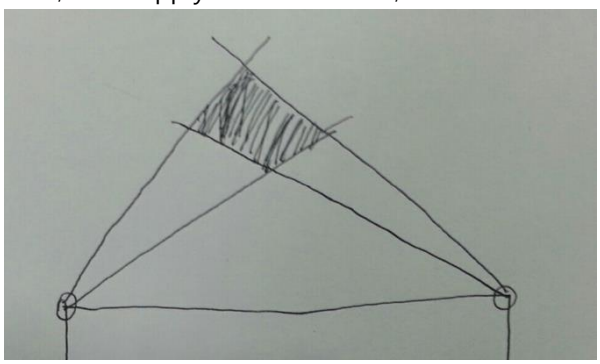
2. Since lack of understanding of the network, eCube-board transmit information to Android. Then android transmit information that was transmitted to Web server If at the board send photos and information to Web server and Android would have been more efficient.

3. The Obtaining angle for direction of motor attached camera. In order to rotate direction to Object detected by ultrasonic sensor, we change 3 plans. In first, we attach sensor at car's 4 edges, and detect object like this.

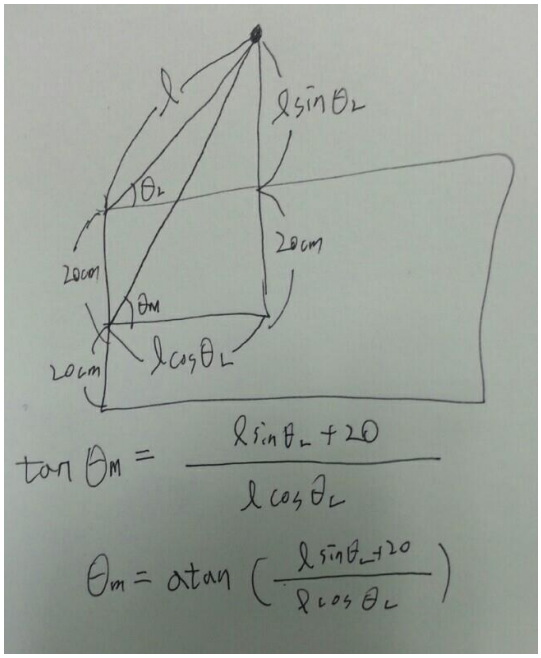


```
s = (distance1 + distance2 + (distance3 + distance4)) / 2;
r = sqrt(s*(s-distance1)*(s-distance2)*(s-(distance3 + distance4)));
r2=(r*distance4/(distance3+distance4));
h=2*r2/distance4;
t2=sqrt((distance2*distance2)-(h*h));
t1=distance4-t2;
theta = atan(h/t1)*180/PI;
```

But, if we apply to this formula,



It will have weakness that measured range small. So we change method to attach two ultrasonic sensors to find angle at one side.



```
void findMidangle() {
  if(left_val > right_val) {
    mid_angle = 135;
    if(distance_l1 <= limit_distance && distance_l1 >= 0) {
      find_l = distance_l1;
    }
    else if(distance_l2 <= limit_distance && distance_l2 >= 0) {
      find_l = distance_l2;
    }

    mid_angle = atan((find_l+sin(left_angle)+20)/(find_l+cos(left_angle)));
    mid_angle = mid_angle*180/PI + 90;
  }
  else {
    mid_angle = 45;
    if(distance_r1 <= limit_distance && distance_r1 >= 0) {
      find_r = distance_r1;
    }
    else if(distance_r2 <= limit_distance && distance_r2 >= 0) {
      find_r = distance_r2;
    }

    mid_angle = atan((find_r+sin(right_angle))/(find_r+cos(right_angle)+20));
    mid_angle = mid_angle*180/PI;
  }
```

But this formula also is not realized finally because theta value is don't applied to middle motor.

4. capture camera operate well. but when board transmit picture to android, picture is seen black. Maybe have problem at transmit process.

ATTACHEMENT

Role

Kim Tae Ho : Main builder , Camera, mled, oled, bus led, dipswitch, cled
Seo Hyun Jun : Arduino, Android, Web server
Kang Jin Sil : Keypad, Buzzer, UI design
Lim Chae Ha : LCD module, TLCD, touch screen
Cho Sung Hae : Android, 7-Segment, Web server, touch screen

Meeting report / Schedule

DATE	20160501	LOCATION	STUDENT UNION 2	ATTENDEE	ALL
CONTENTS	Discuss future plan. 1. Understanding of principles about Arduino 2. How to detect direction 3. Sensor study on our own + organize sensor code				

DATE	20160510	LOCATION	Practical Room	ATTENDEE	ALL
CONTENTS	Write auduino code 1. Checking motion of servo motor/ultrasonic sensor/vibration sensor 2. Understanding and using of breadboard				

DATE	20160511	LOCATION	Practical Room	ATTENDEE	ALL
CONTENTS	Object location detecting use of ultrasonic sensor 1. Use one ultrasonic sensor -> fail 2. Figure out a way with two ultrasonic sensor				

DATE	20160514	LOCATION	Student Union 2	ATTENDEE	ALL
CONTENTS	Discussion for presentation material 1. Modify flaws 2. Think deeply about solution 3. Discuss about solution 4. Write PPT				

DATE	20160520	LOCATION	Practical Room	ATTENDEE	ALL
CONTENTS	Analyze codes of embedded board. Combine codes of Arduino sensors. 1. Understand process of outputting colors on lcd module 2. Understand processing led 3. Analyze code of dot matrix & 7-segment				

DATE	20160521	LOCATION	Practical Room	ATTENDEE	ALL
CONTENTS	Modify Arduino sensors. 1. Buy more sensors add on Arduino 2. Combine additional codes 3. Analyze code connecting Arduino to embedded board Analyze codes of embedded board 1. Analyze code of dot matrix and 7-segment 2. Process code of led 3. Understand process of outputting img on oled Problem : no output textlcd, how to convert to img file				

DATE	20160527	LOCATION	Practical Room	ATTENDEE	ALL
CONTENTS	Complete Arduino sensor project. Solve the problem of outputting tlcd and converting file to img file Produce codes of embedded board 1. Dot matrix -> left : < right : > 2. 7-segment -> time Design lcd module UI				
DATE	20160528	LOCATION	Practical Room	ATTENDEE	ALL
CONTENTS	Analyze android to connect with embedded board Produce codes of embedded board 1. led -> damage degree 2. oled -> damage place problem : damage on led module's bmp file				

DATE	20160530	LOCATION	Practical Room	ATTENDEE	ALL
CONTENTS	Analyze android to connect with embedded board & how to let embedded board release wi-fi Analyze codes of embedded board : keypad Solve the problem of damage on led module -> fail Combine codes which have been completed				

DATE	20160531	LOCATION	Practical Room	ATTENDEE	ALL
CONTENTS	Produce codes of embedded board 1. insert buzzer into keypad 2. solve the problem of damage on led module -> complete 3. connect Arduino to embedded board Modify plan 1. 7-segment : time -> keypad's number 2. oled : damage place -> normal car image				

DATE	20160602	LOCATION	Practical Room	ATTENDEE	ALL
CONTENTS	Produce codes of embedded board 1. Insert touchapp into lcd module 2. Design UI and insert into lcd module 3. Apply 7-segment for damage degree Modify plan : damage place oled -> lcd module				

DATE	20160603	LOCATION	Practical Room	ATTENDEE	ALL
CONTENTS	Produce codes of embedded board 1. Insert touchapp into lcd module -> complete 2. Produce screen if lcd touched 3. Apply dot matrix for damage place				

DATE	20160604	LOCATION	Practical Room	ATTENDEE	ALL
CONTENTS	Produce codes of embedded board 1. Insert camera into lcd module if button touched 2. Complete keypad setting : insert function of password				

DATE	20160607	LOCATION	Practical Room	ATTENDEE	ALL
CONTENTS	Build the virtual server and try to connect with android Let embedded board to release wi-fi Produce codes of embedded board : camera setting Problem : dot matrix is not released properly				

DATE	20160609	LOCATION	Practical Room	ATTENDEE	ALL
CONTENTS	Build the virtual server and try to connect with android Build car model Try to connect android to embedded board Solve the problem of dot matrix Problem : invalid output of camera screen				

DATE	201606011	LOCATION	Practical Room	ATTENDEE	ALL
CONTENTS	Solve the problem of camera screen Problem : cannot produce camera capture -> fail				